

特殊条件下において使用可能な温度測定 IoT モジュールの開発

Development of IoT module measuring temperature usable under particular conditions

技術開発部 生産・加工科 石澤満 柿崎正貴 鈴木健司
 応募企業 株式会社ミウラ

铸件温度測定を無線化及び自動化するため、無線通信機能付きマイコン等による自動温度測定プログラムを開発した。また、高温及び粉塵という特殊な環境に耐えられる筐体を設計し、マイコンや熱電対モジュールと組み合わせ、温度測定 IoT モジュールを開発した。更に、温度測定 IoT モジュールによる温度測定の評価を行うため、当所所有の電気マッフル炉 (FUW230PB) を用い、データロガー (GL240) の出力と比較した。

Key words: IoT、温度測定、Node-RED、Arduino IDE、MQTT

1. 緒言

応募企業は、低圧鋳造という方法で铸件を製造しており、アルミ鋳造時の铸件温度測定を熱電対とデータロガーで行っている。現状は、データロガーに熱電対を配線し、手でスイッチを押して測定しているが、これを自動化・無線化することを検討している。しかし、同企業は単独で開発を行うことは困難であった。

そこで、本事業で無線通信機能付きマイコン等による自動測定プログラムを開発した。また、高温及び粉塵という特殊な環境に耐えられる筐体を設計し、試作機を製作した。

2. 実験と結果

2. 1. システム構成について

2. 1. 1. 全体の構成

本事業で構築した温度測定 IoT システムの全体概略図を図1に示す。温度測定を行う IoT モジュール及び収集した温度データを可視化する IoT サーバで全体のシステムを構成した。温度測定 IoT モジュールで温度を自動測定し、IoT サーバに無線通信で温度データを送信する。次に、IoT サーバで受信した温度データを Node-RED¹⁾ と呼ばれるデータ可視化ツールで可視化し、CSV ファイルに温度データを書き込むシステム構成とした。

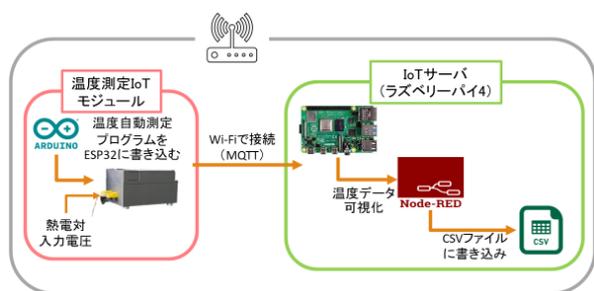


図1 温度測定 IoT システムの全体概略図

2. 1. 2. 温度測定 IoT モジュール

温度測定 IoT モジュールに用いた機材を表1に示す。4 c hで温度を自動測定し、IoT サーバに無線通信で温度データを送信するため、表1の機材を用いた。鋳型に温度測定 IoT モジュールを貼り付けるため、マグネットを使用した。また、鋳型表面温度をサーモグラフィカメラで測定したところ、約 40[°C]であったため、それに耐えられる樹脂及びバッテリーを選択した。

次に、温度測定に使用した熱電対モジュールのスペックを表2に示す。同企業からの要望があり、熱電対モジュールには、測定精度が±2[°C]以下のものを使用した。マイコン (ESP32) で I2C 通信の熱電対モジュールを 4 c h 温度測定する場合、熱電対モジュールに一定の抵抗を付加し、アドレスを変更する必要があるため、配線が煩雑になる。そこで、SPI 及び I2C の両通信方式の熱電対モジュールを 2 個ずつマイコンに接続し、4 c h で温度測定する構成とした。

また、ジャンパ線の空中配線部分に重畳するノイズの影響を軽減させるため、図2のとおり、リード線が宙に浮かぬようマイコン及び熱電対モジュールをユニバーサル基板に実装した。

図3に温度測定 IoT モジュールの構成を示す。図3の部品を組み合わせることで温度測定 IoT モジュールを試作した。

表1 温度測定 IoT モジュールに用いた機材

無線通信機能付きマイコン	ESP32-devkitC
ユニバーサル基板	AE-B2-TH
K型熱電対モジュール	MAX31855 ²⁾ × 2、 MCP9600 ³⁾ × 2
バッテリー	cheero CANVAS 3200mAh
マグネット	EA781EF-14
筐体	PLA フィラメント (3Dプリンタで製作)

表2 使用した熱電対モジュールのスペック

熱電対モジュール	MAX31855 × 2	MCP9600 × 2
マイコンとの通信方式	SPI	I2C
測定範囲	-200[°C]～1350[°C]	-200[°C]～1350[°C]
測定精度	±2[°C] (-200[°C]～+700[°C])	±1.5[°C]
最大分解能	0.25[°C]	0.0625[°C]

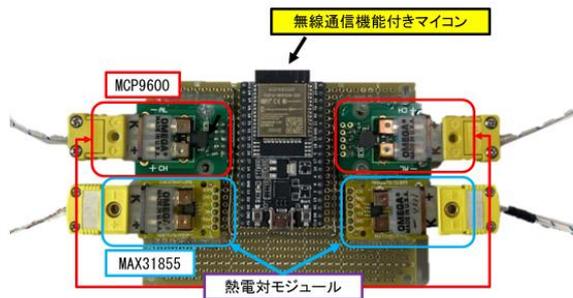


図2 マイコン及び熱電対モジュール



図3 温度測定 IoT モジュール

2. 1. 3. IoT サーバ

IoT サーバのシステム構成を表3に示す。IoT サーバには Raspberry Pi 4⁴⁾ を使用した。温度自動測定プログラムを開発するため、IoT サーバに Arduino IDE⁵⁾ を構築し、無線通信機能付きマイコンにプログラムを書き込んだ。温度自動測定プログラムの仕様は次のとおり。

- 4個の熱電対モジュールで温度を自動測定
 鋳型内部に最低でも4本の熱電対を組み込むため、4個の熱電対モジュールを使用する必要がある。
- ノイズによる影響を軽減するため、1秒間に5回温度を測定し、その中央値をIoTサーバに送信
 また、IoTサーバでNode-REDを使用し、受信した温度データの表示及び温度データをcsvファイルへ書き込むプログラムを開発した。

表3 IoT サーバのシステム構成

ハードウェア	Raspberry Pi 4 model B
OS	Raspberry Pi OS
CPU	Cortex-A72 (ARM v8) 64-ビット SoC @1.5GHz
メモリ	8GB
ストレージ	128GB (MicroSD カード)
プログラム開発環境	Arduino IDE
IoT モジュールとの通信方法	Wi-Fi (MQTT プロトコル ⁶⁾)
データ可視化ツール	Node-RED

2. 2. 温度測定実験

2. 2. 1. 実験準備

温度測定 IoT モジュールによる温度測定の評価を行うため、当所所有の電気マuffle炉 (FUW230PB) を用い、データロガー (GL240⁷⁾) との出力を比較した。

図4のように熱電対を配置し、マuffle炉の温度を実際に使用する温度 (500[°C]～700[°C]) に調節し、50[°C]ごとに温度を測定した。そして、温度測定 IoT モジュール及びデータロガーで測定した温度差が±2[°C]以下になるか否かを確認した。

次に、熱電対の接続方法を図5に示す。マuffle炉に複数の熱電対を同時に挿入することができなかったため、図5のように熱電対のコネクタ部分を各熱電対モジュール及びデータロガーに順番に付け替えながら測定した。

マuffle炉の熱電対



図4 温度測定実験の構成

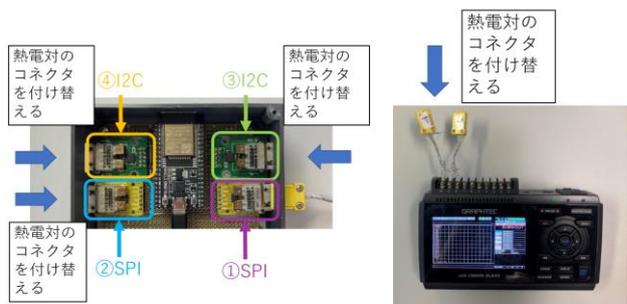


図5 熱電対の接続方法

2. 2. 2. 実験結果

測定温度ごとに9回温度を測定し、その平均値を表4に示す（小数点第2位四捨五入）。

表4より、I2C通信の熱電対モジュールでは、マッフル炉の設定温度に対し、データロガーとの温度差が $\pm 2[^\circ\text{C}]$ 以下におさまっていることが確認できた。また、SPI通信の熱電対モジュールでは $\pm 2[^\circ\text{C}]$ 以上の誤差があるが、データロガーとの温度差は+1%ほどにおさまった。

図6に、各熱電対モジュールの測定値とデータロガーの平均値との差を箱ひげ図で色分けして示す。横軸はマッフル炉の設定温度である。また、各設定温度における縦軸の値の平均値を線で結びグラフ上に示した。

図6より、SPI通信の熱電対モジュールは、温度が上昇するにつれ、データロガーとの温度差が大きくなるが、測定値のばらつきが小さいことが確認できた。I2C通信の熱電対モジュールは、550 $[^\circ\text{C}]$ で測定値にばらつきがあるが、データロガーとの温度差は $\pm 2[^\circ\text{C}]$ 以下であった。

表4 測定温度 (N=9)

データロガーとの温度差 $\pm 2[^\circ\text{C}]$ 以下：青、 $\pm 2[^\circ\text{C}]$ 以上：赤

マッフル炉	①SPI	②SPI	③I2C	④I2C	データロガー
500 $[^\circ\text{C}]$	500.1	498.7	498.7	497.6	497.6
550 $[^\circ\text{C}]$	550.2	550.4	547.4	546.7	548.0
600 $[^\circ\text{C}]$	602.6	601.7	597.3	597.3	598.0
650 $[^\circ\text{C}]$	652.7	653.0	647.5	647.4	647.7
700 $[^\circ\text{C}]$	703.6	703.4	696.7	696.6	697.2

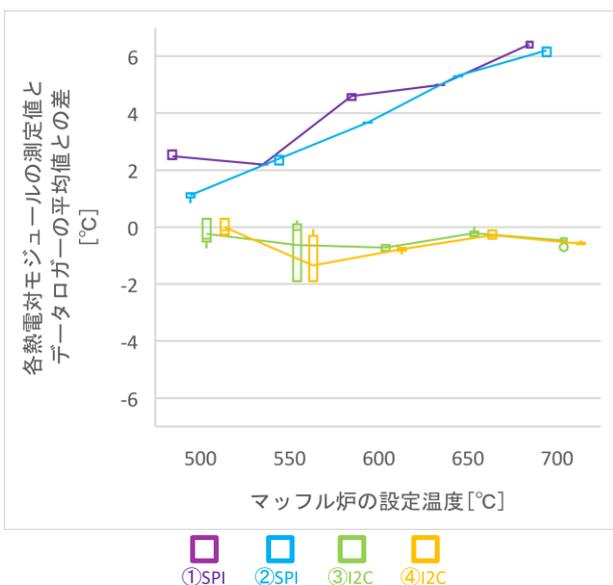


図6 各熱電対モジュール測定温度のばらつき (箱ひげ図)

2. 2. 3. 考察

SPI通信の熱電対モジュールがデータロガーとの温度差 $\pm 2[^\circ\text{C}]$ 以下におさまらなかった原因について、次の3つが考えられる。

1つは、熱電対モジュール内部の基準温度のずれである。通常、熱電対モジュール内部のICで環境温度(気温)を測定し、その温度を基準に熱電対の温度を測定している。したがって、基準温度がずれると測定温度にも誤差が出る。

2つ目は、ノイズによる影響である。熱電対で発生する電圧は μV オーダーの小さい電圧であるため、ノイズによる影響を受けた可能性がある。

3つ目は、ノイズ対策のためSPI通信の熱電対モジュールの入力部に使用したコンデンサによる影響である。ノイズの影響で測定温度が低くでてしまう現象が起きたため、コンデンサを熱電対入力部のプラスとマイナスの間に入れた。しかし、熱電対モジュールの電源及びグランド間にコンデンサを入れることでノイズの影響を軽減できるため、今後は入力部のコンデンサを取り除いても問題ないと考えられる。

また、I2C通信の熱電対モジュールの測定値にばらつきが見られた要因には、前述の1、2が考えられる。

3. 結言

鋳型内部の温度測定を無線化・自動化するIoTモジュールの試作機を開発し、実際に使用する温度でデータロガーとの温度差の程度を確認した。その結果、I2C通信の熱電対モジュールを使用することで、データロガーとの温度差 $\pm 2[^\circ\text{C}]$ 以下で温度を測定することができた。SPI通信の熱電対モジュールは、データロガーとの温度差 $\pm 2[^\circ\text{C}]$ 以下には収まらなかったものの、データロガーとの温度差は最大で6.4 $[^\circ\text{C}]$ であることが確認できた。

今回は温度測定IoTモジュール製作の煩雑化を防ぐため、SPI及びI2Cの2種類の熱電対モジュールを2個ずつ使用した。しかし、温度差を $\pm 2[^\circ\text{C}]$ 以下にする場合、I2C通信の熱電対モジュールを4個使用することで4ch計測が可能である。その場合、I2C通信の熱電対モジュールのアドレスを変更する必要があり、VDD-ADDRピン間に一定の抵抗を付加することになる。

参考文献

- 1) "Node-RED". OpenJS Foundation. <https://nodered.org/>, (参照 2021-02-09)
- 2) "MAX31855 Data Sheet". 株式会社スイッチサイエンス. http://doc.switch-science.com/datasheets/MAX31855_jp.pdf, (参照 2021-12-20)
- 3) "MCP9600 Data Sheet". 株式会社ストロベリー・リ

- ナックス. <https://strawberry-linux.com/pub/20005426B.pdf>, (参照 2021-12-20)
- 4) "Raspberry Pi 4 Model B specifications - Raspberry Pi". Raspberry Pi Foundation. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>, (参照 2022-02-09)
- 5) "Arduino - Home". Arduino Foundation. <https://www.arduino.cc/>, (参照 2022-02-09)
- 6) "IBM Docs ". IBM Corporation. <https://www.ibm.com/docs/ja/ibm-mq/8.0?topic=ssfksj-8-0-0-com-ibm-mq-pro-doc-q002870--htm>, (参照 2022-02-09)
- 7) "データロガー midi LOGGER GL240 仕様 | GRAPH TEC ", グラフテック株式会社. https://www.graphtec.co.jp/site_instrument/instrument/gl240/spec.html, (参照 2021-12-20)