Raspberry Piでキッチンのスマート化を図る！

福井県立武生高等学校　　鈴木真央　田中心夏　川﨑暖々　田中沙優

Abstract

Our goal is to reduce wasted time in daily life by utilizing IoT technology. In the future, we aim to specifically shorten the time lost in the kitchen and make more effective use of that time. This research focuses on developing a foundational system to support that vision. In this study, we focused on condiment containers in the kitchen. By connecting a load cell to a Raspberry Pi and controlling it through programming, we are developing a product that continuously displays the remaining quantity of seasonings in real time.

１　はじめに

1.1　ICT・IoTの普及

近年、世界では情報技術が発展しICT技術が様々な分野で活用されている。ICTとは「Information and Communication Technology」の略称で、ICT技術の活用により人々の生活をより暮らしやすいものにすることが可能である。さらにICTの中にIoTがある。IoTとは「Internet of things」の略称でモノのインターネットのことを指す。例えば私たちがスマートフォンでペットカメラを通じてペットの様子を見れるシステムもIoTである。

1.2　日常生活の不便

私たちはIoT技術に目をつけ、日常生活で不便なことを解消する仕組みを開発することにした。日常で「こんなものがあったら便利」「こういうときに不便」と感じることを挙げ、そこで挙がったのが冷蔵庫にある食材を把握する機能だ。買い物をするときに家に何があって何がないのかを把握でき、家にある食材の賞味期限がいつなのかをいつでもどこでも把握できる機能があれば家事の負担を軽減できると考えた。

1.3　私たちが目指すもの

私たちは冷蔵庫の中の食材をスマートフォンなどで常時確認できるようなシステム開発を目指している。そうすることで買い物における買い忘れや買いすぎを防ぐことができると考えている。そこで今回は調味料を常に管理できるシステムを開発する。調味料に対象を絞ったのは液体よりも固体のほうが量りやすく、野菜などの個数で表示すべきものよりも絶対的な重さで表示できる調味料の方が適していると考えたからだ。しかしスマートフォンで管理する段階に到達するまでにアプリ開発から残量を常に量るシステムまで考えなければならない。そのため今回は調味料の残量を把握するシステムの開発に取り組む。しかしここで注意するのは、今後の段階として残量をデータに変換してデバイス機器に送信しなければならないため、電子ばかりのような単純に重さを量るシステムにならないようにしなければならない。

1.4　参考製品

ここで私たちが目指している製品が商品化されていないかを調べた。「シャープ株式会社（Sharp Corporation）」から提供されているストックアシストという商品が私たちの理想に近いと感じたため、どのような製品なのか詳しく調べた。しかし、ストックアシストは15分に一度しか情報が更新されないのに加え、残量表示が5段階でしか表示されないため細かい残量まではわからないというデメリットがあった。そこで私たちはこの商品にないような即座に正確な残量を測ってくれるシステムを開発することにした。

２　制作方法

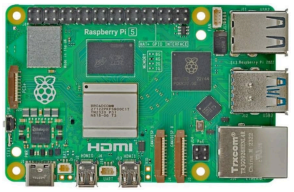
まず私たちは主にプログラミングを行う内部設計と、それを内蔵するための媒体を作成する外部設計に分かれて研究を行った。

2.1　内部設計

2.1.1　使用器具

Raspberry Pi（資料1）、ロードセル（資料2）、ディスプレイ、パソコン周辺機器、その他*

*：HX711などの小さな部品を含む。今回の研究内容を説明する過程に不要なため割愛する。



資料1　Raspberry Pi

資料2　ロードセル

Raspberry Piとは、基本的な機能を搭載したシングルボードコンピューターのことで、比較的安価で扱いやすいため今回の研究に適していると判断した。

ロードセルとは、どれだけ重いかを電気信号に変換するセンサーのことで、ロードセルはRaspberry Piと接続することで、測定結果を数値化し表示することができる。

2.1.1　設計方法

まずロードセルとRaspberry Piを接続する。Raspberry Piは、くら寿司の注文用タブレットなど、身近な場面でも使用されている安価で高機能なコンピューターである。接続後は、インターネット上で公開されているソースコード（図1）を参考にプログラミングを行った。初期状態では正確な重量が測定できない*1ため、重さがわかっている物体を用いてコード内の数値を調整*2し、安定して正しい値が表示されるようにする。その後、後述する調味料入れの重さを差し引く。またこのソースコードを改善し、小さいディスプレイに常に残量が表示されるようにプログラムを追加する。さらに、残量を小数点第1位まで表示するようにし、新しいデータが常に羅列される形で表示されるのではなく1箇所に表示され続けるようにした。

*1：初期状態では使用する環境の違いによって表示される値が異なるため。

*2：ソースコードの41行目に、表示された値/おもりの重量で出た値を入力する。

```python
import time
import sys
import RPi.GPIO as GPIO
from hx711 import HX711

def cleanAndExit():
    print("Cleaning...")

    print("Bye!")
    sys.exit()

hx = HX711(5, 6)

'''
I've found out that, for some reason, the order of the bytes is not always the same between versions
and the hx711 itself. I still need to figure out why.

If you're experiencing super random values, change these values to MSB or LSB until you get more st
There is some code below to debug and log the order of the bits and the bytes.

The first parameter is the order in which the bytes are used to build the "long" value. The second
the order of the bits inside each byte. According to the HX711 Datasheet, the second parameter is M
shouldn't need to modify it.
'''
hx.set_reading_format("MSB", "MSB")

'''
# HOW TO CALCULATE THE REFFERENCE UNIT
1. Set the reference unit to 1 and make sure the offset value is set.
2. Load you sensor with 1kg or with anything you know exactly how much it weights.
3. Write down the 'long' value you're getting. Make sure you're getting somewhat consistent values.
    - This values might be in the order of millions, varying by hundreds or thousands and it's ok.
4. To get the wright in grams, calculate the reference unit using the following formula:

    referenceUnit = longValueWithOffset / 1000

In my case, the longValueWithOffset was around 114000 so my reference unit is 114,
because if I used the 114000, I'd be getting milligrams instead of grams.
'''

referenceUnit = 114
hx.set_reference_unit(referenceUnit)

hx.reset()

hx.tare()

print("Tare done! Add weight now...")

# to use both channels, you'll need to tare them both
#hx.tare_A()
#hx.tare_B()

while True:
    try:
        # These three lines are usefull to debug wether to use MSB or LSB in the reading formats
        # for the first parameter of "hx.set_reading_format("LSB", "MSB")".
        # Comment the two lines "val = hx.get_weight(5)" and "print val" and uncomment these three

        # np_arr8_string = hx.get_np_arr8_string()
        # binary_string = hx.get_binary_string()
        # print binary_string + " " + np_arr8_string

        # Prints the weight. Comment if you're debbuging the MSB and LSB issue.
        val = hx.get_weight(5)
        print(val)

        # To get weight from both channels (if you have load cells hooked up
        # to both channel A and B), do something like this
        #val_A = hx.get_weight_A(5)
        #val_B = hx.get_weight_B(5)
        #print "A: %s  B: %s" % ( val_A, val_B )

        hx.power_down()
        hx.power_up()
        time.sleep(0.1)

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

図１　参考にしたソースコード

(https://github.com/tatobari/hx711py/blob/master/example.py)

2.2　外部設計

2.1.1　使用器具

調味料入れ、3Dプリンター、接着剤

2.1.2　設計方法

まず、Raspberry Piとロードセルを収納するための外部容器を3Dプリンターで作成した。容器は2つの部品に分けて構成されている。

部品①（図2）は、調味料入れが乗る土台となる部分で、下部にはロードセルを接合する。下図の青い線で示されているところが3Dプリンターで作成した箇所である。また、外部容器とロードセルは市販の接着剤を使用して接着し、一体化している。（今後接着する時は同様に市販の接着剤を使用する。）この部品の工夫点が2つある。1つ目はユーザーが調味料を取り出しやすいよう、外部容器の側面を調味料入れよりも低く設計した点である。こうすることで使用時にユーザーの手が外部容器に当たりにくくなり、不便さを解消できると考えた。2つ目は外部容器の下部の側面がロードセルよりも短く、床と接触しない構造となっている。これにも理由があるが、それは後述する。

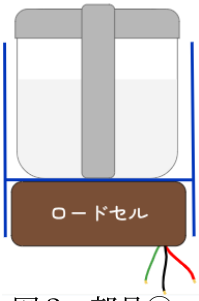


図２　部品②

部品②（図３）は上面に導線を通すための穴を設けており、Raspberry Piを外部容器内に収める形となる。

次に、部品①と部品②を合体させ、部品②とロードセルを接着することで安定性を確保する。これにより、正しく重量を測定できる容器が完成する。

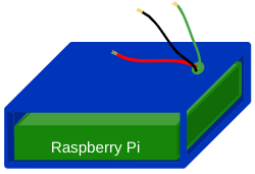


図３　部品②

部品を2つに分けた理由は、調味料入れが外部容器に接触すると重量が分散され、正しい計測ができなくなるためである。これは、体重計で壁に触れると実際よりも軽く表示されるのと同じ原理である。そのため、調味料入れごと調味料の重さを量り、そこから調味料入れの重さを引くことで、接触があっても影響を受けないように設計している。

また、側面の高さをロードセルより数ミリ短く設計した理由は、重さが十分あるものをロードセルに乗せて計測するときに、わずかに沈み込む構造を妨げないようにするためである。これにより、計測の精度を保つことができる。

最終的にサイズ等を微調整し、外部設計は完了である。

3　課題

3.1　液晶表示について

現在調味料入れに付随する液晶に常に重さを表示させるプログラミングを行っているが、まだ文字の大きさや色などを工夫できていないためそこを改善したい。

3.2　デザイン性について

この製品のデザイン性に欠ける箇所をどう改善していくかが課題である。部品①と部品②の間に数ミリ隙間ができ、ロードセルが露出する設計になっているためそのデザイン性を改善していくべきであると考えている。また、調味料入れの下部にRaspberry Piやロードセルを収納しているため縦に長い構造になってしまっている。それをできるだけコンパクトになるように今後改善していきたい。

4　今後の展望

本研究において当該部分の実装が完了したため、次の段階として、取得したデータを単一のデバイスへ送信するプログラムの開発に着手する予定である。

加えて、登録日時を記録する機能を追加することで、冷蔵庫内の食品管理を一層効率化・高度化することも視野に入れている。

5．参考文献

飲んだ量を教えてくれるコースターを作ってみた：
https://note.com/izawa/n/n0b4d4866470a
（2024年6月）

COCORO HOME：
https://cocoroplus.jp.sharp/home/
（2024年6月）