

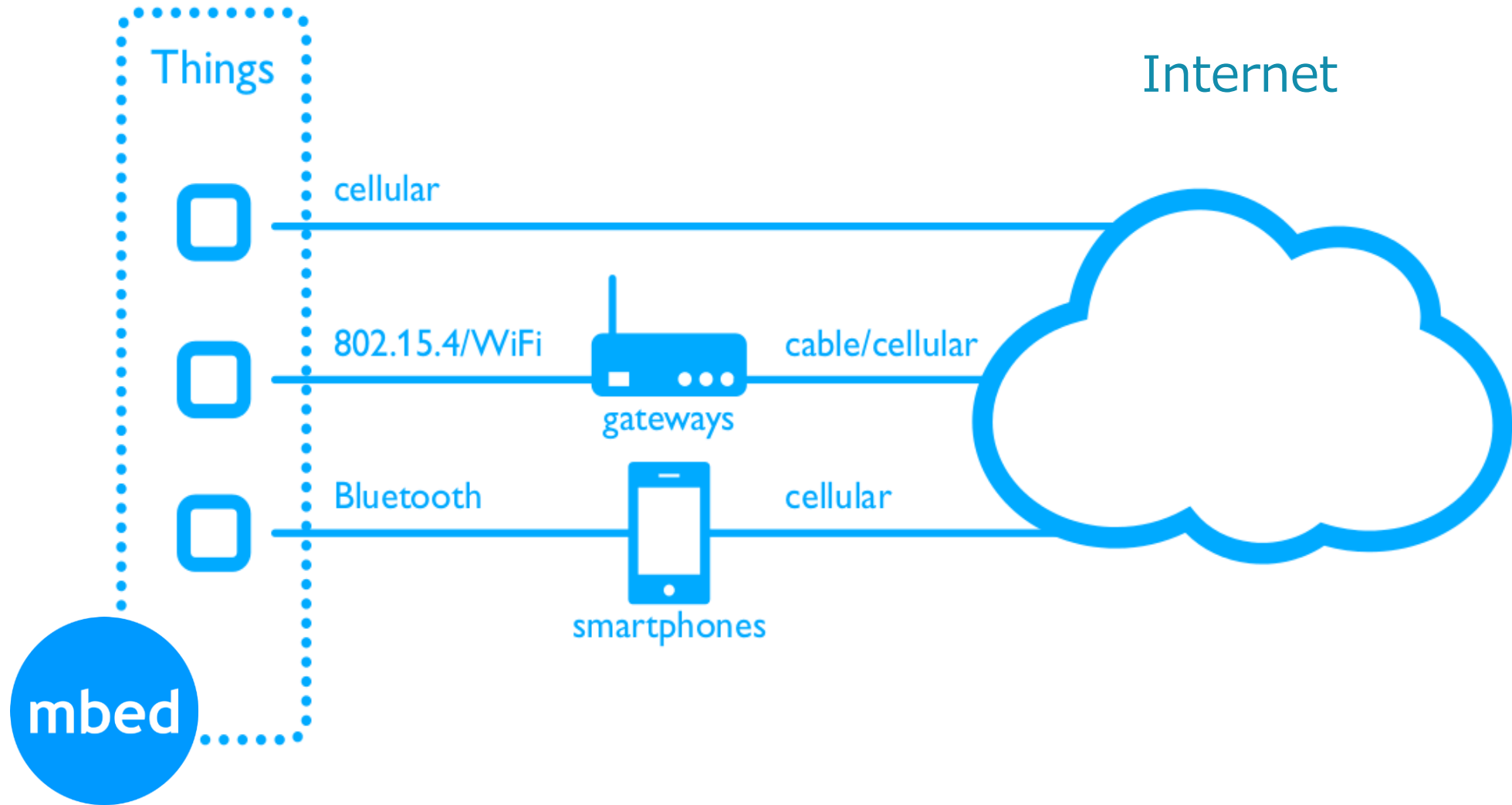
# Cortex<sup>®</sup>-M MCU向け IoT開発プラットフォーム mbed<sup>™</sup>の解説

アーム株式会社  
スタッフアプリケーションエンジニア  
渡會 豊政

# Agenda

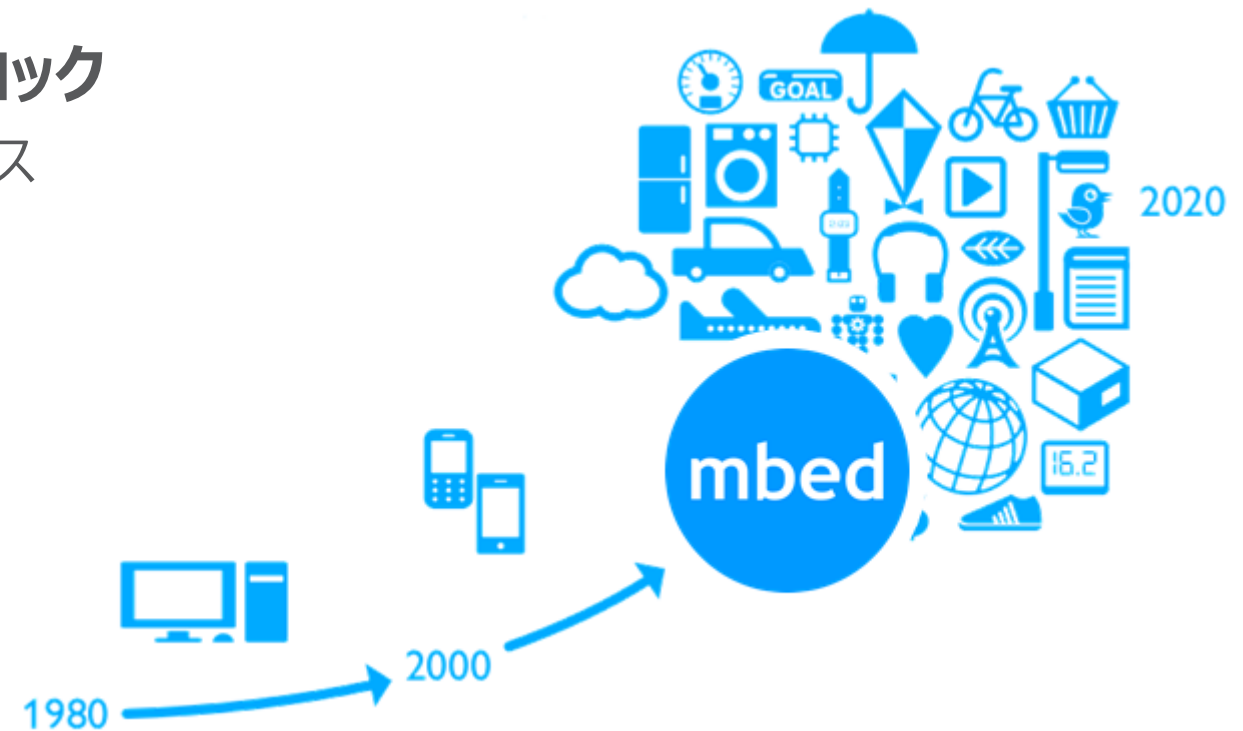
- IoT (Internet of Things) とは ?
- mbedの特徴
- SDK
- HDK
- プラットフォーム
- クラウド開発環境
- コンポーネントライブラリ

# IoT (Internet of Things) とは？



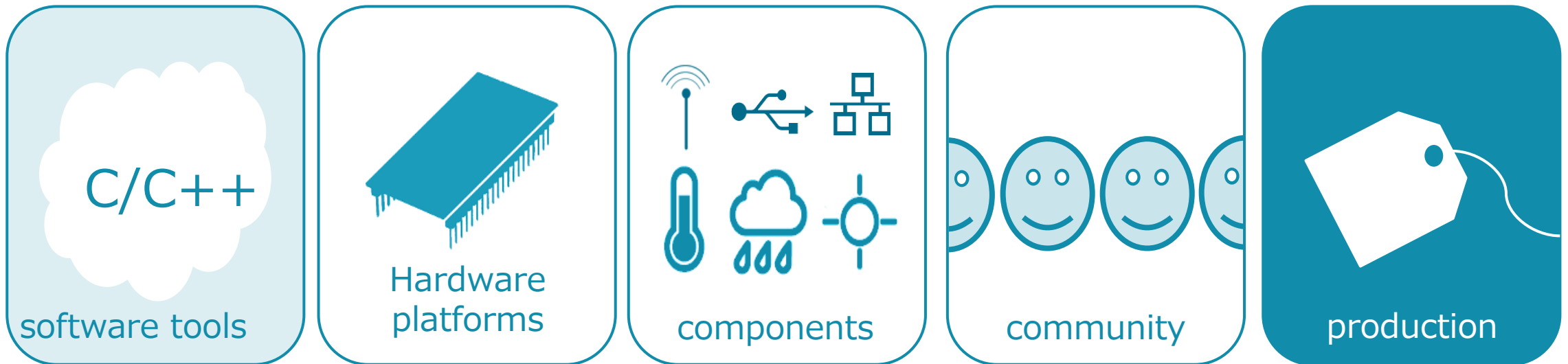
# IoT デバイスの開発プラットフォームとしての mbed™

- **優れたプラットフォームとツール**
  - 簡単なセットアップ、高速プロトタイプ、プロダクション
- **ソフトウェアとハードウェアのビルディングブロック**
  - ARM® MCU、無線、ペリフェラル、クラウドサービス
- **開発者のためのデザイン**
  - オープンソース、サポート
- **開発者とパートナーのエコシステム**
  - コミュニティとのコラボレーション



# mbed プラットフォームとエコシステム

*Enabling the ubiquitous intelligence and connectivity that underpins the Internet of Things*



# mbed の特徴

- ARM Cortex-M を使用した IoT デバイス開発プラットフォーム
- ARM MCU を手軽に始める最短経路
- クラウド開発環境
  - オンラインコンパイラ
- ドラッグ&ドロップ・プログラミング
  - CMSIS-DAPデバッグ機能
- C/C++ APIベース開発
  - 検証済みの豊富なコンポーネント・ライブラリ



# Hello mbed world! ... LED を点滅させる

1. USB ケーブルで、mbed と PC を接続
2. 新規プロジェクト作成
3. ビルド
4. バイナリをドラッグ & ドロップ

DigitalOut クラスのコンストラクタ

指定された GPIO ポートに出力

```
main.cpp X
1 #include "mbed.h"
2
3 DigitalOut myled(LED1);
4
5 int main() {
6     while(1) {
7         myled = 1;
8         wait(0.2);
9         myled = 0;
10        wait(0.2);
11    }
12 }
13
```

# mbed SDK



## ■ 開発者のための設計

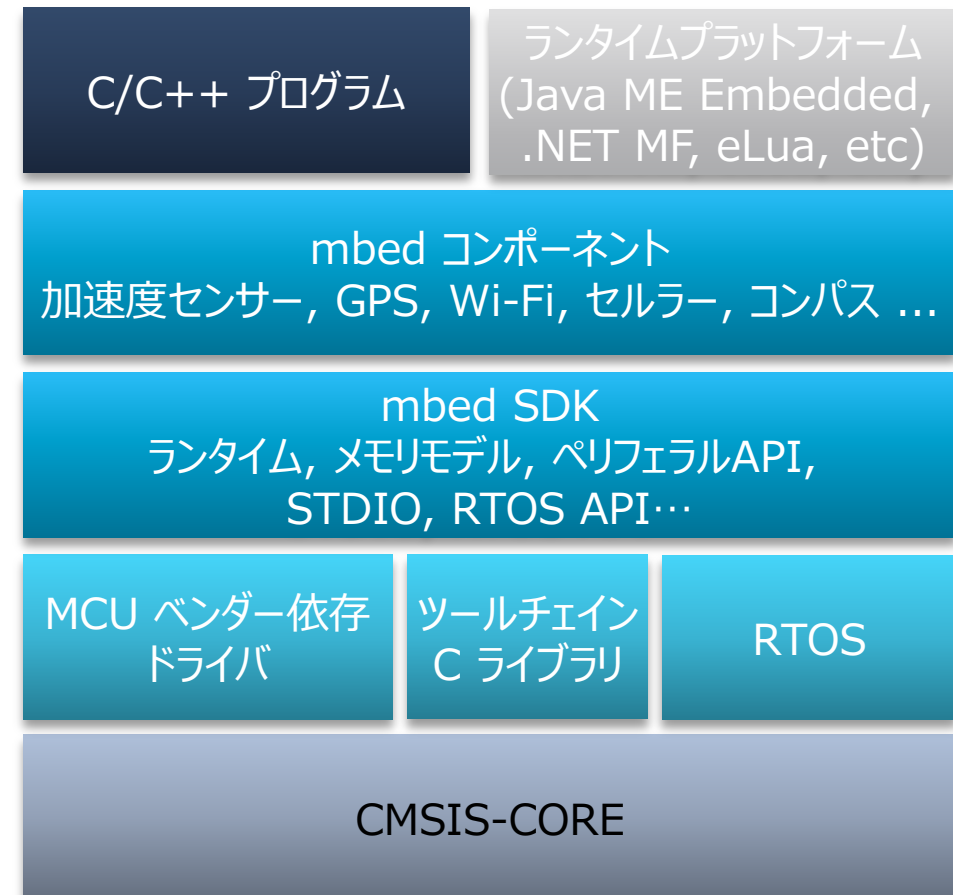
- High-level API と標準化された環境
- 異なるベンダーのデバイスでのポータビリティを確保
- 商用、非商用どちらでも使用可能
- オープンソース - Apache 2.0

## ■ 堅牢かつスケーラブル

- 専門チームによるメンテナンスとサポート
- 開発コミュニティによる貢献

## ■ 業界標準な開発手法

- メジャーなツールチェーンに対応
- CMSIS 準拠





# mbed HDK



- mbed プラットフォームを活用するためのハードウェアのレシピ
  - 回路図
  - 完全にオープンソースなファームウェア
  - ドラッグ & ドロップ・プログラミング
  - シリアルUSB変換
  - デバッガとの接続
- mbed 互換ボードやカスタムボードを開発可能
  - 開発環境、ライブラリの有効利用



# mbed-enabled プラットフォーム



- HDK を使用して mbed-enabled ハードウェアの開発が可能
- 100,000 台以上の mbed-enabled ボードが出荷済み
- 開発ボード、OEM モジュール、プログラマブル製品
- HDK によってプロトタイプからプロダクションに移行可能
- 31種類のプラットフォーム



mbed LPC1768



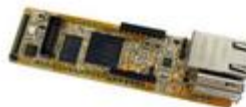
mbed LPC1114U24



Freescale KL25Z



NXP LPC800-MAX



EA LPC4088  
QuickStart Board



Seeeduino-Arch



LPC1114FN28



u-blox-C027

# mbed で何ができるのか

- **mbed SDK で提供されている基本API (C++クラスライブラリ)**
  - Digital I/O, Analog I/O, Network, Communication interface, Timer and Interrupt, File System, RTOS
- **mbed コミュニティが開発したライブラリの再利用**
  - USB, Display, Audio
  - 登録数は 2,000 以上

# API の例 (DigitalOut クラス)

## mbed - DigitalOut Class Reference

### Public Member Functions

	<u><a href="#">DigitalOut</a></u> (PinName pin) Create a <u><a href="#">DigitalOut</a></u> connected to the specified pin.
	<u><a href="#">DigitalOut</a></u> (PinName pin, int value) Create a <u><a href="#">DigitalOut</a></u> connected to the specified pin.
void	<u><a href="#">write</a></u> (int value) Set the output, specified as 0 or 1 (int)
int	<u><a href="#">read</a></u> () Return the output setting, represented as 0 or 1 (int)
<u><a href="#">DigitalOut</a></u> &	<u><a href="#">operator=</a></u> (int value) A shorthand for <u><a href="#">write()</a></u>
	<u><a href="#">operator int</a></u> () A shorthand for <u><a href="#">read()</a></u>

# DigitalOut クラスの使用例

```
main.cpp X
1 #include "mbed.h"
2
3 // DigitalOut クラスインスタンスの定義
4 DigitalOut myled(LED1);
5
6 int main() {
7     // operator= による書き込み処理 (write() と等価)
8     myled = 1;
9     myled.write(1);
10    // read() による読み込み処理と書き込み
11    myled = myled.read();
12    // operator int() で読み込んだ値を反転して、operator= で書き込み
13    myled = !myled;
14 }
15
```

# その他の代表的な API

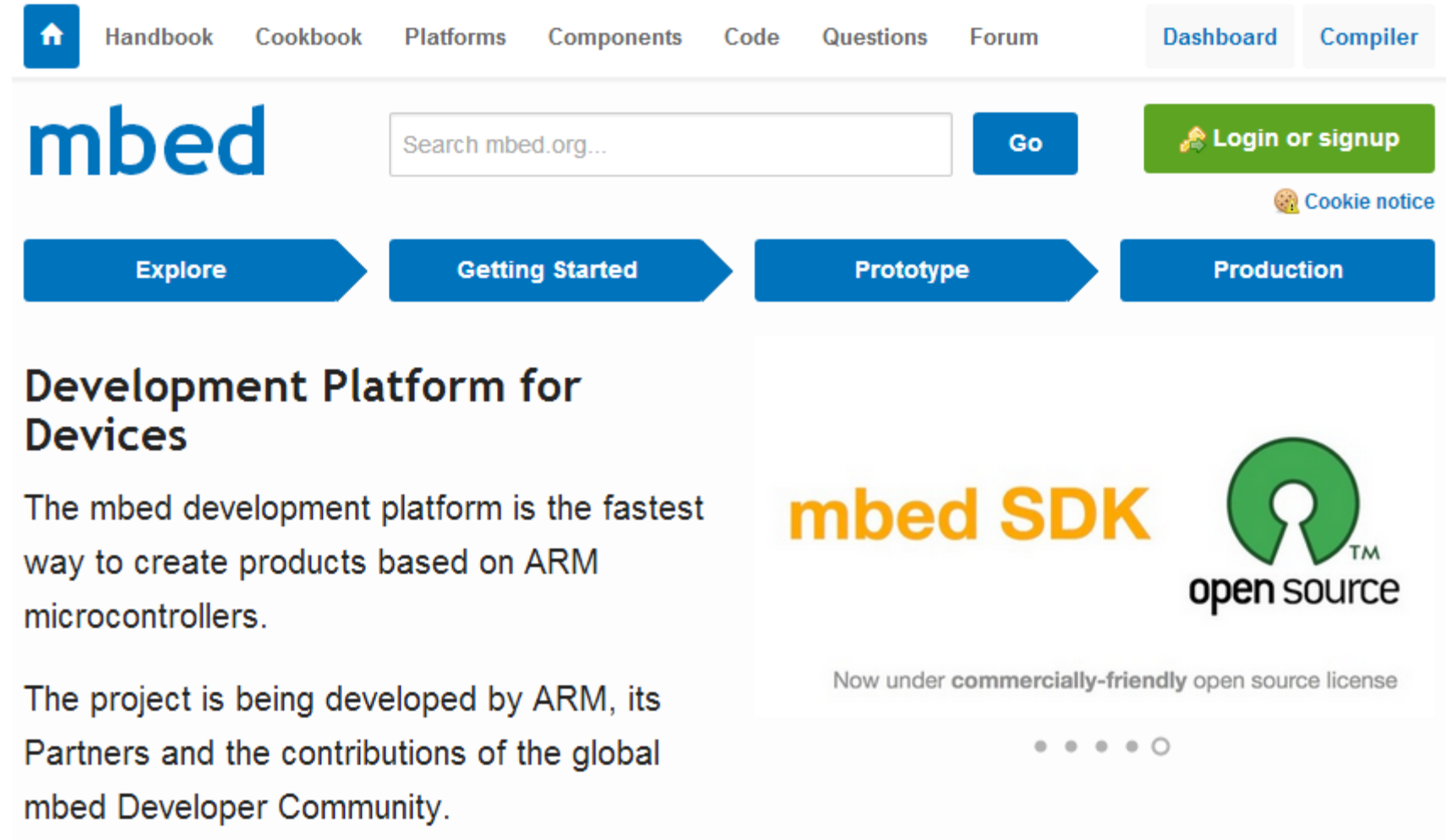
■ デジタル入力	<b>DigitalOut</b>
■ デジタル入出力	<b>DigitalInOut</b>
■ アナログ入力	AnalogIn
■ アナログ出力	AnalogOut
■ PWM出力	PwmOut
■ シリアルバス	Serial, SPI
■ I2C バス	I2C
■ 周期イベント	<b>Ticker</b>
■ USB	USBDevice, USBHost
■ 待ち時間処理	<b>wait</b>

# mbed.org ホームページ

- <http://mbed.org>

- 全ての情報を集約

- Handbook
- Cookbook
- Platforms
- Components
- Code
- Questions
- Forum



The screenshot shows the mbed.org homepage. At the top is a navigation bar with links: Handbook, Cookbook, Platforms, Components, Code, Questions, Forum, Dashboard, and Compiler. Below this is the mbed logo, a search bar with the text 'Search mbed.org...', a 'Go' button, and a 'Login or signup' button. A 'Cookie notice' link is also present. Below the navigation bar are four blue buttons: 'Explore', 'Getting Started', 'Prototype', and 'Production'. The main content area features the heading 'Development Platform for Devices' and a paragraph: 'The mbed development platform is the fastest way to create products based on ARM microcontrollers.' Below this is another paragraph: 'The project is being developed by ARM, its Partners and the contributions of the global mbed Developer Community.' To the right of the text is a large graphic for 'mbed SDK' with the ARM logo and the text 'open source'. Below the graphic is the text 'Now under commercially-friendly open source license' and a series of five dots, with the last one being filled.

# クラウド開発環境

## ■ オンライン IDE

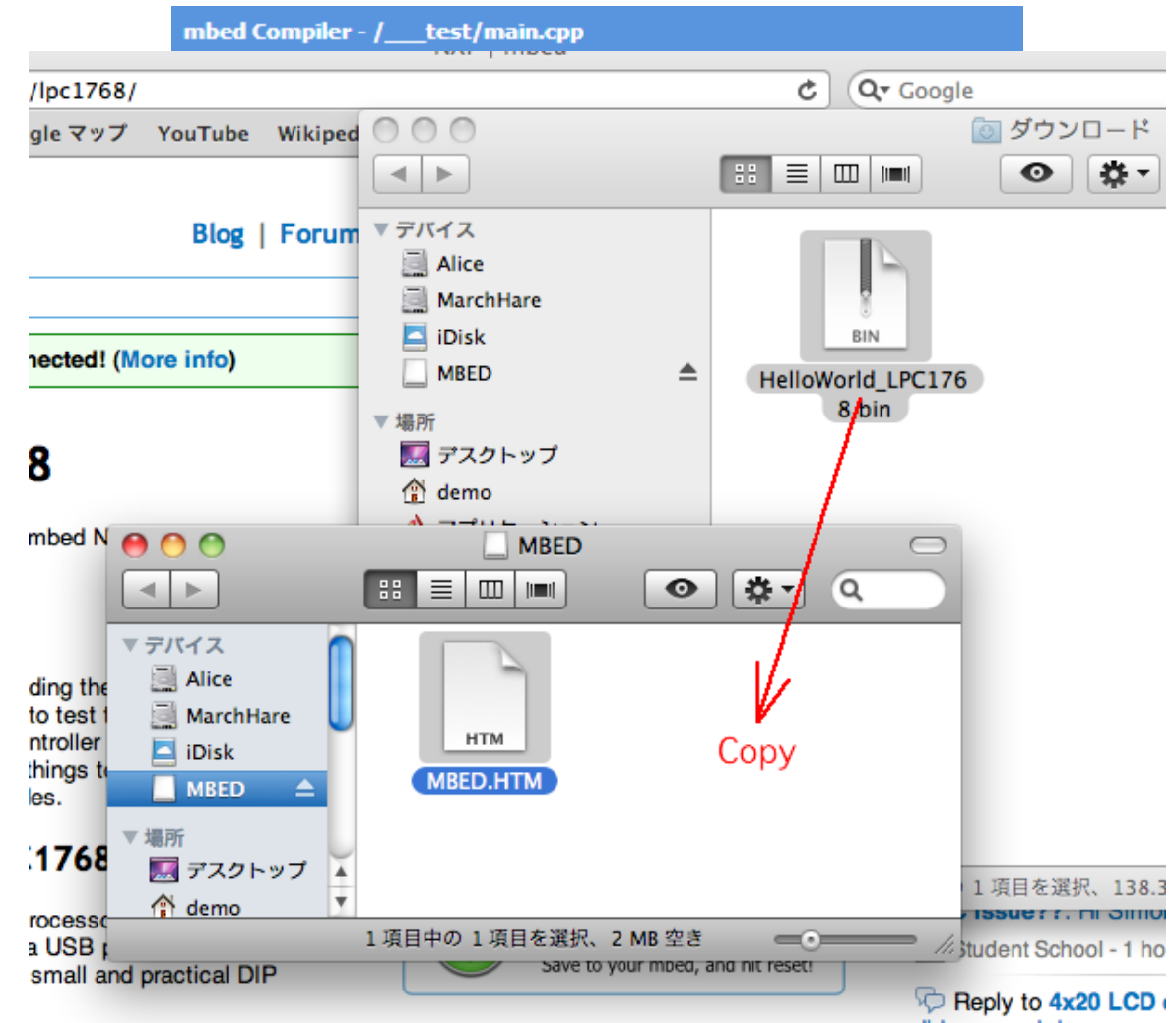
- プラットフォーム非依存の開発環境
- ブラウザベース

## ■ オンラインコンパイラ

- ARM純正の最適化コンパイラ
- [Compile] ボタンを押すと、生成されたバイナリがダウンロードされる

## ■ ターゲットボードへの書き込み

- USB のドライブにドラッグ & ドロップするだけ





# ソースコード管理

- コードレポジトリ
  - コラボレーション・ワークフロー
  - IDEに機能を統合
- オフライン環境にも対応済み

The screenshot shows the GitHub repository page for 'mbed official / mbed'. The page includes a navigation bar with links to Home, History, Graph, API Documentation, Wiki, Issues, and Pull Requests. A description states: 'The official mbed C/C++ SDK provides the software platform and libraries to build your applications.' The 'Revision graph' section displays a list of recent commits, including releases and bug fixes. On the right, the 'Repository actions' section shows 'Import latest build' and 'Following' buttons. The 'Repository details' section provides statistics: Type (Library), Created (15 5月 2012), Imports (224286), Forks (3), Commits (71), Dependents (4191), Dependencies (0), and Followers (362).

Users » mbed\_official » Code » mbed

mbed official / mbed Featured

Last commit 9 days ago

**Description:** The official mbed C/C++ SDK provides the software platform and libraries to build your applications.

[Home](#) [History](#) [Graph](#) [API Documentation](#) [Wiki](#) [Issues](#) [Pull Requests](#)

### Revision graph

70:673126e12c73	Release 70 of the mbed library	default tip	9 days ago , by <a href="#">Bogdan Marinescu</a>
69:4a7918f48478	Release 69 of the med library		9 days ago , by <a href="#">Emilio Monti</a>
68:f37f3b9c9f0b	Bug fixes and new features		23 10月 2013 , by <a href="#">Bogdan Marinescu</a>
67:a9913a65894f	New mbed build with various bugfixes and improvements on all platforms.		19 9月 2013 , by <a href="#">Bogdan Marinescu</a>
66:9c8f0e3462fb	New mbed library build with support for LPC1114.		19 8月 2013 , by <a href="#">Bogdan Marinescu</a>
65:5798e58a58b1	New target (LPC4088), new features (interrupt chaining), bug fixes (KL25Z I2C).		12 8月 2013 , by <a href="#">Bogdan Marinescu</a>
64:e3affc9e7238	New build system structure, new target (LPC1347), bug fixes (I2C read/write errors, LPC11U24 memory map and others)		05 8月 2013 , by <a href="#">Bogdan Marinescu</a>
63:b3110cd2dd17	spi slave and i2c slave support		08 5月 2013 , by <a href="#">Samuel Mokrani</a>
62:7e6c9f46b3bd	Add NXP LPC812		17 4月 2013 , by <a href="#">Emilio Monti</a>

**Repository actions**

[Import latest build](#)

[Following](#)

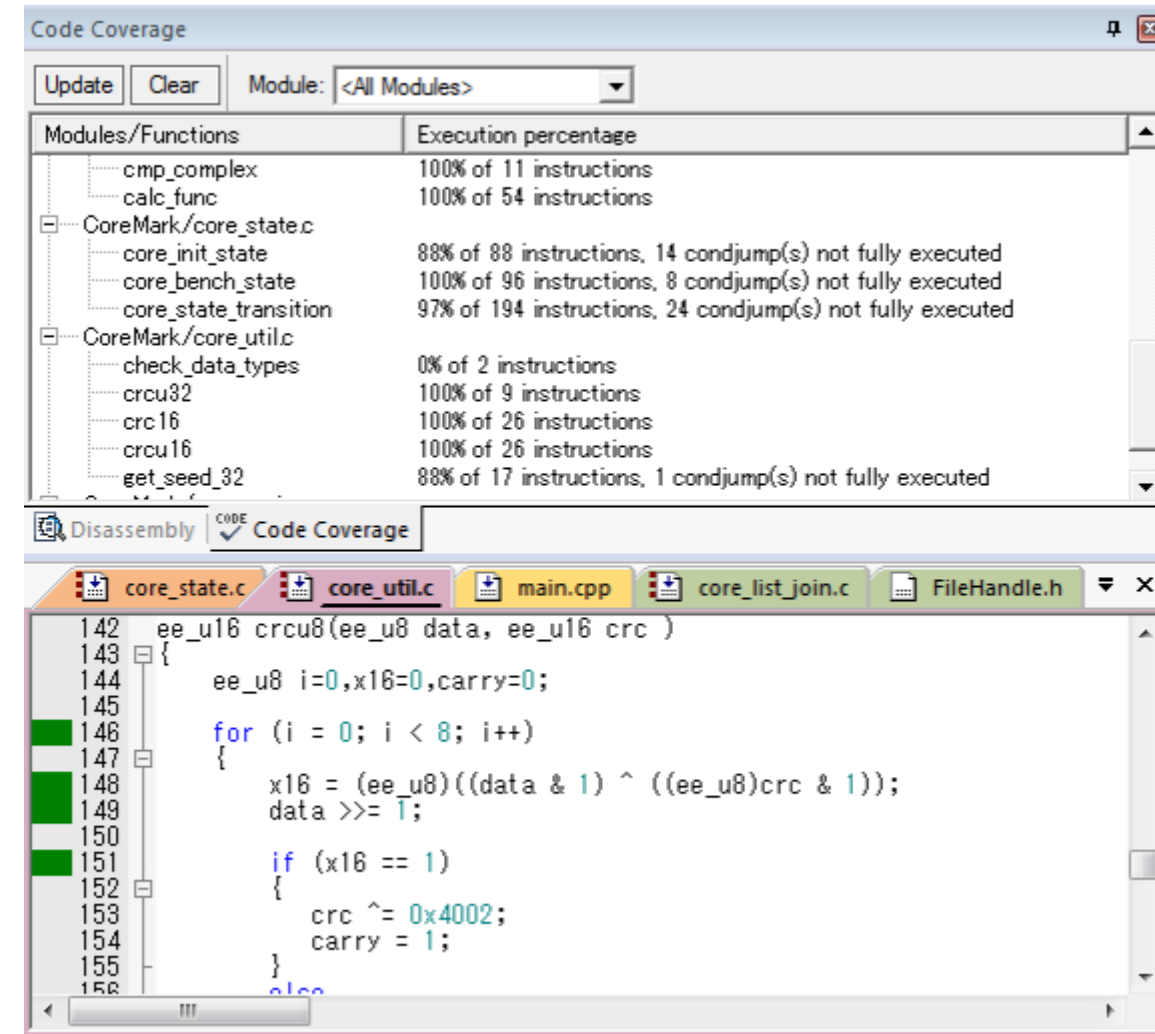
[Make featured again](#)

**Repository details**

Type:	Library
Created:	15 5月 2012
Imports:	224286
Forks:	3
Commits:	71
Dependents:	4191
Dependencies:	0
Followers:	362

# オフライン環境とデバッグ

- `printf()` で変数やメッセージなどをシリアルで出力
- プロジェクトをエクスポートして、オフライン環境で使用することも可能
  - Keil uVision4 (MDK-ARM), DS-5
  - NXP LPCXpresso IDE
  - IAR EWARM
- mbed で使用可能なデバッグ機能
  - C/C++ ソースレベルデバッグ、逆アセンブラ
  - フラッシュメモリへの書き込み
  - ハードウェアブレークポイント（4または8箇所）
  - ウォッチポイント（2または4箇所）
  - CPUレジスタ、ペリフェラルレジスタへのアクセス



# コンポーネント・ライブラリ

## ■ コンポーネント

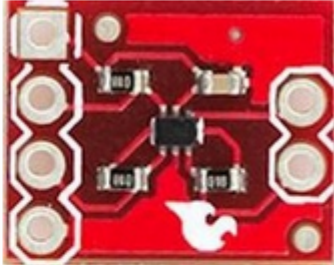
- アクチュエータ
- 通信モジュール
- ディスプレイ
- IoT サービス
- ロボティックス
- センサー
- ストレージ

Components » Sensors » Temperature » TMP102 Temperature Sensor

## TMP102 Temperature Sensor

**Datasheet**  
<http://www.ti.com/litv/pdf/sbos397b>

small SOT563 package, with a 0.0625C



**Notes**

**Pinout**

TMP102	mbed
1 - Vcc (square pad)	Vout
2 - SDA	p9
3 - SCL	p10
4 - Gnd	Gnd

Available on a breakout board from [Sparkfun](#)

[Ask a question](#)

[Import program](#)

n be imported independently

[Import library](#)

independent

# mbed の使用例



Disney Research社 Aireal: Interactive Tactile Experiences in Free Air

# mbed プロジェクトに参加するには

- プラットフォームパートナー
  - mbed-enabled プラットフォーム
- コンポーネントパートナー
  - センサー、通信、ソフトウェア、ツール
  - オープンソース・ライブラリへの移植
- 開発コミュニティ
  - 参加、協業、共有
  - グローバルコミュニティによる開発

mbed

Search mbed.org...

Go

Handbook » Contributing

## Contributing

Here are the various ways you, your company or organisation can begin to contribute to and engage with mbed and the mbed community.

### Write a library for your hardware component or service

Adding your library to mbed will allow users to quickly integrate your product into their product. It also provides a place where users can ask questions about your product or post any issues they have with your library.

The API of a library can be documented using doxygen syntax, this will be displayed on the library's page on the website and is also visible with mbed's online IDE, this makes developing with your library even easier for the mbed community. See [Writing a library](#)

### Write a hello world example for your product

A hello world is a concise program showing how a library is used; it should be easy to read and contain comments where appropriate. It doesn't need to demonstrate every feature of a library but should give the reader a feel for how it works.

### Add your products to the components database

The [Components database](#) is a place where the hardware and software components that make up an mbed based project can live. You should consider adding a component for your product. Currently components are maintained by the community but in the future we will allow partners to own their components so they can maintain them themselves.

Edit this page

#### Table of Contents

1. Write a library for your hardware component or service
2. Write a hello world example for your product
3. Add your products to the components database
4. Answer (and ask) questions
5. Become a partner



# まとめ

- mbed は、IoT の “Things” を開発するための優れたプラットフォーム
- [mbed.org](https://mbed.org) への情報の集約化
- SDK, HDK はオープンソース化され、誰でも開発が可能
- クラウド開発環境による効率的な開発
- 共通のAPIが使用可能な、様々な Cortex-M MCU をサポート
- 検証済みのコンポーネントライブラリの活用