**「熱中症アラート」**

**マイコンプログラミング**

**OK210036　村田 海斗**

**メンバー　宮部、森川、三村**

•概要

今回の課題を通して、熱中症アラートを作成した。これは熱中症の危険度をその場で判断し、注意を促すものである。その危険度はWGBTという暑さ指数として算出される。WBGTには危険度の基準が定まっており、判断された危険度をメッセージとしてLINEに送信し危険度を知らせることができる。またウェアラブル端末として見立てたLCDに温度・湿度を表示する。

1. 背景

マイコンプログラミングの最終課題レポートを作成することにした。グループワークで取り組んだ最終課題は、制作時間が短い。そのため短時間で完成させれるシステムを目指した。アップルウォッチなどのウェアラブル端末は、あらかじめ天気予報をもとに１日の熱中症の危険をユーザーに注意喚起したり表示したりできる。しかしながら、その場の熱中症の危険度を測定しユーザーに知らせるものは少ない。なので、これを目標に取り組むことにした。目的に類似した資料をインターネットで見つけ、内容を見ると自分たちでも実装できそうだったので、作成に至った。また、このようなシステムがあれば自分で注意してても熱中症になってしまうリスクを少しでもなくせ、熱中症でなくなってしまう人を少しでも減らせると考えた。

２．機能

全体の構成と、入出力について順に説明する。

2.1全体構成

本システムでは、LCDディスプレイと温湿度センサーを使用。その場の温度と湿度を取得しWBGTとして算出する。DBGTの値に応じて、表示するテキストを分けておく。そして得た情報（温度と湿度）をLCDに表示する。そして、その情報に該当するテキストをLINEにメッセージとして送信する。しかしLINEに通知するために通知するアカウントのLINEトークンを取得しなければならない。また、温度と湿度が取得不可能であれば、Ｅｒｒｏｒと表示される。発表段階では送信する時間間隔を自分たちで設定した。しかし、この状態では、ユーザーに「ほぼ安全」であることも通知してしまう。なので、発表後の改良点として「注意」以上を感知したときのみLINEに送信されるようにした。また基準についてはWBGTの説明の際に使用する図を参照していただきたい。

2.2　LINEトークン

LINEトークンとは、「LINE Notify」を使用するシステムである。LINE Notifyの公式サイトから「アクセストークン」を発行し、そのトークンを使用することによりWebサービスの登録なしで通知を受け取ることができる。ただし、ユーザーはトークンを設定するために

自分のアプリと連帯する必要がある。

→LINE NotifyはLINE以外のSNSの通知をLINEで受け取る事ができる。PHP・Pythonなどのプログラムとapiを通して自由にLINEに通知を飛ばす事が出来る。

参考資料 >

[LINE Notifyでアクセストークンを発行する - iTAC\_Technical\_Documents (itd-blog.jp)](https://www.itd-blog.jp/entry/line-token)

[トークンとは？キャッシュレス決済の安全を守るその理由 | 【ヒトトキ】三井住友カード (smbc-card.com)](https://www.smbc-card.com/mem/hitotoki/cashless/token.jsp)

[LINE Notifyの使い方を解説！アクセストークンを使ってメッセージを送信しよう | Aprico (aprico-media.com)](https://aprico-media.com/posts/1824)

2.３．WBGT

WBGTとは、暑さ指数である。その指数には基準が与えらえている。今回は温湿度センサ

ーで得た気温をtemp、湿度をhumで変数を置く。

WBGTを導出するためには以下の式が与えられる。

wgbt = 0.735 \* temp + 0.0374 \*hum+ 0.00292 \* temp \* hum – 3.4516

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

2.3．LCD

LCDは、ウェアラブル端末と見立てて使用。

2.4．センサ

温湿度センサを、port7につなげた。

2.5.　LINEに送信するためのテキスト

危険（31以上）→暑さ指数　危険

厳重警戒（28～31）→暑さ指数　厳重警戒

警戒（25～28）→暑さ指数　警戒

注意（25未満）→暑さ指数　注意

1. 実装

3.1全体

本システムのLINEトークン取得方法、WBGTの実装、GrovepiでのLCDと温湿度センサ、通知を送る頻度について順に説明する。

3.2 LINEトークン取得方法

<https://notify-bot.line.me/ja/>　に通知されるユーザーのLINEアカウントでログインする。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

右上のメニューの「マイページ」を選択する。

グラフ

中程度の精度で自動的に生成された説明

マイページ最下部にトークンを発行するボタンがある。

グラフィカル ユーザー インターフェイス, アプリケーション, Teams

自動的に生成された説明

トークンを発行する際に通知先のグループトークを選択。そして発行。

グラフィカル ユーザー インターフェイス, アプリケーション, Teams

自動的に生成された説明

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ

自動的に生成された説明

「LINE Notify」のアカウントを追加するとこれから通知が来るようになる。

3.3　WBGTの実装

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

wgbt = 0.735 \* temp + 0.0374 \*hum+ 0.00292 \* temp \* hum – 3.4516

この式より値を求める。そしてその値がどの基準に該当するかを定めるために、この表通りに場合分けを行う。

文字の書かれた紙

自動的に生成された説明

これによりWBGTの定められた値が４つのうちの１つに該当するようになる。

3.4　、GrovepiでのLCDと温湿度センサ

今回、GrovepiにLCDと温湿度センサを接続。温湿度は、port7に接続。

電子機器の部品

中程度の精度で自動的に生成された説明

3.5　送信頻度

発表時は、設定した時間おきに取得した情報を送信していた。しかしながら、それでは直感に反する行為であり、必要な情報だけを送信するべきであった。そのためユーザーに通知するために「安全」である場合は送信しないシステムに改善した。

４．実行結果

電子機器の部品

中程度の精度で自動的に生成された説明

テキスト

自動的に生成された説明

テキスト, 手紙

自動的に生成された説明

５．考察

今回の課題を通して、発表の際に先生から「重要な基準値」のみでの通知にすることを指摘していただいて、それはすぐに改善することができました。しかし、危険度に応じて通知が多くなるなどの機能を追加で入れることができれば危険度合がユーザーにとってわかりやすいので、それを追加実装したかった。また、LCDも同様で、危険度に応じて色を違う色にし一定の間隔で光らせる追加機能ができれば成果物の理想に近づけれると思った。この成果物のようにデバイスがその場で危険度を取得しユーザーに知らせれたら、なかなか気づけない高湿度の場所での熱中症を未然に防ぐことができるので、ウェアラブル端末機器に搭載されるべきである。

６．感想

ある程度質のある成果物を作れてよかった。指摘されたことをすぐに取り組んで改善できた時はうれしかった。LCDの危険度に応じて色を分けるのはできるのにやりきれなかったのが悔しい。またLCDに加えLEDの点滅を利用して危険度が高いほど、点滅の速さが速いなどの基準に応じた機能を加えたかった。追加で熱中症対策もテキストに加えるべきであった。今回を通して、成果物を作る喜びを一番感じ取れた講義になった。今後、ラズパイを使って、さまざまな自作を行いたい。

→ラズパイの参考書やサイトなどおすすめがあれば教えてください。本屋に行ってもいまいちピンときたものがありませんでした。僕の理想は、「R２D２」みたいなポンコツマシーンとかいたずら機器を作りたいです。