

プログラミング教材を利用したSTEM教育への活用の研究

自然科学系サブプログラム

小川 梨穂

【指導教員】 近藤 一史 大向 隆三 松岡 圭介

【キーワード】 STEM 教育 デジタル プログラミング センサー

1. はじめに

高等学校学習指導要領(平成30年告示)解説理科・理数編⁽¹⁾に記載の平成28年12月の中央教育審議会答申では、STEM教育において問題解決型の学習が重視されていること、ならびに、理科が中核となって探究的な学習な学習の充実を図っていきことが重要であることが書かれている。したがって、STEM教育の要素を取り入れた探求的な学習の時間を理科の授業の中で提案することが、現在の教員に求められていると考える。現在、STEM教育が推奨されている中で学校現場でのICT環境の整備が進んでいる。ICT機器の充実に合わせて今後の学校現場での測定機器はデジタル表示に替わっていくと考えられる。このような背景の中で、プログラミング教材を理科授業の実験の中で導入する取り組みが進められている。

2. 研究の概要

本研究では、高等学校で扱う「熱・エネルギー」の分野である熱量計を用いた混合法による水当量の実験⁽²⁾をプログラミング教材を取り入れながら行う。研究結果や過程から、どのような形でプログラミング教材を理科教育に生かすことが出来るか、プログラミング教材と合わせて使用するのに最適な実験器具はどれかを検討する。

3. 先行研究

卒業研究で混合法によるデジタル測定機器を用いた研究を行った。デジタル表示の温度計と一般の棒状温度計(アナログ表示)を用いて実験を行い比較した結果、デジタル表示の温度計の方が、測定結果が出るまでの時間(表示される温度が一定になるまでの時間)が速かった。これは、測定する際の水中の測定機器の体積と熱伝導率の違いによるものが大きいと考えられる。デジタル表示の温度計に使用されているステンレス鋼の熱伝導率は棒状温度計で使用されている水銀のおよそ2倍程度あり、測定部分の体積はデジタル表示の温度計が 0.1cm^2 、棒状温度計が 2.0cm^2 であり10倍差がある。よって、測定部分の金属部分が同じ温度になる時間が、デジタル表示の温度計の方が棒状温度計と比べて速いため、測定結果が出るまでの時間が速かったと考えられる。このことから、デジタル表示の温度計は測定結果が出るのが速い傾向にあると考えられる。

また、温度センサーについて、「Arduinoを用いたコンピ

ューター計測」⁽³⁾において、Arduino UNO R3(以後、Arduinoと表記されている場合にはこのモデルのことを指す)と温度センサー(LM61CIZ)を使用して、液体の比熱の実験が可能であることが報告されている。また、この研究により、30秒おきに $20^{\circ}\text{C}\sim 60^{\circ}\text{C}$ の範囲の水の温度測定した場合、棒状温度計と比較して誤差の小さい測定を行うことが可能であることが報告されている。

4. 研究内容

埼玉大学教育学部の物理学実験における熱の実験で、比熱、質量を用いて計算から求めた水当量と混合法から求めた水当量が一致しないことが多く見られる。このことについて、プログラミング教材を取り入れて実験を行い、混合法で求めた水当量の値が一致しない原因について探究する。プログラミング教材を利用する利点として、液体の温度変化を見る本実験において、どのように温度変化したか視覚的に分かりやすいという点が挙げられる。

プログラミング教材を使用する上で、従来の実験器具と比較されるデジタル表示の測定機器にて一度実験を行った。

混合法による水当量を求める手順は以下の通りである。

- 1) 銅容器と攪拌器の質量を電子天秤を用いて求める。
- 2) 1に水を加えた後の質量を測定し、加えた水の質量を求める。
- 3) 2の操作の後、水熱量計に容器を入れ、水の温度と加えるお湯の温度を測定する。
- 4) 容器にお湯を加え、即座に蓋をして、熱が逃げない状態で攪拌しながら混合後温度が安定した時の温度を記録する。
- 5) お湯を加えた後の質量を測定し、加えたお湯の量を求める。



図 1：水熱量計



図 2：デジタル温度計

測定した値を次の計算式に当てはめて水当量を求める。

M ：加えた水の量 (g)

m ：加えたお湯の量 (g)

t_1 ：水の温度

t_2 ：お湯の温度

t_3 ：混合後の温度

$$\text{水当量 (g)} = \frac{m(t_2 - t_3)}{t_3 - t_1} - M$$

比熱から求めた値と混合法から求めた水当量の値が一致しない点について、以下の条件設定をして実験を行い、考察をした。

- ・使用する水とお湯の温度
- ・水とお湯の温度と気温との差
- ・使用する水とお湯の量

比熱から求めた水当量は $12.842 \pm 0.019\text{g}$ であり、デジタル温度計を利用した際の混合法から求めた値に近いデータが得られる条件を探すことが原因を見つけるきっかけになると考えられる。

水（お湯）の量と温度の条件を設定するための基準となるデータの一つとして、お湯の温度を 50°C 前後 ($\pm 1^\circ\text{C}$)、使用する水とお湯の量をそれぞれ 100mL 分ビーカーで計測し、水当量を求めた。温度以外の計測はそれぞれ 5 回ずつ行った。

実験の際は水温の変化を少なくするため、使用器具と水温の温度に差がなくなるよう注意して行った。（お湯の量を測定するビーカーにビーカー全体が温まるまでお湯を入れておく、混合後、次の実験用の水を加える前に銅容器を水道水の温度にするなど。）

求めた水当量と測定した温度は以下のようになった。

表 1：水温と水当量

回	水の温度 ($^\circ\text{C}$)	お湯の温 度 ($^\circ\text{C}$)	混合後の 温度 ($^\circ\text{C}$)	水当量 (g)
1	13.0	50.2	30.1	18.55
2	12.9	50.2	30.2	20.51
3	13.0	50.2	30.1	12.82
4	13.4	49.6	29.8	13.42
5	12.9	49.8	30.3	13.98

また、各測定時に加えた水とお湯の質量は次の表 2 のようになった。

表 2：加えた水とお湯の質量

回	水の質量 (g)	お湯の質量 (g)	水当量 (g)
1	99.68	100.58	18.55
2	100.32	104.52	20.51
3	103.53	98.98	12.52
4	106.38	99.23	13.42
5	102.10	103.58	13.98

測定開始時の気温は 14.2°C 、測定終了後の気温は 14.5°C であった。

混合法の実験において、加えたお湯の量を測る際に混合後の液体が蒸発し、測定中に質量が減少傾向にあることが原因の一つかと思われていた。しかし、今回の測定では 2～4 回目にその傾向があったにもかかわらず、測定結果には影響がなかった。加えたお湯の質量が $98.98 \sim 104.52\text{g}$ の範囲であり、ビーカーで測った 100mL と差がないことからそのように考えられる。混合後の水温は 5 回の測定で 0.5°C 以上の差がなかった。また、加えた水とお湯の量も 100mL からおよそ 5mL の範囲に収まっていた。しかし、測定温度であった 1 回目と 3 回目で求めた水当量の差は 6.03g あり、比熱から求めた水当量が 12g 程度であることを考えると大きな差だと考えられる。表 1 と表 2 の結果から、水やお湯の質量について厳密に設定した方が、求める水当量に大きな差が生まれにくいと考える。水については最小目盛が小さい電子天秤を利用することが出来るが、お湯の質量については気温との差が大きい時に揮発する水分量が多いことや温度が実験に影響のある速さで下がることと合わせて測定方法を考える必要があると思われる。また、水の温度を測定する際に、室温から水の温度を測定した場合と、お湯の温度を測定してから水の温度を測定した場合とで、測定結果に違いがあるように感じられることがあった。今回はお湯の温度を測定した後に水の温度を測定するように心がけた。デジタル温度計にそのような傾向があるのか、何秒間表示された温度が安定すれば結果と呼べるのかについても議論が必要であると考えられる。

次に温度測定にデジタル温度計や棒状温度計を使用した場合とプログラミング教材を使用した場合と比較していくため、温度センサーを利用して混合法の実験を行った。温度測定専用 IC (LM35) を使用し、Arduino に接続して実験を行った。温度センサーを使用することで、図 2 の温度計と比べて、時間経過における温度変化を読み取りやすいと考えられる。したがって、水熱量計の中で温度がどのように変化しているか、意図していない熱エネルギーの放出はあるか、という観点から研究を進めていく。

先行研究の温度センサーを利用した測定では 30 秒おきに測定しており、温度が瞬時に変化する混合法の実験におい

て応用できるかが実証されていない。したがって、混合法の実験に使用する前に LM35 を使用した際にどのような挙動をするのか、何秒の間隔であれば測定可能なのかを調べた。

次の図 3, 図 4 のような形で Arduino と LM35 を接続し、測定を行った。0.1 μ F のコンデンサーはノイズ除去の為に組み込んである。

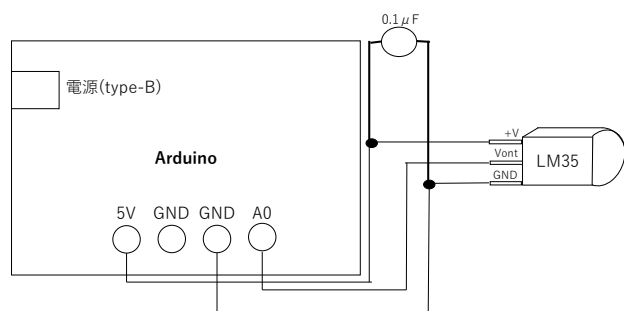


図 3 : Arduino と LM35 を繋いだ模式図

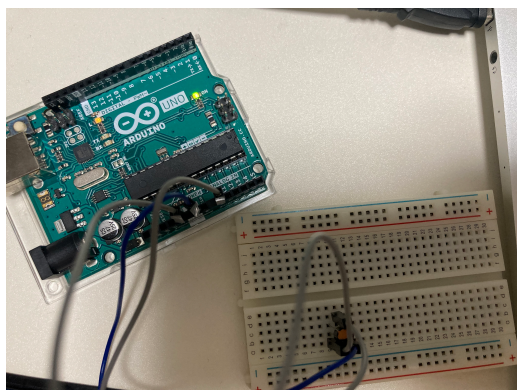


図 4 : 図 3 を繋いだ様子

温度センサーをドライヤーで 30 秒間温めた時の温度変化を 1 秒おきに測定した結果、次のようなグラフが得られた。

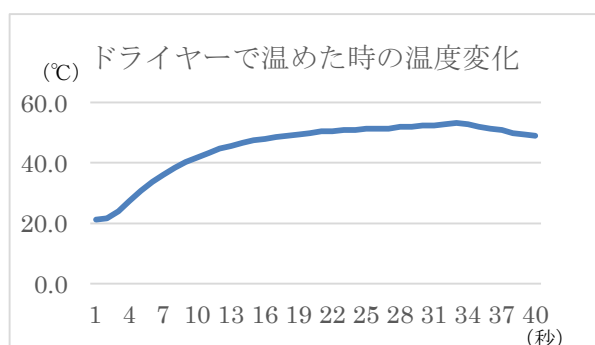


図 5 : 1 秒おきに測定した時のグラフ

測定時に出力される電圧は、実際の温度の約 2 倍の状態であり、かつ少数点以下までは出力されない。そのため、少数点第一位まで読み取り可能なデジタル温度計や棒状温度計ほど精度が高い測定をすることは不可能である。しかし、図 5 のグラフを見ると、比較的滑らかに温度変化しており、

急激な温度変化には対応しやすいと考えられる。50°C 前後で温度が停滞したのは、ドライヤーの温度の高さによるものだと考えられるので、温度の範囲による違いは観察できなかったと分かった。

次に、何秒の間隔であれば測定可能なのかを Arduino のタイムスタンプ表示機能 (タイムスタンプは測定時の時刻が 0.001 秒まで打刻される機能) を利用して調べた。何秒おきの条件を変え、0.1 秒遅れるまでに何回測定したかを調べた。

結果は以下の表 3 のようになった。

表 3 : 0.1 秒遅れるまでの測定回数

0.25 秒おき	84 回
0.5 秒おき	96 回
1 秒おき	96 回
2 秒おき	101 回

表 3 の結果から、約 100 回測定するごとに 0.1 秒遅れるが、測定間隔を短くしても影響がないことがわかった。また、0.1 秒おきに測定した場合、測定回数が 99~100 回で 10 秒経過したことから、0.1 秒間隔の測定までは可能だと言える。

以上の課題研究 I で行った結果から、Arduino の温度測定の処理速度であれば、温度変化の激しい混合法の実験に用いて問題ないのではないかと考えられる。

マイコンボードを活用して、より STEM 教育に関連づけることができるように、プログラミング教材に焦点を当て、実験を行った。

課題研究 I で行った実験では、データが数字で出力されるため、グラフとして表示するには、シリアルモニタで表示された時間とデータを Excel に入力してからグラフ化する必要があった。この作業はパソコンの使い方を学ばせるような授業であれば問題ないが、理科の授業という観点や、STEM 教育の一環として考えた場合、測定したその場でグラフ化の方が温度変化の様子が伝わりやすいのではないかと考えた。

簡単に温度測定の結果をグラフ化するという観点で調べた結果、プログラミング教材によく用いられている microbit の使用を検討した。

microbit ではプログラムが正常に動くかを確認できるシミュレーターという機能と、受信したデータをグラフ化する機能 (データを表示 デバイス) がある。microbit にプログラムを読み込ませた後、Microsoft MakeCode 上で数値の変化のグラフを見ることができる。また、受信したデータを

計測した時間と一緒に CSV ファイルで保存することができる。



図 6：MakeCode 上で表示されるグラフとデータ

	A	B
1	time (経過時間)x	
2	0	24
3	1.752	24
4	3.504	24
5	5.257	24
6	7.008	24
7	8.761	24
8	10.512	25
9	12.265	26

図 7：保存した CSV ファイルを開いた時の様子

また、課題研究 I で課題となっていた防水の面で、温度センサーに防水加工するのが最もコストの安い方法である。しかし、防水加工を行うとセンサー周りが大きくなり使用しづらくなる。

そこで、既に防水加工された温度センサーを用いることにした。TFabWorks から販売されている、microbit 用温度センサー TPW-TP2 を使用した。TFabWorks の公式サイト上の商品ページに、MakeCode のリンクが張られており、そのリンクから MakeCode を開くと、TFabworks から販売されている商品に対応した microbit の拡張機能が設定された状態の画面になる。通常の MakeCode を開いた場合には、拡張機能 (Bluetooth などの設定が可能になる) は必要に合わせて検索する必要があるため、実験器具に合わせて検索する必要がない分、操作が慣れていない人でも使用しやすいと考える。

Microsoft MakeCode でのプログラムは次の図 8 である。

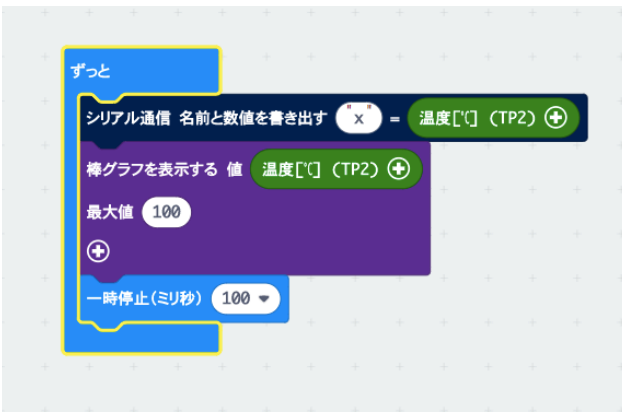


図 8：MakeCode で書いたプログラム

接続方法は次の図 9 であり、パソコン→microbit→温度センサーの順で接続するのみである。

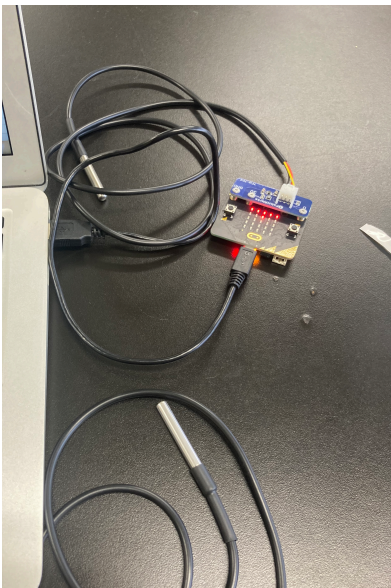


図 9：パソコンに microbit と TPW-TP2 を繋いだ様子

microbit での温度測定は、接続やグラフ作成が楽である点が利点だと感じた。問題点としては、図 8 のプログラムでは 100 ミリ秒=0.1 秒単位で計測するプログラムになっているが、実際に動かすと約 1 秒ごとにしか計測されなかった点である。さらに 1 秒単位でプログラムした場合にも、1 秒ごとにはならず、2 秒前後ごとになり、プログラムと実際の測定間隔に大きなズレがあった。具体的な秒数のズレは以下の表 4 の通りである。

表 4：microbit の測定間隔と設定秒数との差

設定した秒数	実際の測定間隔	設定秒数との差
0.100 秒	1.75 秒	1.65 秒
0.200 秒	1.85 秒	1.65 秒
0.500 秒	2.15 秒	1.65 秒
1.000 秒	2.74 秒	1.74 秒

表 4 から microbit での測定では 1.7 秒前後必ず差がある

ことが分かる。また、CSV ファイルを確認すると、表 4 の法則よりもさらに測定間隔が大きい時が図 10 のマーカーを引いた部分のように数回確認出来た。これは、0.1 秒間隔で測定可能な Arduino と大きな差であり、急激な温度変化を見る必要がある本研究でも課題となる。

1	time (経過時間)	温度 (℃)	測定間隔 (秒)
2	0	27	
3	1.723	27	2.65
4	4.373	28	3.477
5	7.85	29	2.639
6	10.489	31	2.651
7	13.14	32	2.667
8	15.807	32	2.637
9	18.444	33	3.477
10	21.921	33	2.639
11	24.56	33	2.652

図 10：時間→温度→測定間隔をまとめたファイル

そこで、測定間隔が短い Arduino で使用可能な防水センサーを探したところ、DS18B20 が安価で、Arduino で使用可能であったため、DS18B20 でも microbit と同様に水温測定を試すことにした。

DS18B20 を使用する場合、Arduino のツール機能→ライブラリを管理→ライブラリマネージャにて

”DallasTemperature”、”OneWire” をインストールしてからコードを書いた方が簡単にプログラムすることが出来る。Arduino のライブラリ機能は、他の人が書いた Arduino 用の言語でのプログラムを保存し使用できる機能で、include でファイル内のプログラムを書き込んだことと同義になる。”DallasTemperature” と ”OneWire” は温度測定のプログラムにてよく使用されているものである。

実際に使用したプログラムは次の通りである。

```
#include <DallasTemperature.h>
#include <OneWire.h>

#define ONE_WIRE_BUS 10
#define SENSER_BIT 9

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

void setup(void) {
    Serial.begin(9600);
    sensors.setResolution(SENSER_BIT);
}

void loop(void) {
    sensors.requestTemperatures();
    Serial.println(sensors.getTempCByIndex(0));
}
```

図 11：Arduino での測定で使用したプログラム

ONE_WIRE_BUS はコードを差す場所を指定しているのみであるため、必要に応じて変更して問題ない。また、SENSER_BIT は温度センサーの分解能を指定するものであり、0.5℃、0.25℃、0.125℃、0.0625℃に対応して、9、10、11、12 ビットを選択することが出来る。精度を高くすると値のふらつきが大きくなりやすい点と、この段階では、1℃単位でしか表記されない microbit と比較するため、9 ビットに設定した。

DS18B20 と Arduino を次の図 11 のように繋いで温度を測定した。抵抗は 10kΩ 2 個を並列に繋いで 5kΩ の抵抗として使用した。

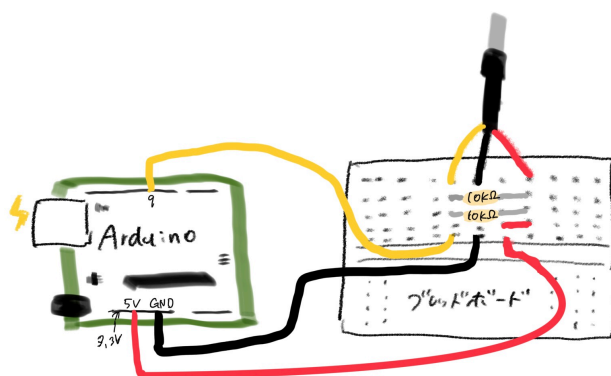


図 12：Arduino と DS18B20 を繋いだ図

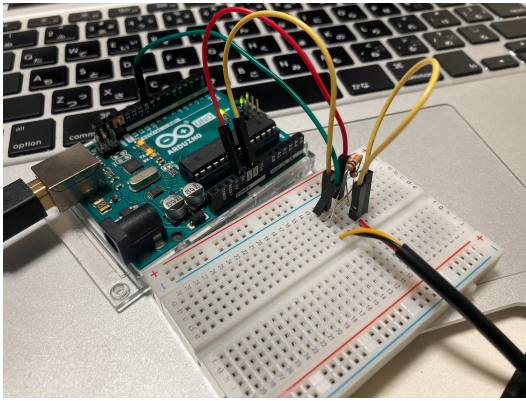


図 13 : DS18B20 と Arduino を繋いだ様子

Arduino での温度測定では、LM35 を用いた時よりは測定間隔が大きいものの、0.1~0.2 秒間隔で測定することが出来る点が魅力に感じた。Arduino ではシリアルモニタにてタイムスタンプと受信データを見ることが出来る。また、2022 年に追加されたシリアルプロッタにて、microbit と同様に受信したデータのグラフを見ることが出来る。microbit と比較した Arduino の欠点としては、シリアルモニタとシリアルプロッタを同時に表示することが出来ず、シリアルモニタとシリアルプロッタを切り替える際に、その時表示されていたデータが消えてしまう点である。microbit ではグラフ表示と CSV ファイルでのデータ保存が可能のため大きな違いと言える。また、シリアルプロッタで表示される温度のメモリが自身で設定出来ず、表示中は具体的な測定データの数字が分からない点も実験で活用する上で問題点であると感じた。

```
14:22:27.547 -> 178000011444
14:22:27.547 -> 16.3199996948
14:22:27.547 -> 19.2000007629
14:22:27.547 -> 3
14:22:27.547 -> 15.3600006103
14:22:27.547 -> 14.8800001144
14:22:27.547 -> 15.8400011062
14:22:27.547 -> 15.8400011062
14:22:27.547 -> 14.4000005722
14:22:27.547 -> 17.7600002288
14:22:27.547 -> 16.8000011444
14:22:27.547 -> 16.3199996948
14:22:27.547 -> 19.2000007629
14:22:29.131 -> 18.2399997711
```

図 14 : シリアルモニタ上のデータ

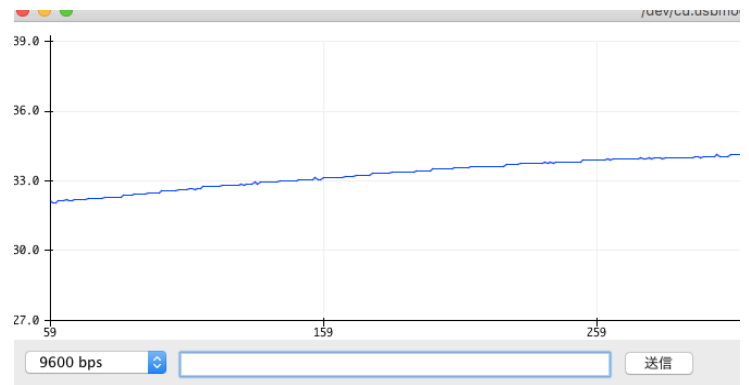


図 15 : シリアルプロッタで描いたグラフ

DS18B20 での測定を行っている中で、microbit 用の温度センサー TPW-TP2 と基本的な接続方法が変わらないののではないかと感じ、TPW-TP2 を Arduino に接続して測定を試みた。プログラムは図 11 と同じものを使用し、電圧のみ TPW-TP2 に表記されている 3.3V と近い 3V に接続して測定を行った。

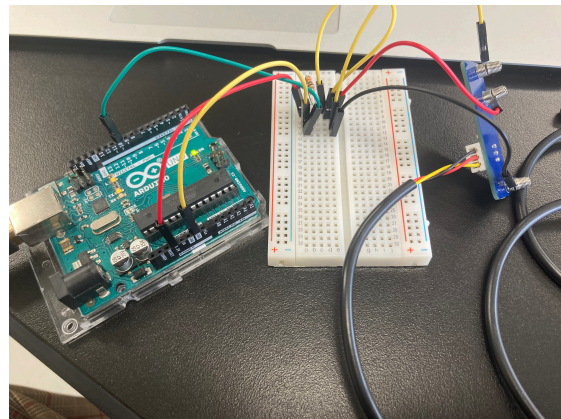


図 15 : Arduino に microbit を繋いだ様子

その結果、DS18B20 と同様の測定結果を得ることが出来た。センサー精度がどちらも $\pm 0.5^{\circ}\text{C}$ であることから、TPW-TP2 を Arduino での温度測定で使用しても問題ないと考えられる。今回の実験では、コードの接続をバネ部分に挟み込んだだけであり、この接続方法は生徒に器具の使い方を指導する上で不適切であると考えるので、実際に授業で使う際には適切な教員のアプローチが必要である。

LM35 での測定、microbit での TPW-TP2 を使った測定、Arduino での測定から、防水でない温度センサーが 10 ピース 1758 円 (Amazon 価格, LM35 の場合) = 1 個 176 円であるのに対し、防水温度センサー DS18B20 は 3 本 798 円 = 1 本 266 円 (Amazon 価格) であり、加工する手間や使用感を考えると、温度センサーは最初から防水温度センサーでも問題ないのではないかと考えられる。特に学校の理科の実験でプログラミング教材を取り入れる場合は、地学の気象観測のようにゆっくり長期間かけて測定するようなものよりも、急激な温度変化を観察する液体の温度測定の場合が多いと考えられるためである。精度を求める場合には、別の温度センサーの採用が検討されるが、それ以外の場合は

DS18B20 で問題ないと思われる。また、TPW-TP2 は 1 本 2310 円 (TFabWorks サイト価格) であり、他に使用した 2 個と比較すると高値であるが、microbit で使用する場合、回路を組む作業を必要としないのは利点である。本研究では対象としていないが、小学生など小さな金属類を扱わせると怪我が心配があるような場面では有用だと思われる。

次に、水当量を求める混合法の実験では、「水」と「お湯」と「混合後」の温度の 3 つの温度を測定する必要があり、特に「お湯」の温度と「混合後 (混合中)」の温度が測定結果に強く影響すると考えられるため、比熱から求めた水当量との混合法で求めた水当量の値の差が大きい問題の原因を調べるには、最低でも 2 つ同時に温度を測定した方が良いのではないかと考えた。そこで、これまでに使用してきた温度センサーを使用して 2 点同時に測定データを取る方法を考えた。

まず、パソコンに USB Type-A の差込口が 2 個あることを利用して、Arduino をそれぞれに接続し、同時に測定できないか試みた。

その結果、パソコン自体は 2 つの Arduino を検知はしているものの、表示上 1 つのシリアルモニタ (またはシリアルプロッタ) しか表示することが出来ず、同時に表示することが出来なかった。したがって、この方法では 2 個同時に温度の測定データを取る事が不可能であった。

この解決策として、Arduino 1 つに温度センサーを 2 つ接続し、センサーごとに番号を振り分けて測定することが考えられる。Arduino のシリアルプロッタの機能では、数式などで複数の線を描くことが可能であり、理論上可能であると考えられるが、本研究では適切なプログラムと回廊図に辿り着けなかったため、別の方法で 2 点で同時に測定出来ないか考えた。

そこで、Arduino と microbit を同時に使用すれば良いのではないかと考えた。Arduino はアプリのような形でパソコン上は処理されており、microbit は Google Chrome のタブ上で測定データを見ることが出来るからである。

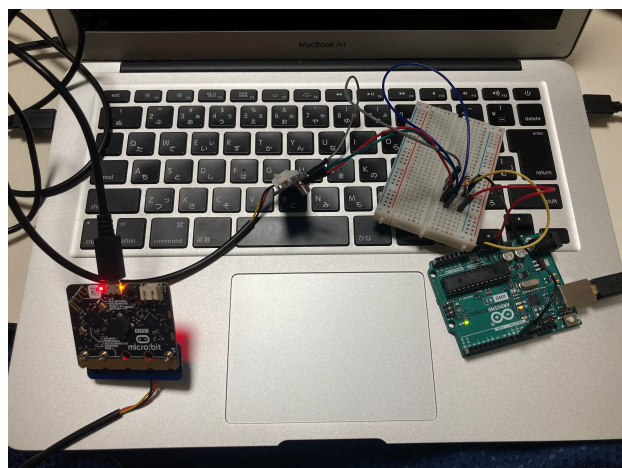


図 16: Arduino と microbit を同時に繋いだ様子

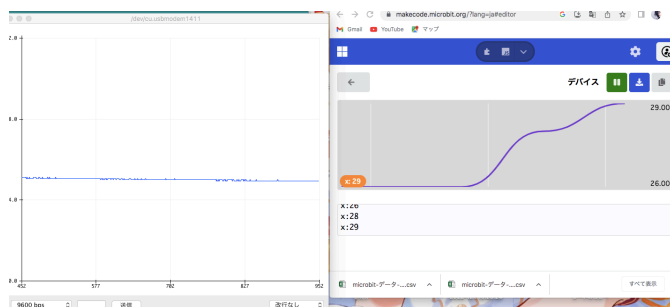


図 17: シリアルプロッタと microbit のグラフを表示したパソコン画面

この結果、図 17 のように 2 つ同時に測定することが可能であった。1 つのパソコン画面に 2 つのグラフを同時に表示することが出来、授業等で使用しても問題ないと考えられる。また、同じ温度の液体を測定した際の温度差が microbit と Arduino 間で 1°C 程度あり、これは誤差の範囲内と考えられるため、同時に測定した場合も測定データには問題ないと考えられる。

この結果から、温度センサー TPW-TP2 と Arduino、microbit を使用して、混合法による水当量の実験を行った。参考用のデータとして水銀の棒状温度計 (読み取り誤差 0.1°C) も用意して実験を行った。混合時の温度は測定間隔の短い Arduino で行い、お湯の温度と水の温度は microbit で行った。

センサーとしての性能を調べている時は問題なかったのだが、実際に実験に使用しようとした時に、水銀温度計と温度センサーで $1\sim 2^{\circ}\text{C}$ 温度が違うというトラブルがあった。具体的には以下の表 5 の通りである。

表 5: 温度センサーと棒状温度計の測定結果

	温度センサー	棒状温度計
水の温度 ($^{\circ}\text{C}$)	17	15.3
お湯の温度 ($^{\circ}\text{C}$)	27	25.7
混合後の温度 ($^{\circ}\text{C}$)	22	20.2

水当量は本来正の数になるものであるが、温度センサーの測定値を使って水当量を求めると負の数になってしまい、実験で求めることが出来なかった。この主な原因として、混合法による水当量の実験は温度が 1°C 違うと測定結果で求められる水当量の差がかなり大きくなるものであることが挙げられる。

したがって、内部の温度変化を求めるのに温度センサーであつたりプログラミング教材を使うことは問題ないが、混合法のような精度が求められる実験では棒状温度計のように読み取り時の誤差が小さいものを使用した方が実験の効率が良いと思われる。

5. まとめ

使用したプログラミング教材（microbit と Arduino）と測定機器自体に問題はないが、今回の混合法の実験であると、マイコンボードによる測定間隔の差や機械的誤差の大きさによる影響が大きく、物理学実験における課題を解決する糸口を見つける方法を見つけることが出来なかった。しかし、液体を 2 点同時に測定する手段であったり、Arduino、microbit それぞれのプログラミング教材としての価値を知ることが出来た。特に microbit は専用の商品を使用すると金額が嵩むものの、初心者でも使いやすく、高校生が利用するだけでなく、高校生が中学生や小学生に教えるような授業形式でも使用可能なのではないかと考えられる。さらに microbit は JavaScript 表示にすることができ、プログラミングに深く興味がある生徒にも対応することが出来る。Arduino は独自の言語を使っている点が難しいが、microbit よりもプログラミング言語を扱っている感覚が強く、ブレッドボードなどを物理の内容と絡めることが出来れば、生徒が思考する内容が多く、深く多面的な学びに生かすことが出来るのではないかと考える。また、2 点同時に液体を測定できることによって、蒸留の実験での液体と水蒸気の温度などをグラフとして表示するなどの手段で高等学校の授業でも活用できるのではないかと思う。

本研究では 1 人で全ての実験を行ったが、実際の学校現場で実験を生徒が行う時は、複数人で行うことが多いと思われる。実験操作中に 2 つ同時にグラフが表示されても処理できる情報には限度があるが、複数人のグループで、他の人が実験操作をしている時に複数のグラフを同時に見ることが出来るのは、物事を平行して考えるのが苦手な生徒が実験内容を比較しやすくなるなどの効果があるのではないかと考える。

6. 今後の展望

本研究ではパソコン上の処理だけであったが、最近の学校現場では、教室等場所を選ばずに使用できるスマートフォンやタブレット端末を活用した授業も注目されている。

iPhone の App Store に存在する“BLE Scanner”というアプリを使用し、Bluetooth にて図 4 と同じ温度センサーを使用した状態でパソコンとアプリを接続してみたところ、パソコンからのデータを受信することは出来た。しかし、出力された数字を温度に変換する必要がある、更にグラフ化するには、別の Web サイト等でデータを変換する必要があった。Web サイトでは会員登録や、データ送信用のマイコンボードと受信用のマイコンボードを分けて使用する必要があったため、本研究で用意した microbit と Arduino ではグラフとして表示することは難しいと判断した。しかし、今後さらに STEM 教育が研究されていく中で、BLE Scanner や似た機能を持つ“Cisco Secure Client”などといった Bluetooth を使用してスマートフォンなどでマイコンボードからデータを受信することが出来るものが、プログラミングを含んだ授業に取り入れられていく可能性は高い。

本研究では深く追求することが出来ず、分かりやすい使

い方等にたどり着くことが出来なかったが、今後スマートフォンなどの端末を活用する方法を広めることに繋げていきたいと考える。

7. 参考文献

- (1)「【理科編 理数編】高等学校学習指導要領」理科改訂の趣旨(2)理科の具体的な改善事項 文部科学省 (2018) P. 8
- (2)「物理学実験」吉田卯三郎 三省堂 (1940)
- (3)「Arduino を用いたコンピューター計測」第 2 節 温度計測システムを用いた実験の結果と性能評価 石田卓矢 (2015) P. 7~8, P. 43~51
- (4)TFabWorks>商品紹介>micro:bit 用防水温度センサー【TPW-TP2】
<https://tfabworks.com/product/tfw-tp2/>