

培地残量の把握を目的とした アラートデバイスの作製

田中 孝国^{*1}, 出川 強志^{*2}, 小林 康浩^{*3}, 田中 昭雄^{*3}

Making of an alert device to determine the amount of culture media

Takakuni TANAKA, Tsuyoshi DEGAWA, Yasuhiro KOBAYASHI and Akio TANAKA

During continuous culture, bacterial death and reduction of growth potential due to media depletion are major issues. This problem is particularly acute when multiple reactors are operated. In this experiment, a weight sensing and LINE notification system was developed to prevent depletion of culture medium. As a result, it was found that the LINE notification system can avoid medium depletion. This technology can facilitate inexpensive management of medium volume during long-term culture.

KEYWORDS : Alert device, Continuous culture, LINE notification

1. まえがき

我々は、複数の自動微生物培養装置(バイオリアクター)を稼動する際に、実験者の負担を軽減させる自動培養装置の作製を実施してきた。昨年度は、安価なポンプの作製法およびロードセルによる培地残量の把握する装置(ブザー音とディスプレイモジュールで培地残量を通知する仕組み)を作製し運用した。その一方で、実験室内のブザー音の不快であること、および実験室外ではブザー音が聞こえないなどの課題があったり。今回は培地残量が少なくなった際に、実験者のスマートフォンに LINE 通知する装置(アラートデバイス)を開発、作製したので報告する。

2. アラートデバイスの作製

本研究で作製したアラートデバイスは、培地残量を把握し、一定量を下回ると実験者に通知する装置である。培地残量を計測する装置として、フローセンサータイプが考えられるが、長期培養時のコンタミネーションを避けるため、フロート型ではなく重量感知型を採用した。市販品として、冷蔵庫内の卵の重量を通知する装置(Panasonic, 重量検知プレート(卵ケース付き), 約 7,000 円)が販売されている²⁾。この装置は、重量 0.04 [kg]~5 [kg] に対応しており、専用アプリを用いてメッセージを送ることができる。しかし、我々の培地は約 50 [kg]/台 の容量があるため、この装置は使用出来ない。そのため、新たなアラートデバイスの作製を実施した。

*1 物質工学科(Dept. of Materials Chemistry and Bioengineering), E-mail: tanakatakakuni@oyama-ct.ac.jp

*2 技術室(Technical Office)

*3 電気電子創造工学科(Department of Innovative Electrical & Electronic Engineering)

培地の重量を感知させる主要な部品として、昨年度に引き続きロードセルを採用した。具体的にはロードセルによって電気信号を発信し、ロードセル用ADコンバータを用いて数値化し、その後 Raspberry Pi によって重量に変換および LINE 通知させることを考えた。昨年度と異なる点は Arduino ではなく Raspberry Pi を用いたこと、ブザー音ではなく LINE で通知することである。

Raspberry Pi は Linux ベースの OS が利用でき、Windows や macOS のようなグラフィカルな環境での操作が可能である。GPIO(General Purpose Input / Output)と呼ばれるインタフェースを搭載しているのが特徴である。GPIO は、デジタル入出力が行える端子で、ここに LED やモーターといった電子デバイスを接続することで、Raspberry Pi から制御を行える³⁾。ロードセル(シングルポイント(ビーム型)SC133 20kg、秋月電子通商 112034)とロードセル用 AD コンバータ(HX711 使用 ロードセル用 AD コンバータ モジュール基板、秋月電子通商 112370)、Raspberry Pi(Raspberry Pi 4 Model B 2GB、秋月電子通商 114839)を電線で接続し、はんだ付けた。その後、ロードセル部分を培地容器の下に設置し、重量から培地の残量確認を一目で分かりやすくした。アラートデバイスは1台約¥18,000 程度で作製することができた(図1)。

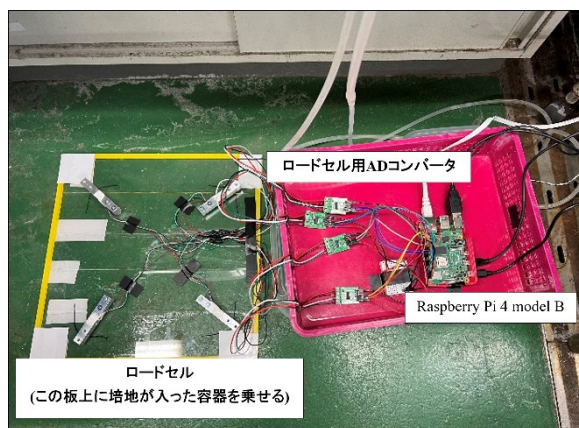


図1 アラートデバイス

本研究で用いたロードセルの測定可能範囲は0~20[kg]であり、このロードセルを4つ用いたため、0~80[kg]である。尚、ロードセル4つから異なる電気信号が送信されるため、ADコンバータは4つ使い、それぞれの電気信号を数値化した。続いて、Raspberry Pi のセットアップ方法について述べる。Raspberry Pi は、起動するために OS をまずインストールする必要がある。Raspberry Pi 専用の

OS「Raspberry Pi OS」を micro SD カードにダウンロードする。Raspberry Pi を操作するために、モニター、マウス、キーボードを接続する。モニターに表示される指示に従いながら、国名、言語、時刻、ユーザー名、パスワード、使用するネットワーク、デフォルトブラウザを選択する。OS を最新版にアップデートし、再起動する⁴⁾。次に、プログラムのセッティング方法を示す。Raspberry Pi 上に最初からインストールされている開発ソフト「Thonny Python IDE」を起動する。プログラムコードの部分に、目的のプログラミングを入力する。「SAVE」をクリックし、プログラミングコードを保存する。「RUN」をクリックし、プログラミングコードを実行する。この時、図2の実行結果にプログラミングの実行結果が表示される⁵⁾。プログラムは既存のサイトのものを参考にした。重量を感知するプログラム⁶⁾と、LINE を送信するプログラム⁷⁾を組み合わせ、アラートデバイス用に編集している。

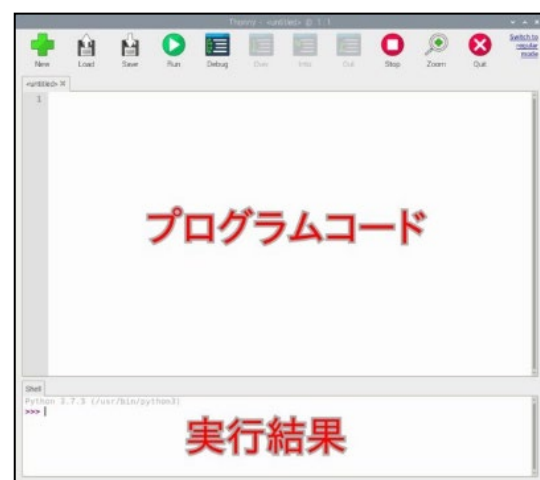


図2 Thonny Python IDE
統合開発環境 Integrated Development Environment

アラートデバイスの回路図を以下の図3に示した。プラスチック板の上にロードセルをセットし、その上に培地が入った容器を置き、感知した重量が電気信号として発信される。その電気信号をロードセル用ADコンバータが数値に変換し、得られた数値を Raspberry Pi 上で重量に変換している。一連の動作で感知した重量が1.5[kg]以下になったところで、LINE が送信される仕組みになっている。LINE の送信には、LINE Notify サービスを利用した。尚、試作後に本研究で用いたロードセルは、ひずみゲージを用いた重量検知をするため(ひずみが生じる測定対象物に電気絶縁物を介し

て接着しておき、測定物の伸縮に比例して金属が収縮することを利用して抵抗値を変化させ、電気信号を送信する原理⁸⁾、ロードセルが歪むための隙間を作る必要があることに気づいた。そのため、下に敷くプラスチック板とロードセルの間にナット(六角ナット M4 規格)を縦に2つ重ね(高さ 6.4 [mm])、2か所(図4中の矢印部分)に挟むことで、ロードセルが培地の重さにより歪むための空間を確保した(図4)。

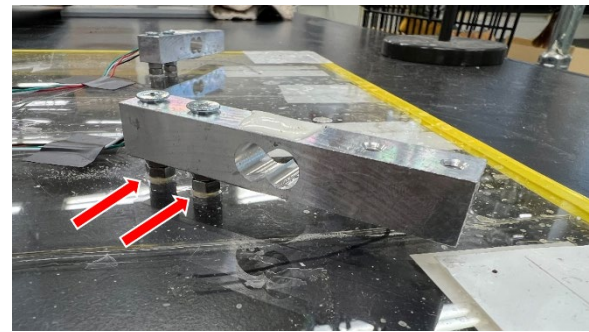


図4 ロードセル

図中の矢印部分にナットをそれぞれ入れ、高さを調整している。

3. 結果

今回作成したプログラムは付録に載せた。昨年度作製した装置と比較すると、アラートデバイスは、離れた場所であっても培地量の把握を実験者のスマートフォンで把握できるようになった。培地残量が 17.50~16.50 [kg] になると LINE を送信することに成功したが、特に培地の追加を実施しなかったところ、3 日間で約 35,000 件の膨大な LINE が送信された(図5)。その結果を受けプログラムを改良したところ、LINE の受信件数を 20 件程度にするためには、培地残量の範囲を 17.50~17.49 [kg] 程度など 1.0 [g] の重量範囲が適していることが判明した。本装置はこれらの設定を完了した後、問題無く動作していることが確認できている。

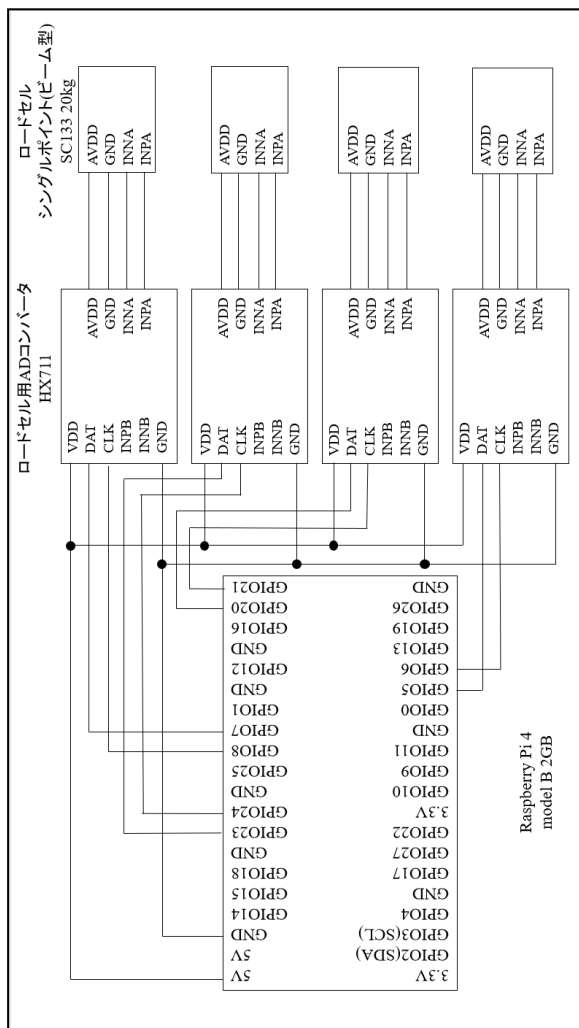


図3 アラートデバイスの回路図

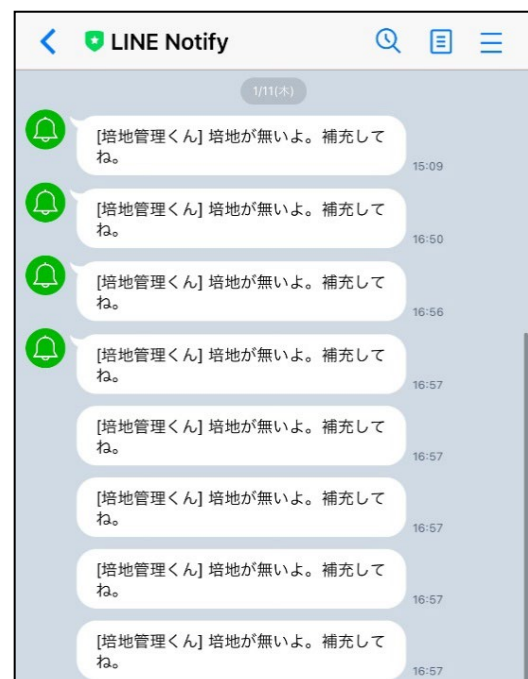


図5 受信した LINE メッセージ

4. あとがき

今回、連続培養時に培地枯渇を防ぐことを目的とした、重量感知および LINE 通知システムについて作製した。その結果、LINE 通知により培地枯渇が回避できることが判明した。本技術は、長期培養時の培地量管理を安価に容易にすることができる。

参考文献

- 1) 田中孝国 他：海洋性 Anammox 細菌の自動培養装置作製の試み (第一報), 工業用水, No.678, pp.74-78 (2023.05)
- 2) <https://ec-plus.panasonic.jp/store/ap/storeaez/a2A/ProductDetail?HB=NY-PZE1B1> (閲覧日 2024.03.22)
- 3) 福田和宏：Raspberry Pi 電子工作 実践講座、ソーテック社、p.10 (2019)
- 4) <https://www.indoorcorgielec.com/resources/raspberry-pi/raspberry-pi-setup/> (2024.03.22 閲覧)
- 5) <https://www.indoorcorgielec.com/resources/raspberry-pi/raspberry-pi-python/> (2024.03.22 閲覧)
- 6) <https://note.com/izawa/n/n0b4d4866470a> (2024.03.22 閲覧)
- 7) <https://sozorablog.com/pythonline/> (2024.03.22 閲覧)
- 8) https://tml.jp/knowledge/strain_gauge/about.html (2024.03.22 閲覧)

付録 作成したプログラムについて

LINE 送信をさせる培地残量の設定範囲を変える際は、if val1+val2+val3+val4...の行の数値を変化させればよい。ただしプログラム中の数値は [g] 単位である。プログラム中に出てくるトークンとは、アクセストークンのことで、Web サービスと連携せず通知を設定することができる LINE Notify の機能である。LINE でトークルーム作成後に LINE Notify ページにて無料で発行される。再発行は困難であるため、発行後は記録しておく必要がある。

```
#! /usr/bin/python2
```

```
import time
import sys
import requests
```

```
EMULATE_HX711=False
```

```
#キャリブレーション値
referenceUnit = 120
```

```
if not EMULATE_HX711:
    import RPi.GPIO as GP10
    from hx711 import HX711
    print('import module')
else:
    from emulated_hx711 import HX711
    print('import emulated')
```

```
def cleanAndExit():
    print('Cleaning...')
```

```
if not EMULATE_HX711:
    GP10.cleanup()
```

```
print('Bye!')
sys.exit()
```

```
hx1 = HX711(5,6)
hx2 = HX711(20,21)
hx3 = HX711(23,24)
hx4 = HX711(7,8)
```

```
print('program start')
```

```
hx1.set_reading_format("MSB", "MSB")
hx2.set_reading_format("MSB", "MSB")
hx3.set_reading_format("MSB", "MSB")
hx4.set_reading_format("MSB", "MSB")
```

```
hx1.set_reference_unit(referenceUnit)
hx2.set_reference_unit(referenceUnit)
hx3.set_reference_unit(referenceUnit)
hx4.set_reference_unit(referenceUnit)
```

```
hx1.reset()
hx2.reset()
hx3.reset()
hx4.reset()
```

```
hx1.tare()
hx2.tare()
hx3.tare()
hx4.tare()
```

```
#これが表示されたら載せてよい
print("Tare done! Add weight now...")
```

```
#1 秒間隔で重さを取得,表示
while True:
    try:
        val1=hx.get_weight(5)
        val2= hx.get_weight(5)
        val3= hx.get_weight(5)
        val4= hx.get_weight(5)
        #print("val1=",round(val1,2),
"val2=",round(val2,2),"val3=",round(val3,2),
"val4=",round(val4,2))
        value= val1+val2+val3+val4
        print("value:",round(value,2)," g")

#重量が 15kg 以下 14kg 以上なら line で通知(バケツ
の重さ 2.5 キロを考慮)

If val1+val2+val3+val4 < 17500 and val1+val2+val3+val4
> 16500:
    print("send line")
    url = #https://notify-api.line.me/api/notify"
    token="[ここに取得したトークンを入力する]"
    gY8i5cb"
    headers = {"Authorization" : "Bearer "+ token}
    message = "培地が無いよ。補充してね。 "
    payload = {"message" : message}
    r = requests.post(url, headers = headers,
params=payload)

hx1.power_down()
hx2. power_down()
hx3. power_down()
hx4. power_down()
hx1.power_up()
hx2. power_up()
hx3. power_up()
hx4. power_up()
times.sleep(1)

except (KeyboardInterrupt, SystemExit):
    cleanAndExit()
```

[受理年月日 2024 年 8 月 6 日]