

クラウドを利用した組み込みマイコン活用技術

〈ESP8266 - RX621〉

高度ポリテクセンター

クラウドを利用した 組み込みマイコン活用技術

高度ポリテクセンター

1

目次

1. クラウド・IoT
2. TCP/IPネットワーク
3. Wi-Fiマイコン
4. Arduinoマイコン開発環境
5. Wi-Fiマイコンプログラム実習
 - 5-1. GPIO
 - 5-2. シリアル通信
 - 5-3. A/D変換
 - 5-4. I2C通信(BME280センサ)
 - 5-5. Wi-Fi通信
 - 5-6. クラウドAmbient
6. RXマイコンWi-Fi化
 - 6-1. RXマイコン基板
 - 6-2. RXマイコン開発環境
 - 6-3. RXマイコンプログラム実習
 - 6-4. RXマイコンWi-Fi通信実習
7. MQTT通信
 - 7-1. MQTTプロトコル
 - 7-2. MQTT通信mosquitto
 - 7-3. Wi-FiマイコンのMQTT通信
 - 7-4. クラウドbeebotte

2

訓練目標

クラウド、IoTの概要と活用事例について理解する

Wi-Fiマイコンで各種プログラムを作成できる

(GPIO、ADC、I2C、TCPネットワークなど)

Wi-FiマイコンでクラウドサービスAmbientを活用できる

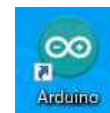
既存の組み込みシステムをWi-Fiマイコンで無線化できる

Wi-Fiマイコンを使用してクラウドによるMQTT通信ができる

3

実習ソフト

Arduino 2.3.2: Wi-Fiマイコンプログラム開発環境
(シリアルモニタ含む)



CS+ v8.04 (CC-RX v3.02): RXマイコンプログラム開発環境



SocketDebuggerFree Ver2.0: パケットモニタソフト



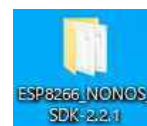
Wireshark Ver4.2.3: パケットキャプチャソフト



FlashDownloadTool Ver3.8.5: ESP8266フラッシュ書込み



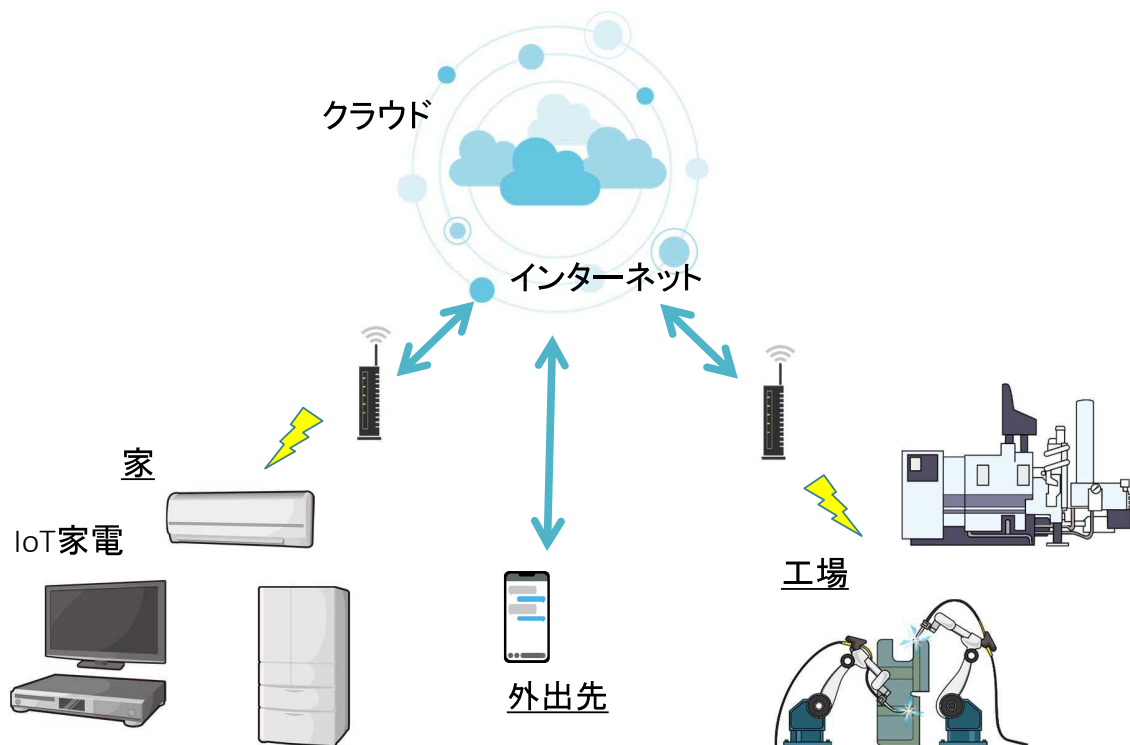
ESP8266_NONOS_SDK-2.2.1: ESP8266ファームウェア



mosquitto Ver2.0.18: MQTT通信ツール
Broker, Subscriber, Publisher

4

1. クラウド・IoT



5

クラウド・IoT導入メリット

クラウドサービスを利用することによって、ネットワーク経由でデータやソフトウェアを利用できる



システム構築や管理などの手間を省くことで業務の効率化・コストダウンが図ることができる

モノをインターネットに接続して、各種サービスを利用できる



家のエアコンを、スマホを使って外出先から制御できる
工場内の生産状況を、PCなどでどこからでも確認できる

6

クラウドサービスの分類

SaaS (Software as a Service)

ソフトウェア、アプリケーションの機能をネットワーク経由で利用するモデル

Gmail、Dropboxなど

PaaS (Platform as a Service)

アプリケーション開発に必要な実行環境を利用するモデル

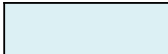
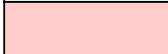
AWS, Azure, Herokuなど

IaaS (Infrastructure as a Service)

CPU、メモリ、ストレージなどのコンピュータリソースを提供するモデル

AWS, Azureなど

SaaS	PaaS	IaaS
データ	データ	データ
アプリ	アプリ	アプリ
ミドルウェア	ミドルウェア	ミドルウェア
OS	OS	OS
ハードウェア	ハードウェア	ハードウェア
ネットワーク	ネットワーク	ネットワーク

	ユーザの範囲
	クラウド事業者の範囲

7

クラウドサービス

AWS IoT

Amazon Web ServicesのIoT向けサービス

マイコン向けRTOSのFreeRTOSと親和性が高い

Azure IoT

Microsoft が提供するIoT向けクラウドサービス

Ambient

個人運営: 下島さん

IoT機器から送られてくるデータをWebブラウザ上でグラフ化できるサービス

Beebotte

クラウドMQTT、Publish-Subscribe通信モデル

8

2. TCP/IPネットワーク

コンピュータネットワーク

ネットワーク分類

プロトコル

TCP/IP

Ethernet

IPアドレス

UDP, TCP

ネットワークコマンド

13

コンピュータネットワーク

PCや組み込み機器など様々な装置をネットワークに接続することによって、付加価値が得られる

ネットワークシステム例

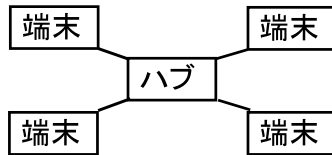
ネットワークシステム	概要
C/Sシステム	クライアント(リソースの利用)／サーバ(リソースの集中管理)型システム
Webシステム	インターネット技術の3階層アーキテクチャを活用したシステム
インターネット	世界中のネットワークが相互接続しており、ウェブや電子メール等のサービスが実現
クラウドコンピューティング	ネットワーク上のサーバ、ストレージ、サービスなどを利用できる環境

14

ネットワークのトポロジ別分類

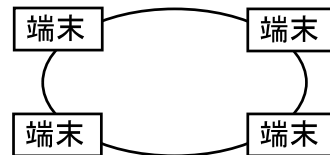
スター型

Ethernet (100BASE-TX) ...



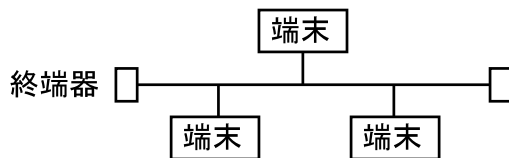
リング型

トークンリング、FDDI ...



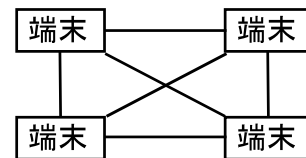
バス型

Ethernet (10BASE-2) ...



メッシュ型

インターネット、Zigbee ...



15

ネットワークのエリア別分類

LAN (Local Area Network)

オフィス、ビル、学校などの構内や家庭内での自営ネットワーク

LAN構築には、資格・免許が必要ない

要求に合わせて自由に構築できる

WAN (Wide Area Network)

複数のLANを遠隔で結んだネットワーク

通信エリアは、国内各所あるいは海外まで広がる

通常、通信事業者等が提供するアクセス回線や専用回線を借りて運用する

PAN (Personal Area Network)

個人が周辺機器などを扱う範囲でのネットワーク

通信距離は、数m程度の範囲

代表的なインターフェース: USB、IEEE1394など

センサネットワークWPAN: Bluetooth、Zigbeeなど



16

3. Wi-Fiマイコン

ESP-WROOM-02

Wi-Fiマイコン基板

ATコマンド

43

ESP-WROOM-02

Espressif System社(中国・上海)が開発したWi-Fiマイコン
無線LANチップESP8266EX(MCU Tensilica L106が内蔵)

ESPモジュール(ESP-WROOM-02)は、
ESP8266EX+メモリ+アンテナ



特徴

低価格

国内の電波法に基づいた認証を取得済み

インターネットアクセスのプロトコルスタックが実装済み

44

Wi-Fi搭載マイコンESP8266

項目	内容
Wi-Fi	802.11 b/g/n 2400~2483.5MHz
MCU	32bit Tensilica L106 (Cadence)
クロック	80MHz(最大160MHz)
RAM	36kB(最大)
インターフェース	GPIO, 10bitADC, I2C, SPI
電源電圧	3.0~3.6V
消費電流	170mA(送信時)、80mA(平均)
待機電流	60uA(RTC使用ディープスリープ時)
サイズ	5mm X 5mm

45

ESP-WROOM-02仕様

項目	内容
構成	ESP8266EX+フラッシュメモリ+アンテナ
フラッシュメモリ	4MB または 2MB(最大16MB)
水晶振動子	40MHz または 26MHz, 24MHz
電源電圧	3.0~3.6V
サイズ	18mm X 20mm X 3mm

使用方法

- ・ATコマンドを実行して動作させる
- ・Arduinoマイコンとしてプログラム書込みして動作させる

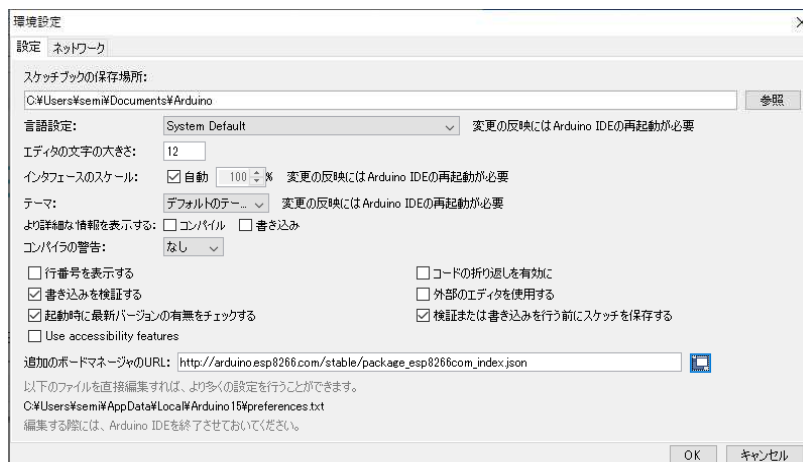
46

4. Arduinoマイコン開発環境

ArduinoIDE2.3.2

ファイル→環境設定
追加のボードマネージャのURL

`http://arduino.esp8266.com/stable/package_esp8266com_index.json`



71

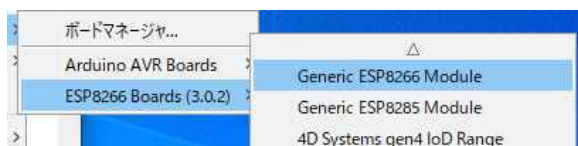
ArduinoIDEボードの追加

ツール→ボード→ボードマネージャ
ESP8266 3.0 インストール



72

ボードの選択

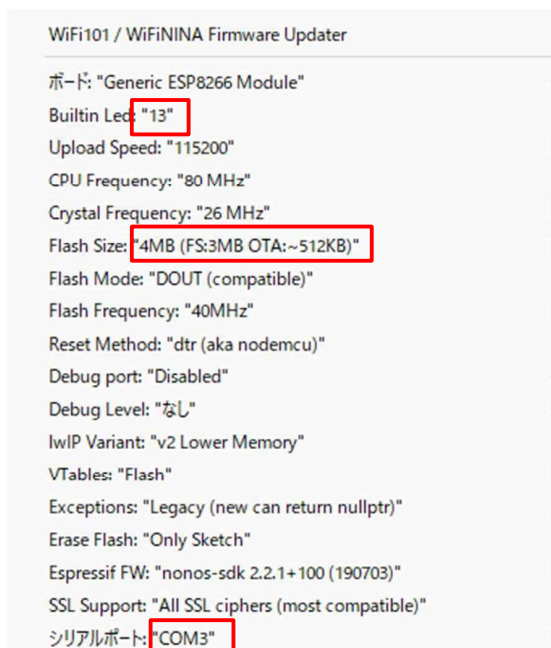


Generic ESP8266 Module
CPU Frequency: 80MHz
Crystal Frequency: 26MHz
Flash Size: 4MB(FS:3MB...)
Flash Frequency: 40MHz

...

シリアルポート: COMx
(USB-シリアル変換モジュール)

※デバイスマネージャーで確認できる



73

LEDカププログラムの作成

ファイル→スケッチ例→01.Basics→Blink

ファイル→名前付けて保存

Blink01

ツール→Builtin Led: 13

```
Blink01 | Arduino 1.8.13
ファイル 編集 スケッチ ツール ヘルプ

Blink01
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

74

5. Wi-Fiマイコンプログラム実習

- 5-1. GPIO
- 5-2. シリアル通信
- 5-3. A/D変換
- 5-4. I2C通信(BME280センサ)
- 5-5. Wi-Fi通信
- 5-6. Ambient

※Arduinoの[ファイル]→[スケッチ例]を元に作成したプログラム

77

5-1. GPIO

General Purpose Input Output (汎用I/Oポート)

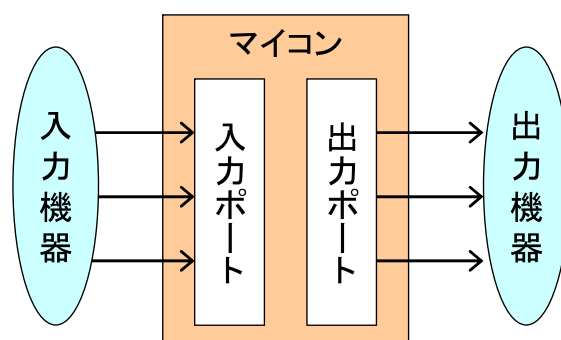
使用する前に、ポートを入力か出力に設定する

入力ポートはピンの入力電圧値で、HIGH(3.3V)かLOW(0V)を読み込む

出力ポートはピンにHIGHかLOWを書き込むことで電圧を出力する

ArduinoのGPIOは、ピンごとに処理する

機器によって、使用するピン数が異なる



78

デジタル入出力関数

pinMode(pin, mode)

ピンの動作を入力か出力に設定

digitalWrite(pin, value)

HIGHかLOWを、指定したピンに出力

digitalRead(pin)

指定したピンの値を読み取る(HIGHかLOW)

79

GPIO制御プログラム

Button01: タクトSWのON/OFFでLEDが点灯/消灯

```
1: const int ledPin = 13;
2: const int buttonPin = 12;
3: int buttonState = 0;
4:
5: void setup() {
6:   pinMode(ledPin, OUTPUT);
7:   pinMode(buttonPin, INPUT);
8: }
9:
10: void loop() {
11:   buttonState = digitalRead(buttonPin);
12:
13:   if (buttonState == LOW) {      // pushbutton is pressed
14:     digitalWrite(ledPin, HIGH); // turn LED on:
15:   }
16:   else {
17:     digitalWrite(ledPin, LOW);  // turn LED off:
18:   }
19: }
```

※[スケッチ例]→[02.Digital]→[Button]

80

5-2. シリアル通信

1本あるいは2本の線で、データをシリアル(直列)に転送する方法

シリアル通信	内容
RS-232C	モデム用のシリアルインターフェース規格 一昔前はPCに標準でシリアルポートが備えられていた
UART	非同期式のシリアル信号変換回路(マイコン内蔵) RS-232C信号レベル変換IC、シリアル-USB変換モジュールを使用してデータ通信することが多い
SPI	モトローラ社が提唱したシリアル通信規格 コンピュータ内部のデバイス間で使用される SCK, MISO, MOSI, SSの4つの信号線、シリアルの中では高速
I2C	フィリップス社が開発したシリアル通信規格 比較的低速なデバイス間で使用される SCL, SDAの2本の信号線、I2C対応デバイスが豊富

83

通信方式

同期式

送信側ではデータ中に同期をとるためのクロック信号を組み込んで送信し、受信側では受け取ったデータからクロック信号を分離することができる方式

非同期式(調歩同期式)

同期をとるためのクロック信号を組み込まずに、データの1文字で同期をとる方式

データの先頭と末尾にスタートビットとストップビットをつけて送信し、受信側ではスタートビットを合図に同期をとる

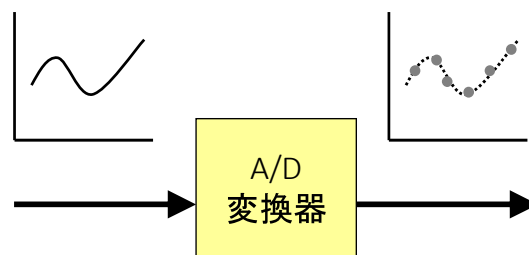
84

5-3. A/D変換

A/D変換器(ADC)
アナログ信号をデジタル信号に変換する機器

サンプリング(標本化)
一定間隔ごとの信号の大きさを調べて、取り出す
1秒間あたりのデータ数をサンプリング周波数で表す

量子化
サンプリングで得られたデータを、定められた段階に近似する
任意の値と量子化レベルとの誤差が発生する(量子化誤差)



入力電圧 (mV)	出力信号 (10bit)
0	00,0000,0000
3.2258	00,0000,0001
6.4516	00,0000,0011
...	...
3300	11,1111,1111

93

ESP8266のAD変換器

分解能: 10bit
(変換後データ: 0~1023)

入力電圧範囲: 0~1V
(実測で1Vより小さい電圧値で1024に)

AD入力ピン: TOUT(14番ピン)

接続機器: ボリューム10kΩ
(抵抗入れて、0~1Vに)

温度センサLM61
(0°C→600mV, 40°C→1V)

※課題

ADC1bit当たりの電圧値はどれくらいか？

94

5-4. I2C通信

2線式双方向バス

バスライン2本: SDA(シリアルデータ)、SCL(シリアルクロック)

マスタ/スレーブの関係で通信

固有アドレスにより、各デバイスを指定

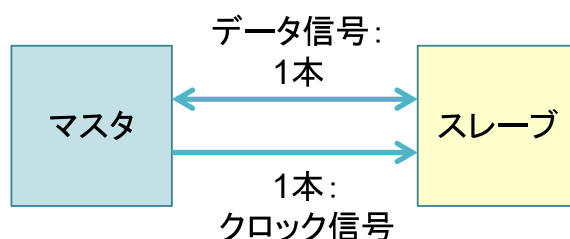
I2Cバス対応デバイスはインターフェースを搭載しているため、I2Cバスと直接相互通信できる

8ビットシリアル転送モード

Standard-mode: 100kbps

Fast-mode: 400kbps

Fast-mode Plus: 1Mbps



99

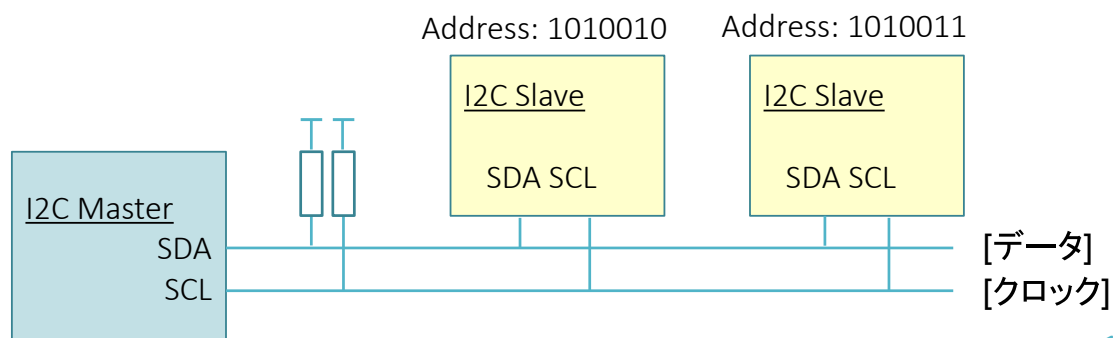
I2C接続方法

スレーブは複数接続することができ、デバイスの固有アドレスにより、区別する

プルアップ抵抗が必要である

マスタ: 同期用クロックの送信、データの読み込み・書き込みの指示

スレーブ: マスタの指示に従って動作する



100

5-5. Wi-Fi



無線でLANに接続する技術

PC、スマホ、テレビやレコーダー等のWi-Fi対応家電などがワイヤレスでつながり、ルーターを介してインターネットにも接続できる

IEEE802.11: 無線LAN(ワイヤレスLAN)の国際標準規格

Wi-Fi: IEEE802.11を使用する無線LANに関する登録商標

ESP8266: 2.4GHz帯の802.11 b/g/nに対応している

年	世代(名称)	規格名	最大通信速度	周波数帯域
2019年	第6世代(Wi-Fi6)	IEEE802.11ax	9.6Gbps	2.4GHz/5GHz
2013年	第5世代(Wi-Fi5)	IEEE802.11ac	6.9Gbps	5GHz
2009年	第4世代(Wi-Fi4)	IEEE802.11n	600Mbps	2.4GHz/5GHz
2003年	第3世代	IEEE802.11g	54Mbps	2.4GHz
1999年	第2世代	IEEE802.11a	54Mbps	5GHz
		IEEE802.11b	11Mbps	2.4GHz

115

Wi-Fi通信プログラム

Wi-Fi通信全般: ESPマイコンが実習室内のWi-Fiに接続する

WiFiClientXX: TCPクライアントとして動作する

UdpXX: UDPでデータ通信する

WiFiTelnetToSerialXX: Telnetサーバとして動作する

WiFiManualWebServer: Webサーバとして動作する

116

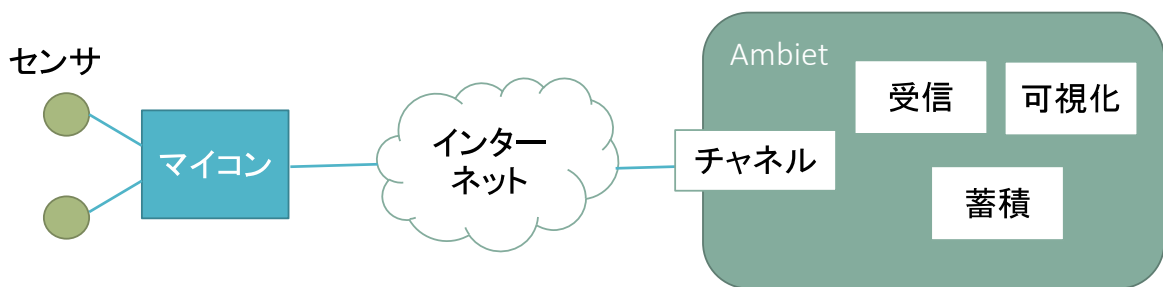
5-6. Ambient

IoTデータ可視化サービス <https://ambidata.io/>

マイコンなどから送られるセンサデータを受信し、蓄積し、可視化(グラフ化)できる

チャンネル数:8ch、可視化: 無償で使用できる

16ch以上、データ監視・通知: 有償になる



Ambient

ユーザ登録: メールアドレス用意

チャンネル作成・設定:
マイコンから送信されるデータ
温度、湿度

チャート作成・設定:
チャート(グラフ)の軸・データ範囲

マイコンプログラミング:
Ambientライブラリの使用
チャンネルID、ライトキー



6. RXマイコンWi-Fi化

6-1. RXマイコン基板

6-2. RXマイコン開発環境

6-3. RXマイコンプログラム実習

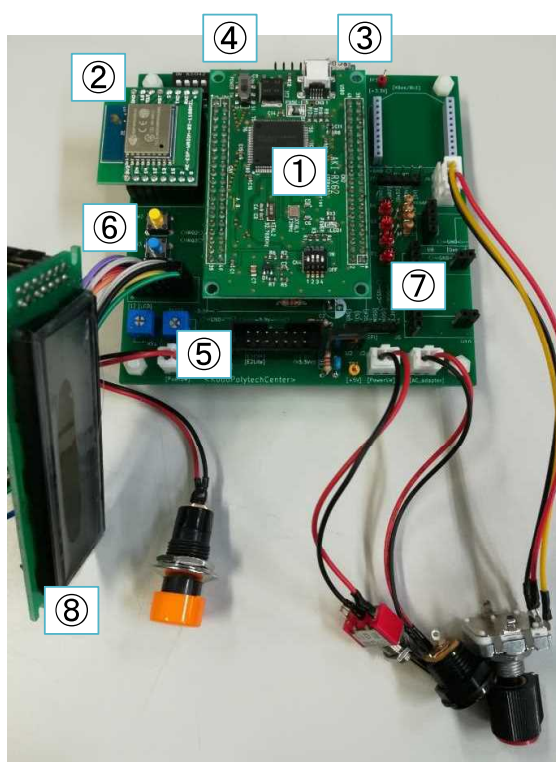
6-4. RXマイコンWi-Fi通信実習

※ Wi-Fi機能がないRX621マイコンに、ESP-WROOM-02を接続してWi-Fi化を実現する

※ RX621マイコンのセンサデータをクラウドAmbientにアップできるようにする

169

6-1. RXマイコン基板



- ①RX621マイコンボード
- ②ESP-WROOM-02
- ③USB-シリアル変換モジュール
- ④温度センサ
- ⑤ボリューム
- ⑥タクトSW
- ⑦LED
- ⑧キャラクタLCD

170

RXマイコンーESP8266接続

ESP8266のリセット・イネーブル信号は、
RX621のポートEで制御

ESP8266のシリアル送受信は、
RX621のSCI0に接続

RX621のSCI0を通じて、ESP8266
のWi-Fi通信処理を実行していく

RXマイコン	ESP8266
PE0	RST
PE1	EN
RxD0	TXD
TxD0	RXD

173

6-2. RXマイコン開発環境

統合開発環境:

CS+ for CC V8.07

RX C コンパイラ:

CC-RX V3.04

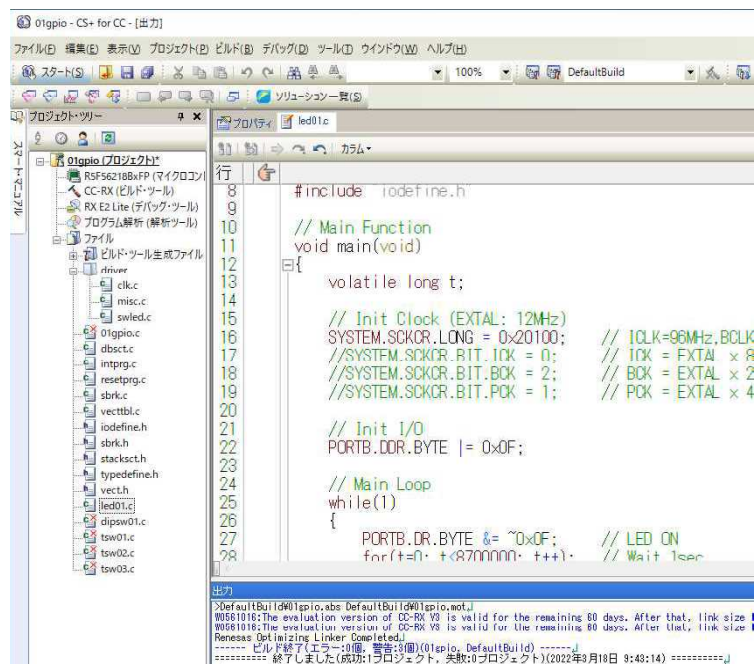
プロジェクト:

実習ごとに切替え

ソースファイル:

xxxN.c: メイン処理

intprg.c: 割り込み処理



174

6-3. RXマイコンプログラム実習

RX621の基本機能であるI/O制御、A/D変換、シリアル通信をプログラム実行しながら確認する

プロジェクト： 処理内容

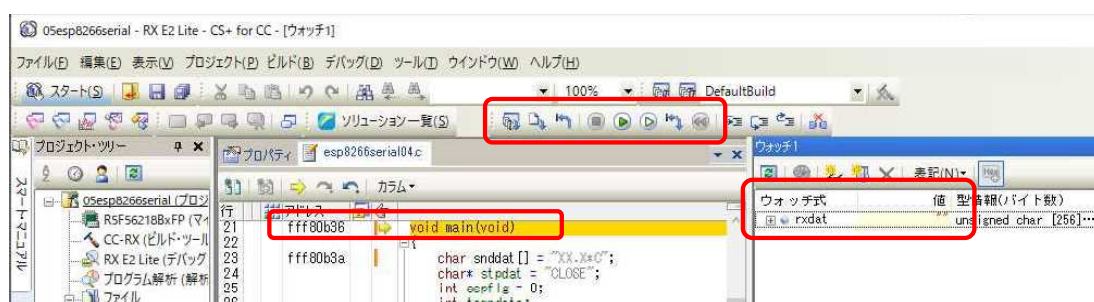
- 01gpio: I/O制御によるスイッチ、LEDの制御
- 02lcd: I/O制御によるキャラクタLCDの制御
- 03adc: A/D変換によるボリューム・温度センサのデータ取得
- 04sci: SCIによるPCとのシリアル通信、受信割り込み

175

RXマイコンプログラム実習手順

1. プロジェクト開く
2. ソースファイルを開き、コードを確認・編集する
3. ビルド&デバッグツールへダウンロードする
4. デバッグ操作(ステップ命令、変数参照など)、実行を行い動作確認する

メイン処理定義のxxxN.c、割り込み処理の定義のintprg.cをビルド対象にする



176

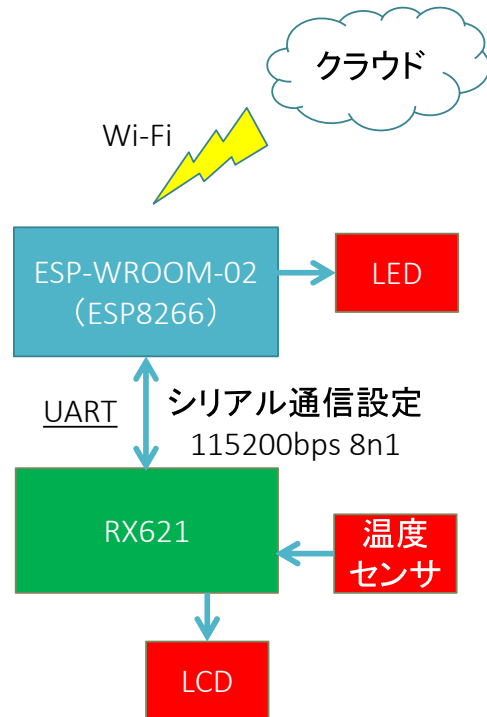
6-4. RXマイコンWi-Fi通信実習

・ESP8266での処理

- ② 設定済みのWi-Fiに接続する
- ④ RX621からのデータをシリアル受信して、Wi-Fi通信処理・クラウドへのアップ処理を行う

・RX621での処理

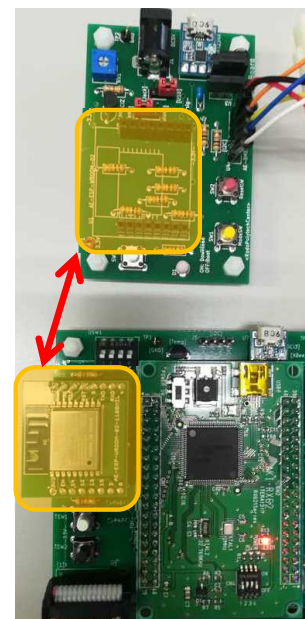
- ① ESP8266をイネーブルにした後、リセットする
- ③ SCI0でESP8266にデータを送り、Wi-Fi通信処理を任せる



181

RXマイコンWi-Fi通信実習手順

1. ESP8266プログラム作成:
ArduinoIDEでコーディング
2. ESP8266プログラム書込み:
ESP-WROOM-02 BaseBoardを用いて
Arduinoでマイコン書込み
3. ESP8266プログラム実行確認:
Arduinoシリアルモニタのデータ入出力
からESP8266単体で動作確認
4. RX621プログラム作成
CS+でコーディング
5. RX621・ESP8266の実行確認
ESP-WROOM-02をRXマイコン基板に付け替えて実行



182

7. MQTT



Message Queue Telemetry Transport

IBM社とEurotech社のメンバーで考案されたTCP/IPによるPub/Sub型のデータ配信プロトコル

IoTやロボット制御、クラウドサービスなどで軽量なデータを頻繁にやり取りするのに適したプロトコル

203

7-1. MQTT仕様



OASISという標準化団体が管理している

仕様は公開されており、使用許諾範囲内であれば自由に利用できる

MQTT Version 3.1.1 (2015年11月)

MQTT Version 5.0 (2019年3月)

204

7-2. MQTTブローカー



mosquitto

MQTTのオープンソース実装(OSS)

MQTT Version3.1.1のブローカー機能

UbuntuなどのLinuxやWindowsでも動作可能

MQTTクラウドサービス

Beebotte

Heroku CloudMQTT

211

mosquitto

ダウンロード(バイナリ)

<https://mosquitto.org/download/>

Windows版

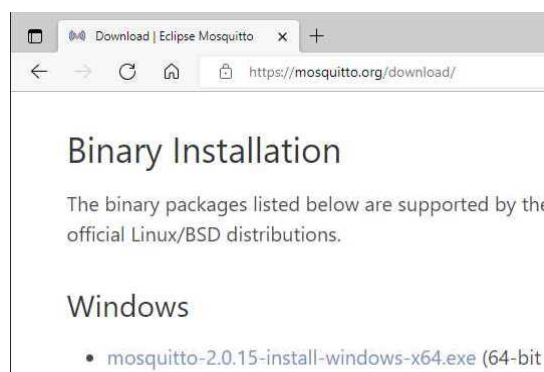
mosquitto-2.0.15-install-windows-x64.exe

インストール

デフォルトの場合

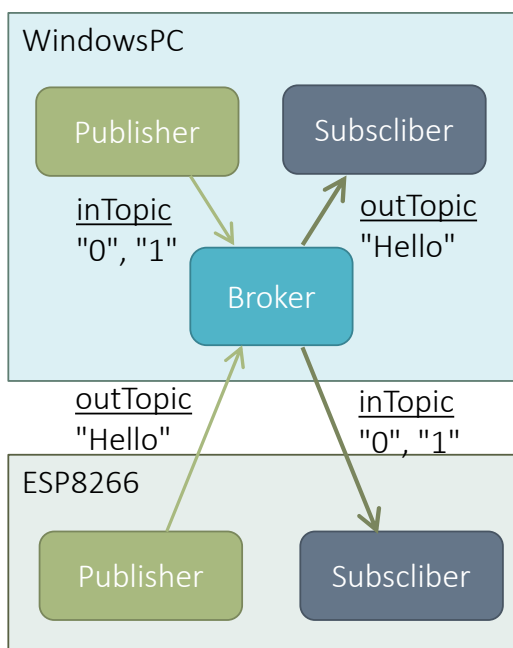
C:\Program File\mosquitto

にインストール



212

7-3. Windows-ESP8266MQTT通信



WindowsPC: コマンドプロンプト

```
mosquitto  
mosquitto_pub  
mosquitto_sub
```

ESP8266: Arduino

MQTT通信プログラム



221

ESP8266MQTT通信プログラム

初期化処理

Wi-Fi接続設定、ピン出力設定
MQTTサーバ接続・コールバック設定

Publisher動作(2秒間隔で送信)

トピック: outTopic

送信メッセージ: "hello world" + "#[カウント数]"

Subscriber動作

トピック: inTopic

受信メッセージ: '1'以外 - LED消灯
'1' - LED点灯

222

7-4. beebotte

クラウドMQTTの一つ
(Platform as a Service)

AWS IoT Core

HEROKU(Cloud MQTT)

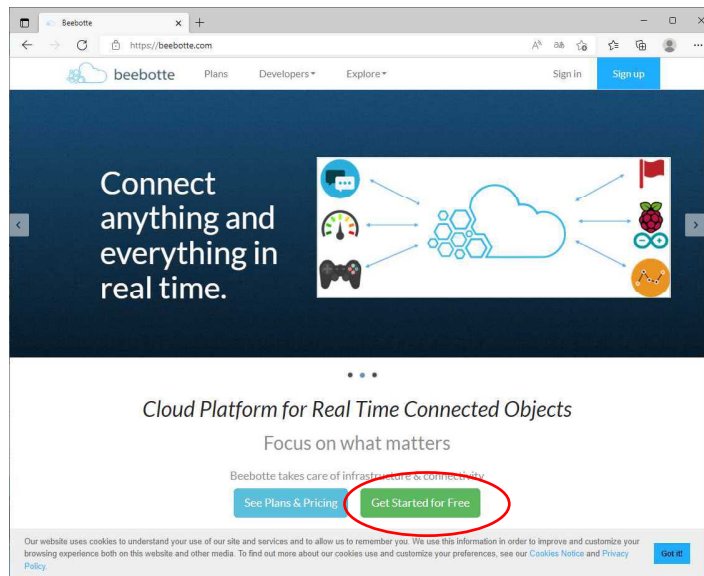
...

無料枠 (XS Free Plan)

チャンネル無制限

50000メッセージ/1日

...



https://beebotte.com

235

SIGN UP

Username

E-mail

Password

を入力して、アカウントを作成

236

参考Webサイト

garretlab Arduino
<https://garretlab.web.fc2.com/>

プチモンテ ESP-WROOM-02
https://www.petitmonte.com/robot/howto_esp_wroom_02.html

きむ茶工房ガレージハウス
<http://zattouka.net/GarageHouse/micon/linkPC/WiFi/WiFi1.htm>

Qiita: Arduino NanoとBME280を...
<https://qiita.com/nanase/items/f34e03c29410add9c4d0>

Beebotteの使い方メモ
<https://gist.github.com/yoggy/28196ba084f9c406c75967289fbb3dca>

249

参考資料・引用先Webサイト

トランジスタ技術2016 9月号
CQ出版

RX621 ユーザーズマニュアル ハードウェア編
ルネサスエレクトロニクス

Ambient
<https://ambidata.io/>

Eclipse Mosquitto
<https://mosquito.org/>

Beebotte
<https://beebotte.com/>

秋月電子通商
<https://akizukidenshi.com/>

250

■本テキスト掲載記事の利用に関するご注意

本テキストの著作権は、独立行政法人高齢・障害・求職者雇用支援機構 千葉支部 千葉職業能力開発促進センター 高度訓練センター（愛称：高度ポリテクセンター）にあります。

本テキストの一部または全部について、著作権法の定めのある範囲を超え、当方に無断で転載、複写、譲渡、販売、使用することを禁じます。

また、本テキスト記載の内容は、技術の進歩などにより予告なしに変更されることがあります。

■本テキスト記載の社名、製品名について

本テキスト記載の会社名と製品名などは、それぞれ当該各社の登録商標または商標です。それら団体名、商品名は本テキスト制作の目的のために記載されており、当方としては、その商標権を侵害する意思、目的はありません。

クラウドを利用した組みマイコン活用技術 [↑ESP8266 - RX621↓](#)