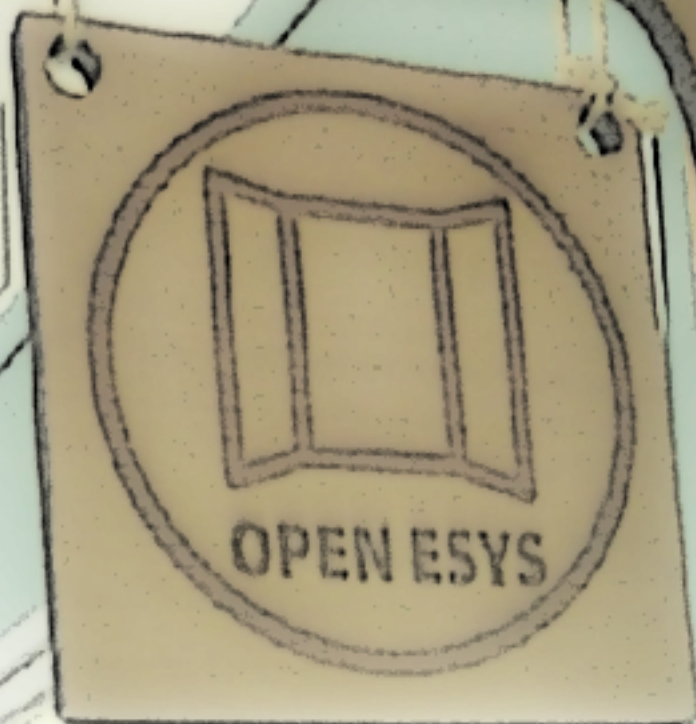


OpenESys



新春号



工学の
新しいカタチ

Programming
Camera
Audio
Compiler
etc...

Index

★★★ Topic1 台湾行ってきました p5



Column 便利なショートカットコマンド(1/2) p8



★★★ Topic2 広角レンズを買ってみた p9

★★★ Topic3 オーディオの記事 p13



Column 便利なショートカットコマンド(2/2) p18



★★★ Topic4 オイル交換はいかがですか??? p19

Column 大手キャリアから格安SIMに乗り換えて留年する(1/3) p22



★★★ Topic5 ビムライフ! p23

```
int main(int argc, char const *argv[])
{
    printf("Hello World.");
    return 0;
}
```

★★☆ Topic6 Simple Programming p29

★★★★ Topic7 Java でピアノを作る p35



★★★★ Topic8 ESP-WROOM-02 で遊んでみた p39



Column 大手キャリアから格安 SIM に乗り換えて留年する(2/3) p44

★★★★ Topic9 Web ブラウザで動画を作ろう! p45

```
1 var vid = new Whammy.Video(frames);
2 vid.add(canvas);
3 vid.compile();
```

Column 大手キャリアから格安 SIM に乗り換えて留年する(3/3) p48

Column Slack を使おう(´・ω・`)!!!!!! P49

とっつきやすさの目安として以下のような指標を用意したので参考にしてください

☆☆☆……趣味レベル

★★☆……工学かじったことある人 or これから工学やってみたい人向け

★★★★……強い人向け



台湾行ってきました

こんにちは。OpenEsys の工学しない担当さいぞーです。創刊号ということで何か面白い記事でも書こうかと思ったのですが、特に面白い題材もなかったなのでこの間台湾に行ってきたときの旅行記を書きたいと思っています。

年明けもはやから台湾に行ってきたわけですが、ここでは特にヘンテコだった二つのスポットについて紹介します。

□九份



首都台北からバスで約一時間、山の上にある街九份に行ってきました。ここは映画のロケ地だったりかの有名な「千と千尋の神隠し」のモデルになった街（らしい）だったりするので日本人が多かったです。

やはり一番の見所は建物。建築様式も材料もバラバラな建物が乱立しており、とにかくしっちゃかめっちゃかな状態に。

写真の建物が九份のシンボリック存在で、美しい景色の見える喫茶店でした。ちょっと一服でもしようとメニュー表を見ると急須のお茶が一人 **300 元（約 1200 円）**。帰った。



↑ 中華風なのか洋風なのか・・・

他に九份を代表するものとして、長い階段があります。映画「非情城市」にも出てきた場所らしく、台湾のキラキラ女子大生たちが自撮り棒をもって写真撮影していました。

階段を上りきった先には寺院があり、これもなかなかキラキラしていて目に付くも

のでした。

台北からのアクセスもさほど悪くないので、台湾に行く機会があれば是非行ってみてはいかがでしょうか。



↑九份の中央を横切る階段

□慈湖紀念雕塑公園

皆さんは蒋介石という人物をご存知でしょうか。毛沢東率いる中国共産党と対立し現在の台湾の礎を築いた、いわば台湾建国の父なる人です。それほどまでの人物なので台湾では彼の銅像を非常によく見かけます。しかし台湾建国から100年以上が経ち、人々の間では「もうそろそろ蒋介石の銅像作らなくてもよくね？」という風潮が声に出さずとも広がっていきました。

そこで作られたのがこの公園です。国中から追い出された寄贈された蒋介石の銅像を一同に介し展示しています。公園の様子がこれ。



蒋介石しかいねえ！

どこもかしこも蒋介石。座った蒋介石、直立した蒋介石、馬に乗った蒋介石、子供と戯れる蒋介石・・・

あれも蒋介石？いやあれは**通行人**だ。

といった感じで蒋介石がゲシュタルト崩壊しそうな空間。いい感じに霧に包まれていて本当に現世にいるのか怪しくなりました。

中でも面白かったのが奥の方にあったジャイアント蒋介石で、作りかけなのかどこから外したのか**体がバラバラ**に。ドックで修理中の**ロボット兵**か何かかと。



↑ジャイアント蒋介石。10メートルくらい

☆☆☆ Topic1 台湾行ってきました

こちら台北からバスを乗り継いで1時間くらいなので、蒋介石が好きでたまらない人は行ってみると幸せになれると思います。

というわけで台湾旅行記、いかがでしたでしょうか。台湾は日本に近く、比較的安価で行くことができるのではじめての海外旅行などにうってつけだと思います。この記事を読んで台湾に行ってみたい！思った人がいてくれたら幸いです。

ご精読有難うございました。

Column 便利なショートカットコマンド(1/2)

こんにちは、esys14 のこたろといいます。

突然ですがあなたはレポートなどで文章をコピー&ペーストをするときどうしていますか?? 右クリックでメニューを出してコピーをクリックしてまたメニュー出して貼り付け……みたいなことやっていませんか?? **そんなあなた、必見です。**このコラムではコピペやネットでのワード検索、はたまた Microsoft の Word 等で文字を大きくしたり小さくしたりする動作を、マウスを使わないでキーボードだけでやってみよう!!という内容です。これを覚えればいつものレポートが倍近く早くなるでしょう(※上達度は個人差があります)。

注意!……このコラムでは Windows での動作を元に書かれています。他の OS(Mac や Linux)で同じ動作をするかわからないのでそこはご了承ください。

さて、ショートカットコマンドを使うにあたってぜったいに必要なキーがあります。(例外も幾つかありますが……)

「**ctrl(コントロール)キー**」です(controlの略ですね)。一般的なキーボードなら右下、左下に一つずつあるかと思います。多くのショートカットコマンドはこの ctrl キーを押しながら他のキーを押します。

もう 1 つ覚えておきましょう。「**shift(シフト)キー**」です。ctrl キーの近くにあります。このキーは Word や Excel で重宝

します。

説明が長くなりましたね。早速紹介していきましょう!

まずはみんな大好きコピペシリーズから

ctrl+c……コピー

ctrl+v……ペースト

ctrl+x……切り取り

これらのコマンドは文章をドラッグ選択して使うのはもちろん、ファイル内のデータの移動にも使えます。「え、これ押すだけ??」そうです、これだけです。どうです?どんどん知りたくなってききましたか?

次はデータの保存、やり直しなどのコマンドです。

ctrl+s……データの保存

ctrl+z……一度操作したものを元に戻す

F12……名前を付けて保存

F2……(ファイルを選択している状態で)

ファイル名の変更

F キーシリーズが出てきましたね。F キーはそれ自体でショートカット機能を持っています。F5 キーでホームページの更新をするのはみなさん一度はやって見たことはあるのではないのでしょうか。

さて、次のコラムでは Word を用いてマウスを使わないでレポートを書いてみるとうしましょう!!(2 に続く)



こんにちは。今回工学システム学類の学類誌を発行するということで、難しいこととか書けないしどうしようかなーと悩んでいたところ、ちょうど新しくカメラのレンズ「SIGMA 10-20mm F3.5 EX DC HSM (NIKON 用)」という広角レンズを買ったのでそれについて書いてみようと思った次第でございます。

※注意

- ・ 私が普段使用する一眼レフカメラは NIKON D7000 です。したがって、今回ご紹介するレンズも NIKON 用です。
- ・ 私の所持レンズは単焦点レンズ「AF-S NIKKOR 50mm F/1.8G」と今回の広角レンズの2本です。
- ・ 今回この記事で掲載する写真などの著作権は全て筆者にあります。まあ割りとうでも良いです

□購入！

今回、レンズは amaz○n さんから購入い

たしました。ちなみにお値段、税込 47,550 円です！！（2015 年 12 月 21 日購入時）

.....まあカメラのレンズにしては普通の値段です。むしろ、安い部類のレンズと言えるかもしれません。（一眼レフのレンズって高いんですね.....）

□開封！

さてさて、amaz○n さんから届いたものがこちらになります。



箱は黒を基調にした感じです。カッコいいですね。

早速開けてみます。

☆☆☆ Topic2 広角レンズを買ってみた



箱の中にはレンズ本体、花型フード、レンズポーチ、説明書が入ってありました。

レンズを一目見て意外と大きいなと思いました。それもそのはず、今まで使用していたレンズは「AF-S NIKKOR 50mm F/1.8G」という単焦点レンズのため、それと比べると今回の広角レンズは随分と大きいです。ちょっと比較してみました。



左が単焦点レンズ、右が今回紹介する広角レンズ

こうして見ると、1.5倍くらいですかね。

□新レンズ観察！

新レンズを隅々まで見ていきましょう。このレンズは焦点距離が10mmから20mmまで変化させることができます。簡単に言えば、数字が小さい方が画角の広い写真を撮ることができるということです。このレ

ンズはかなり画角の広い写真を撮ることが出来ます。10mmの画角は超広角の部類に入るレンズと言えますね。



レンズの球体部分が少し飛び出ています。しかし、レンズフィルター（プロテクター）は問題なく付けられるそうです。（私はまだ付けてないです）フィルター径は82mmです。

□装着！

カメラ本体と装着するとこんな感じになります。



うん、やっぱりいいですね～

手にとってみると、重すぎなく、心地よいくらいの重さが手に伝わってきていい感じですよ。さて、写真を撮ってみようとファインダーを覗いてみて驚きました。今まで50mmの単焦点レンズを使っていたので、

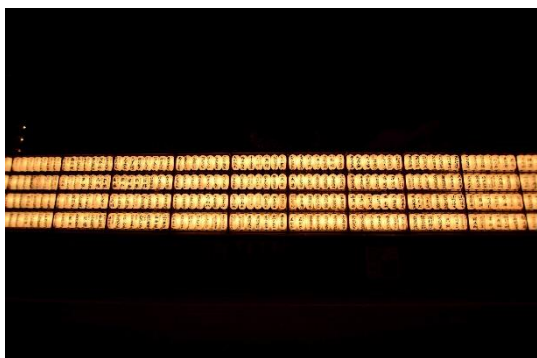
☆☆☆ Topic2 広角レンズを買ってみた

ファインダーからの景色は少し中望遠気味に見えていました。しかし、このレンズの場合、自分が直接見る視界よりも広い風景が広がっていました。焦点距離 10mm ではその風景がさらに広がり、こんなに広い画角で写真を撮れるんだとびっくりしちゃいました。

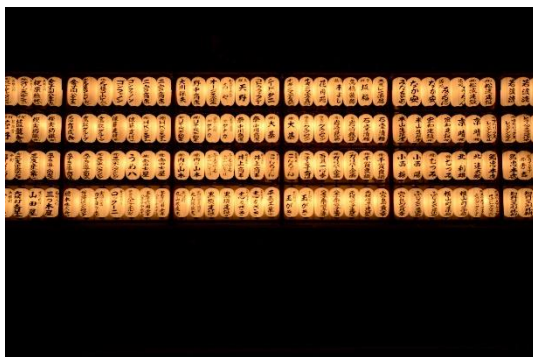
□明るさと画角！

今回の広角レンズ「SIGMA 10-20mm F3.5 EX DC HSM(NIKON 用)」の最大の魅力は焦点距離 10mm から 20mm を通して F 値を F3.5 に保ったまま撮影できるということです。つまるところ、暗いところでもいい感じの写真が撮ることができるということです。(説明が雑ですみません。)

今回の広角レンズは 10-20mm のズームレンズであります。果たしてどのくらい画角が違うのか？こちらの画像はほぼ同じポイントから焦点距離を変えて撮った写真です。



↑焦点距離 10mm(F3.5)

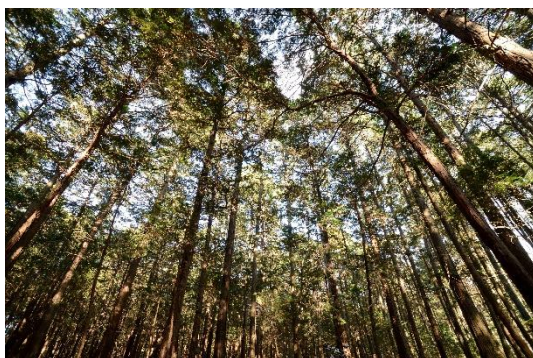


↑焦点距離 20mm(F3.5)

ご覧の通りこれくらい画角が変わります。F3.5 ではこのように暗い場所でも問題なく写真を撮ることができます。

□パース！

広角レンズには特徴的な歪み、パース(パースペクティブ)が出ます。パース(遠近感)はざっくりと言うならば、近くものは近く、遠くものは遠くに見えるということで、広角レンズの場合これが際立ちます。



例えばこちらの写真は画面下端の木々が斜め上に向かって伸びているように写ります。広角レンズはパースの扱いが難しいレンズとされていますが、これがビシッと決まった写真はとてもかっこいいのです。(私

はそんなじつと決まった写真はまだ撮れません...)

□最後に

とまあこんな感じで、今回購入した「SIGMA 10-20mm F3.5 EX DC HSM (NIKON 用)」は素人目線ですが、とても良いレンズだと思いました。他にも色々いい所、悪い所等あると思いますが私の拙い知識ではお伝えすることができません.....

続きはご自身でどうぞ、ということでこれを機にこちらのレンズ試してみたいかでしょうか...？

雑な紹介記事ですみません。私もこのレンズと共にこれからどんどん技術と知識を身につけていきたいと思います。



オーディオの記事

こんにちは、2 年次のいとうです。今回は創刊号ということで、何を書けばよいのかな……?と本記事を何についての記事にするか迷いましたが趣味の一つであるオーディオについて書きます。

□オーディオの世界観

さてオーディオ趣味にはポータブルオーディオや据え置きオーディオ、カーオーディオなどいろいろなスタイルがあって、またイヤホンやヘッドホン、スピーカーを使うものなど、“オーディオ”と一括りに表現してもたくさんの楽しみ方が存在します。さらに、低音が強い音が好きな人やボーカルがよく聴こえる音が好きな人、全体的な音楽の空気感を楽しみたい人など、音の好み人がそれぞれでそこがオーディオの面白い点でもあります。しかし、オーディオの世界において普遍的な目標を挙げるとすれば、“良い音の追求”はまさしくそれと言えるでしょう。個人によって“良い音(好きな音)”の定義は異なると思いますが、オーディオを趣味にしている人は全てその人のこだわりを持って、良い音を追求している

のです。良い音を実現させるためには良い電源、良いケーブル、良いアンプ、良いスピーカーなど、とてもたくさんの AV 機器やケーブル類が必要になってきます。その中でも高品質と呼ばれるものはケーブル一本とっても何万円とかかってしまうのです。中には高級 LAN ケーブルや高級 SD カードを使う人もオーディオの世界には存在します。この記事では、そういった高音質を追求するにあたってオーディオ・ファイル(オーディオオタク)が使う製品やツールについてジョークを交えて紹介します。

ここで一つ、とても重要なオーディオの掟をここに記しておきます。オーディオの世界では“データや仕組みがよく分からなくても、音が変わったならば効果がある”と判断することです。



（定価230,000 円）
高音質ケーブル

□本気で良い音を聴きたい人のためのオーディオグッズ

このトピックでは、本気で良い音を楽しみたい人のためにさまざまなオーディオグッズを紹介します。オーディオの世界では常識的なものから、ちょっぴりマニアックなものまで筆者が面白いと思うものを取り上げます。ジョークを交えて文章を書いてみますのでどうか最後まで読んで頂ければ幸いに思います。

▼インシュレータ

インシュレータにはスピーカーやアンプの下に 3 つか 4 つ置いて音響機器の振動を抑える目的があります。振動の吸収を目的としたものや、振動を速やかに外部に逃がすことを目的としたものがあり、材質は金属や石材などさまざまです。普通はオーディオボードと呼ばれる専用の台の上に設置します。物理的な振動の変化は確実に存在するので音は変わると思います。聴き手が良い音になったと思うのなら良い音に変化するのでしょう。これは据え置きオーディオなどする場合には基本的なアイテムになります。



▼ローズバイブレーション

発売されたのはだいぶ前ですが、最近良く話題に上がる製品です。最近になって某イヤホン専門店で新規取り扱いを開始したためでしょう。

使用方法と目的についてですが、これは DAP(Digital Audio Player)に貼ったり、アンプ内部の真空管の頭に貼り付けたりして使うそうです。窓ガラスに貼る人もいるのだとか。このローズバイブレーションを貼ることにより対象物の振動を抑えます。目的はインシュレータに似ていますね。実際に DAP に貼ることで出力される音に変化が出たり、真空管に貼り付けることで音のキレが良くなったりするようです。真空管については物理的な振動によりノイズを生じるものですので、ローズバイブレーションによる音質向上は理にかなっています。(窓ガラスの制振については言及しないでおきます……。)

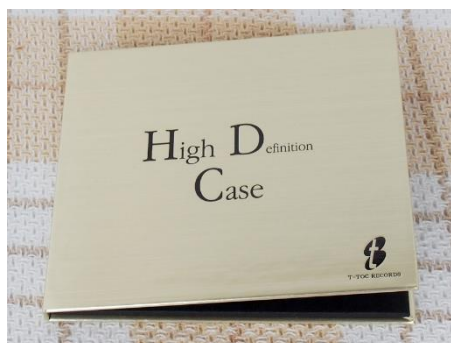
私がツイッターでフォローしている方が何人か購入なさっていてレビューを見ることがあるのですが、微細ながら本当に音質は変化するようです。中にはお尻に貼ってるフォロワーさんもいらっしゃいました。大地を震わせるような躍動感のあるおならが出るようになったそうです。

▼高音質 CD ケース

これは下記に製品の画像を掲載します(掲載許可を頂きました)。一見、何の変哲もない CD ケースに見えますが、実はとんでもない“魔法”の CD ケースです。

☆☆☆ Topic3 オーディオの記事

表面には High Definition Case と記載され、光沢も見られ、高級感が感じられます。何やら高音質の予感がします。



いったいこの CD ケースにはどのような目的、効用があるのでしょうか。上図の写真のように、本当に普通の CD ケースに見えますが、なんと、「ディスクをこの CD ケースに入れるとディスクの音質が向上します。」(メーカー公式ページ参考)

具体的には、ディスクを 30 秒～60 秒間程度この CD ケースに入れるとディスクの音質が向上してダイナミックレンジ、立体感、奥行きが飛躍的に拡大するそうです。この CD ケースの効果の原理については、

”トレー内部に特殊イオンコントロール素材と特殊生体エネルギー素材を 7:3 でハイブリッド採用していることで、ディスクに蓄積している電磁波、静電気、磁場等の悪影響を中和する”、とメーカー公式ページに明記されております。私の知識不足もあってか CD に何が起ころのか全く理解で

きませんが、ユーザーのレビューには「CD ではデジタル臭い高域のカサカサ感が無くなり、ボーカルに艶が出る。」「SACD で高音質盤を聴いたら、奥行きが広く感じた。ブルーレイディスクは輝度が上がり、画に奥行きが出る。」など複数の歓喜の声が確認できます。どうやら CD 以外の光学ディスクにも効果は十分に得られるようですね。私は遠慮しておきますが、興味のある方は是非、購入を検討されると良いかと思います。その時はレビューをお聞かせ願えたら幸いに思います。

▼ミュージックストーン

これは下記に製品の画像を掲載します(掲載許可を頂きました)。個人的に、ミュージックストーンはこれまで紹介したどの製品よりも画期的な手法で音質向上に成功しているものだと考えています。よってこの製品を最も熱く紹介し、最後の製品紹介にしたいと思います。

音楽用光学ディスクは現在まで幾多の変遷を辿って現在の CD ディスクが普及しました。(現在はブルーレイディスクで音楽を聴くこともある。) また、普及させることは全く考えず、ポリカーボネイト、プラスチック、蒸着金属、硝子など多くの素材を使用して有利な CD ソフトを開発しようとしたオーディオファイルもいらっしゃいます。

しかし、例えば SHM-CD(Super High Material)と銘打った CD 盤はポリカーボネイトの透明率を上げる事によって高音質になったという宣伝文句を度々目にしますが、この雑誌を読んでいる理工学群や情報学群

に所属する皆さんにとっては”CD の原理を考えれば材質による音質の変化は考えられない”と思うはずです。私もそう思います。まあ、材質によってデータが変わってしまっただけは記憶媒体として CD は欠陥品であり、一般的に流通させる記憶媒体としては無価値なものとなるのでこれは自明ですね。CD より音質が良いとのことでは有名になった SACD(Super Audio CD) は DSD(Direct Stream Digital)方式を取ることで従来の PCM 方式と比較して音質の向上に成功したわけで、材質によるものではありません。DSD 方式はオーディオに興味がある人ならば一度は耳にしたことがあるでしょう。ポータブルプレーヤーで再生が可能であるものが複数登場していますし、ポータブルオーディオプレーヤーも個々数年で急激に発展しました。

おっと、ちょっと話が脱線してしまいましたね、まあとにかく私は CD について、多少存在する可能性は否定はしないと考えるが材質による音質の変化は認めたくない、と考えています。

話を本題に戻しまして、このミュージックストーンの目的、効用を紹介します。なんとこのミュージックストーンは、前述のとおり私は認めたくはありませんが CD の材質により CD の音質は大きく変化するという前提の上で、”それらの CD の材質による音の変化を一瞬で天然石の振動に書き換える”ことを目的としています。ここで一旦この目的について整理しましょう。上記の目的をもう少し噛み砕いて表現すると、「CD は材質により大きく音質が変化する

(デジタルデータの数値が変化する)が、ミュージックストーンを CD に接触させると材質による音の変化を一瞬で天然石によって生じる振動に変換することができる。」といった感じですかね。ちょっとどう表現しようか考えこんでしまいました。多分、CD の振動が天然石になると思うんですけど (名推理

ここで一つ、提案をしたいと思います。CD のデジタルデータを書き換えるのは難しいと思いますが、レコードにこの石を乗せたら、レコードが天然石になり、音質が向上する可能性があるかもしれないという提案です。レコードなら材質によって音質は変わると思いますし、多分、レコードにこの石を乗せたらレコードは天然石になると思うので、音質の向上が期待できるでしょう。ほら、最近超高強度コンクリートでアナログレコード作ったドイツのエンジニアさんとかもいらっしゃるしね！！エンジニアの Recardo 氏の Facebook アカウントの URL を参考に記載します。ご興味のある読者諸君は是非見ると良いと思います。

(<https://www.facebook.com/bln.fm/video/s/10154411340333572/>)



また、メーカー公式サイトにはこの天然石をプレスレットのように使用すると、自然界の波動を全身に浴びて苛々を抑えリラ

☆☆☆ Topic3 オーディオの記事

ックスすることが出来るとも書いてあります。あなたが最近イライラしているなら是非おすすめの一品として挙げたいと思います。これもうわかんねえな。



ここまで、今回は4つの製品について紹介しました。実際このうち私が使用しているものは一つのみになりますので、この記事を読んで製品の購入を決定したという方がいましたら、是非私にご一報していただくと幸いに思います。特に私は高音質CDケースが気に入っています。生体エネルギー素材というものを感じ取りたいのです。

では今回の記事ではこの辺でおしまいに

します。あまり中身のある記事がかけたとは思いませんが、第2号からはポータブルオーディオについてのバランス出力の意義についてや、ケーブルによって音質が変化することを証明することについてなど工学的な記事を執筆したいと考えておりますのでお楽しみに待っていてくださると嬉しく思います。

Column 便利なショートカットコマンド(2/2)

さあ、ここからは Word に備わっている便利なショートカットを紹介しましょう。

ctrl+b……太文字にする

ctrl+i……文字を斜体にする

ctrl+e……中央揃え

ctrl+shift+<……文字を小さくする

ctrl+shift+>……文字を大きくする

ctrl+shift+f……フォントの変更

ctrl+n……新規ファイルの作成

これらのコマンドがあれば簡単なレポートは書けそうですね。実際にやってみましょう。

まず Word で新しい文書を開いてください。例えば「タイトル」と打ち、その下に名前誕生日など適当に文字を打っておきます。「タイトル」と書いた文字を、shift キーを押しながら方向キーで選択してください(わからなかったらマウスで選択)。選択が終わったら「**ctrl+shift+>**」で文字を大きくします。好きな大きさにできたら「**ctrl+b**」で太文字にして「**ctrl+e**」で中心に持ってきてみましょう。どうですか?? ctrl キーに指を置いたまま他のキーを押すだけでそれっぽいものができました。いちいちマウスでアイコンをぽちぽちするより楽だと思いませんか??(※意見には個人差が((ry)

今度はタイトルの下に書いた文字をコピーしてみましょう。先ほどと同じよう

にコピーしたい文字を選択し、「**ctrl+c**」でコピー、「**ctrl+v**」でペーストしてみましょう。「**ctrl+v**」を押すたびに同じ文字列が出てきますね。コピーがこんなに簡単にできるなんて素晴らしいです。これで簡単に論文の製作が……なんでもないです。

最後にこのファイルを保存しましょう。保存をするコマンドは「**ctrl+s**」でしたね。これで保存ができました。レポートの保存もいちいちフロッピーディスクをクリックしなくて済みますね。

どうでしたでしょうか。少しレポートが捗りそうな気がしましたか?「おおまかな説明でなんもわかんねえよ!」という人、ごめんなさい。文章力高めます。「少し興味持てた!」という人が 1 人でもいたら嬉しい限りです。

ショートカットコマンドはもちろん検索すればたくさん出てきます。Word や Excel はアイコンにカーソルをのせるとその操作のコマンドが表示されるものがあります。自分がよく使うコマンドを一通り覚えてパソコンを便利に使ってみては??



はじめまして、OpenEsys のサーバー管理担当の esys15 のさばです。昨年の暮に念願の車(ek ワゴン)を購入した後、その翌日に 2 泊 3 日の京都旅行に行ったり大洗や鹿島に釣りに行ったりしてなんだかんだ結構距離を走りました (約 4,500km)。ということで今回は自動車のエンジンオイルの交換レポートを記事としたいと思います

□エンジンオイルって何それ美味しいの???

エンジンオイルはエンジンの冷却、洗浄、防錆、潤滑、密閉など様々な役割があります。

エンジンオイルは食用油と同様に酸化します。またエンジン内部の摩擦によりゴミやカスがオイルに混ざります。オイル交換せずに何万キロも走っているとエンジン内部に汚れが溜まり、最悪の場合エンジンがかからなくなるケースがあります。

このような事態を防ぐためにもエンジンオイルは定期的に交換しましょう。

交換の目安は自動車の取扱説明書に記載してありますのでそちらを参考にさせていただきたいのですが、おおよそ 5,000km また

は 6 ヶ月の短い方で交換が勧められていると思います。ターボ車、軽自動車はエンジンオイルが汚れやすいため短めの交換期間が設定されています。

□Let's オイル交換!!!

それではオイルを交換していきましょう!

必要なものは、オイルチェンジャー、新しいオイル、オイルジョッキ、捨てるオイルを入れるための袋とバケツなどです。



まず、現在のオイルの様子を調べてみま

★★★ Topic4 オイル交換はいかがですか???

す。エンジンルームを開け、オイルレベルゲージ（リング状の細い鉄の棒）を引き抜き、オイルを拭き、もう一回元の場所に刺して抜きます。



オイルがレベルゲージの上下の凹みの中間ぐらいのところまで付着していますね。量はこれで良いようです。もしレベルゲージの上の凹みより上までオイルがあったらオイルが多すぎ、下の凹みより下しかオイルがなかったらオイルが少なすぎということになります。オイルレベルを測るときは必ずエンジンを止めて数分たってからにしましょう。そうしないとオイルパンにオイルが落ちきらず、少なめにレベルが測られることになってしまいます。(これを理解していないENEOSの店員が「オイル量が少ないので足したほうが良い」と教えてくれました。)

それでは5分ほどエンジンをアイドリングさせましょう。オイルを温めることにより、抜き取りやすくなります。

次に、オイルチェンジャーをセットします。私が購入したのは電動タイプで、他には負圧式（ポンプでタンク内気圧を大気圧より下げてオイルを抜き取る方式）などがあります。



この時吸い出すオイルを回収するために中に袋を敷いたバケツを用意すると良いでしょう。間違ってもオイルを下水に流してはいけませんよ（環境保護）。

吸入用ホースをオイルレベルゲージに、排出用ホースをバケツの袋の中に突っ込み、バッテリーにチェンジャーの端子を接続してスイッチをオン！

チェンジャー「ズゴゴゴゴゴゴ！！！！」

泥水色のオイルはが勢い良く吸い上げられ、バケツの中の袋に溜まっていきます。その袋を外側から触ると生暖かく、バズみを感じざるを得ません。5分ほどで勢いが弱まり、10分でほとんど出なくなったので吸い出し作業を終わらせました。

チェンジャー一式を片付けた後、新しいエンジンオイルをジョッキに移し、オイル投入口を開け、注ぎ込みます。この時、勢いが良すぎると溢れてしまうのでご注意ください。

入れ過ぎるとエンジンによくありませんのでオイルを入れながら時々レベルゲージ

★★☆ Topic4 オイル交換はいかがですか???

で量をチェックすると良いでしょう。



以上でエンジンオイルの交換が終了しました。

□まとめ

今回は上抜きというレベルゲージからオイルを抜く方法でしたが、他に下抜きというジャッキアップしてオイルドレンからオイルを抜く方法もあります。

上抜きではオイルに含まれるゴミを回収するオイルフィルターが交換できないので、2 回に 1 回は下抜きをして同時にオイルフィルターを交換しましょう。

□最後に

DIY はいいぞ。

Column 大手キャリアから格安 SIM に乗り換えて留年する(1/3)

こんにちは、工学システム学類 3 年になります、いとうと申します。

本コラムは、大手キャリアから格安 SIM に乗り換えて留年すると銘打ったものになりますが、決して留年を推奨するコラムではないということを始めに断っておきます。

さて、読者の皆さんは携帯電話をどの会社に契約していますか？

ドコモや au、ソフトバンクなどの大手キャリアに契約している方が多いことと思います。

私もつい先日(3 日前くらい)まで au で携帯を契約していました。

大手キャリアには安心感がありますよね。家族割りなどを利用し、一家みんな共通の大手キャリアと契約しているというご家族も少なく無いと思います。

しかし、この記事を読んだあと、その安心感は大手キャリアへの疑念と不信感に変わるでしょう。

(※決して A 社、D 社、S 社へのネガティブ・キャンペーンの意はありません。)

では、モバイル SIM 各社への乗り換えについての基本情報を以下にまとめます。

(2 に続く)

```
1 # つくば市ごみの日bot (非公式) ◀
2 ◀
3 スマートホンの方は下の 'View all of README.md' をタップしてください。◀
4 ◀
5 ## 注意! ◀
6 ◀
7 この bot は個人の趣味で運営している非公式の bot です。◀
8 この bot に関する質問等はつくば市ではなく、◀
9 [@high_moctane](https://twitter.com/high_moctane) に◀
10 してください。◀
11 よろしくお願ひします。◀
12 ◀
NORMAL | devel > README.md | unix < utf-8 < markdown | 3% | 10:1
```

□はじめに

この記事を担当する はいもくたん です。

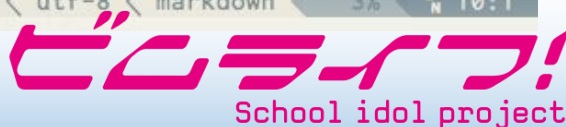
題名は「ビムライフ!」ですが、今回の内容は「ラブライフ!」とは関係ありません(´・ω・`) 期待させちゃってごめんね(´・ω・`) ピンクポンポン(´;ω;`)!

今回はプログラムや HTML を授業などで書いたことがある人、パソコンでレポートを書いたことがある人などにオススメの Vim について書いてみたいと思います(´・ω・`)

□テキストエディタとは(´・ω・`)?

テキストエディタはご存知ですか? パソコンで文章を書くための便利なソフトウェアのことです。

文章を書くというと、例えば Word や一太郎などが思い浮かびますが、ここでいう「テキストエディタ」はちょっと違います。ワープロソフトのように文字を太くしたり下線を引いたりして文書を作るためのソフトではなく、本当に文字を打ち込むだ



けのソフトウェアです。Windows を使っている人なら「メモ帳」、Mac を使っている人なら「テキストエディット」が標準装備の「テキストエディタ」です。

「メモ帳」や「テキストエディット」は大した機能を持っていないのですが、ここで紹介するテキストエディタは非常に高機能です。画像編集ソフトでいえば「ペイント」と「Photoshop」くらい違います。

□テキストエディタの用途

テキストエディタは主にプログラムを書く人に使われています。プログラム以外でテキストエディタを使うことは一般的ではありません。しかしテキストエディタを使うと文章を書くのが本当に楽になるので、使い慣れている人はあらゆる場面でテキストエディタを使用します。例えば、授業のノートやレポートの作成、果ては Twitter などなど……。他にも小説を書く人がテキストエディタを使うこともあるようなので、パソコンで文章を書いたことのある人には

無縁の話ではなさそうですね(´・ω・｀)

テキストエディタに興味湧いてきた方は是非インターネットで調べてみてください。文書作成が今までと比較にならないほど楽になりますよ(´・ω・｀)

□いろいろなテキストエディタ

さて、テキストエディタにはどんなものがあるのでしょうか？ 前述したメモ帳、テキストエディットの他に、「Emacs」

「Sublime Text」「mi」「Gedit」「Atom」「nano」「サクラエディタ」「vi」…… などなど、こ

タが存在します（ここに書いてないからといって怒らないでください(´; ω ;`´)）。今回はその中で「Vim」というテキストエディタを紹介します(´・ω・｀)

□テキストエディタ「Vim」(´・ω・｀)

Vim は vi というテキストエディタをもとに改良されてできたエディタで、かれこれ 20 年以上メンテナンスされてきています。見た目はこれ以上ないほど簡素ですが、中身は非常に多機能で高速にテキストを書

```

1 + Desktop/
2 - Documents/
3 | + Materials/
4 | - Projects/
5 | + high-moct..nippet_snippets/
6 | + nonogram-solver/
7 | + test_repo/
8 | - tsukuba-gomi-bot/
9 | + bin/
10 | + config/
11 | + db/
12 | - lib/
13 | | - bot/
14 | | | bot.rb
15 | | | garbage.rb
16 | | | message.rb
17 | | | project.rb
18 | | | bot.rb
19 | | | extend_date.rb
20 | + log/
21 | + src/
22 | | Gemfile
23 | | Gemfile.lock
24 | | LICENSE
25 | | README.md
26 + Downloads/
27 + Dropbox/
28 + Google_Drive/
29 + Library/
30 + Movies/
31 + Music/
32 + Pictures/
33 + Public/
34 + Tsukuba_Univ/
35 + VirtualBox_VMs/
36 week01.md

5 require "pp"
6
7 require_relative "project"
8
9
10
11
12 # Bot module
13 module Bot
14   include Project
15
16   # twitterアカウント関連のことをする
17   class Bot
18     @@ = Project
19
20     attr_reader :twitter, :stream
21
22
23   def initialize(app_name, stream: false)
24     keys = YAML.load_file(
25       @@P.root_dir + "config/#{app_name}_keys.yml"
26     )
27
28     @twitter = init_twitter(keys)
29     @stream = init_tweetstream(keys) if stream
30
31     rescue => e
32       @@P.log.fatal($0) { @@P.log_message(e) }
33       raise
34     else
35       @@P.log.debug($0) { "#{app_name} のBotインスタンス生
36         > 成" }
37     end
38
39     # 普通のツイート
40     def update(string, id: nil, screen_name: "", try: 3, fo
41       > oter: "")

```

↑ Sublime Text 風にもできる(´・ω・｀)

ここには書ききれないほどのテキストエディ

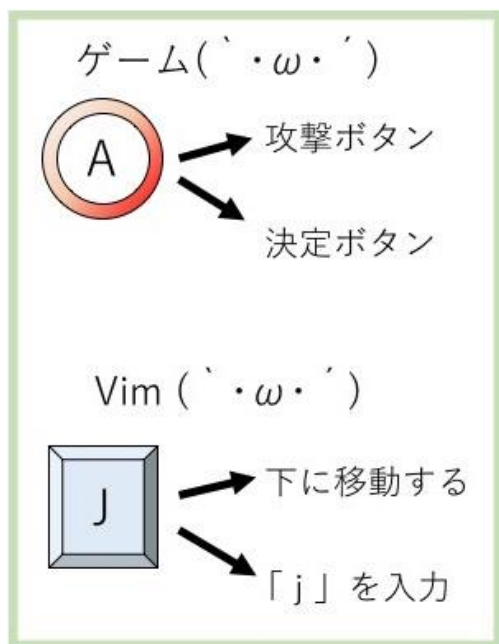
くことができます。

★★☆ Topic5 ビムライフ!

Vim の特徴として、「モード」の存在、Vim script による拡張、非常に多く利用者がいる点などが挙げられるでしょう。今回はそれらについて紹介していきたいと思います。

□ Vim は文字が打てない (´・ω・`)?

Vim はよく「文字を打つことができない」と言われてしまいます。これは vi や Vim 特有の「モード」が原因です。Vim はこの「モード」のせいで難しいとか、変態的なキーバインドとか言われてしまいますが、一度わかってしまえば理にかなっていることが実感できると思います。この記事では Vim をゲームとコントローラに例えて「モード」について解説してみます(´・ω・`)



↑ 場面によって役割が違う(´・ω・`)

□ モードごとに役割が変わるキー(´・ω・`)

ゲームコントローラのボタンは場面によって役割が変わります。例えば ○ボタン や A ボタン は戦闘中では「攻撃」ボタンとして、メニュー画面では「決定」ボタンとして使い分けているはずです。他にも Start ボタン のように、戦闘中の状態から Pause 画面へと場面を切り替えるために使われているボタンもありますね(´・ω・`)

Vim でも同じことをしています。ゲーム上の状態である「戦闘中」や「ポーズ」というのが Vim の「モード」に対応し、コントローラの各種ボタンがキーボード上の各キーに相当します。例えばキーボードの h、l、k、j キー は「Normal モード」と呼ばれる状態では上下左右に動くカーソルキーとして働きますが、文字を入力する「Insert モード」では普通のテキストエディタと同様に hjkl と文字を入力するキーとして振舞います。i キーは「モード」を切り替えるためのキーのひとつで、「Normal モード」から「Insert モード」へと状態を変えることができます(´・ω・`)

こうして「モード」を切り替えることで限られたボタン数で多種多彩な操作ができるようになるのです(´・ω・`)!

□ モードの利点(´・ω・`)

「モード」あることでどんないいことがあるのでしょうか(´・ω・`)?

それは、手の移動が最小限で済むことです(´・ω・`) キーボードのホームポジションから手を離すことなく、文字キーと最低限の修飾キーだけで、ありとあらゆる操作をすることができます。普通のテキストエディタではマウスやカーソルキーを使わなければならない場面でも Vim では文字キーを操作することで済んでしまうのです。手を大きく動かすことは案外面倒で、Vim の操作に慣れてからは カーソルキーに手を伸ばすことでさえ非常に億劫になってしまいました(´・ω・`)

□ Vim の設定(´・ω・`)

ここまで紹介してきた機能は、実は Vim 独自のものではなく Vim のご先祖様にあたるテキストエディタから受け継いできたものです。Vim の本領発揮はここからです(´・ω・`)

Vim は「コマンドラインモード」で使われるコマンドを組み合わせることで機能拡張を行うことができます。Vim のコマンドは非常に多機能で、単なる文字列操作にとどまらずウインドウの操作などの Vim の根幹に関わる制御まで行うことができます。しかも通常のプログラミング言語のように if 文などの制御構文や連想配列などのデー

```
[tweetvim] - (~) - VIM

@tsukubagominohi
ごみ情報をつぶやく非公式ゆるふわbotです(´・ω・`) このbotを;
フォローすると冷蔵庫にごみカレンダーを貼らなくてよくなります
(´・ω・`) リンク先を読んでお使いください(´・ω・`) 間違い訂
正は@high_moctaneへ(´・ω・`)
https://github.com/high-moctane/tsukuba-gomi-bot/blob/maste
/README.md
statuses : 1395
friends : 190
followers : 83

つくば市ごみの日bot (非公式) @tsukubagominohi
@high_moctane
今日のラッキーアイテムは
かん (レア度 ★★☆☆)
です(´・ω・`) 862 from つくば市ごみの日 Bot (非公式)
14:46

-----
つくば市ごみの日bot (非公式) @tsukubagominohi
@high_moctane
次のかんの回収日は、今日を含めないと
北地区: 1月29日(金) (4日後)
西地区: 2月4日(木) (10日後)
東地区: 2月2日(火) (8日後)
南地区: 2月5日(金) (11日後)
です(´・ω・`) 054 from つくば市ごみの日 Bot (非公式)
14:46

-----
つくば市ごみの日bot (非公式) @tsukubagominohi
23,1 0%
```

↑ プログラムも書ける Twitter クライアント「Vim」

タ構造を使うことができます。実は Vim のコマンドはプログラミング言語になっているのです(´・ω・`)! この言語は VimScript と呼ばれ、VimScript により様々な拡張をすることができるようになるのです(´・ω・`)!!

□ Vim の拡張機能(´・ω・`)

世界中の Vim ユーザーが VimScript を用いてプラグインを作成しています。そしてそれらはインターネット上に公開されているので、自分の環境に導入して Vim にいろいろな機能を追加することができます。

ここでは面白そうなものや個人的に便利だったものをいくつかピックアップして紹介してみたいと思います(´・ω・`)

★★☆ Topic5 ビムライフ!

The image shows a Vim editor window with a C program and its assembly output. The C code is as follows:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int i, sum = 0, product;
6     for (i = 1; i <= 10; i++) {
7         sum += i;
8         product *= i;
9     }
10    printf("sum: %d, product: %d\n", sum, product);
11    return 0;
12 }
```

The assembly output is shown below the C code:

```
frame #0: 0x00000000100000f2d a.out'main + 61
* rip = 0x00000000100000f2d
* rflags = 0x0000000000000a83

disassembly
frame #0: 0x00000000100000f2d a.out'main + 61
* rip = 0x00000000100000f2d
* rflags = 0x0000000000000a83

registers
0% < L1:1 N tmp.c c 46% < L7:5 Error: L5(2)
1 tmp.c|9 col 9| warning: variable 'product' is uninitialized when used here [-Wuninitialized]
2 || product *= i;
3 || ~~~~~
4 tmp.c|5 col 28| note: initialize the variable 'product' to silence this warning
5 || int i, sum = 0, product;
6 || ^
7 || = 0
8 || 1 warning generated.
[No Name] > - unix < utf-8 < qf 12% < L1:1
```

↑ デバッグも楽々(´・ω・`) (vim-lldb と watchdogs)

▼ TweetVim

ツイッタークライアントです。Vim から
直接つぶやけるので非常に便利です(´・
ω・´)

▼ vim-lldb

デバッガである LLDB を Vim から使うことができます。プロ序の課題もこれで一瞬ですね(´・ω´)

- ▼ watchdogs

非同期のシンタックスチェッカーです。

プログラムをコンパイルする前に間違いを
 教えてくれるので導入しておくべきでしょ
 う(`・ω・´)

▼ previm

Markdown などのプレビューをブラウザで見ることができます。テキストを変更するとすぐに結果が表示されるので重宝しています(`・ω・`)

▼ vimtex

Vim で LaTeX を書くことができます。
Latexmk というプログラムを呼び出して

LaTeX の自動コンパイル、自動プレビューができます。レポート書くときにお世話になっています(´・ω・｀)

□Vim ユーザーは非常に多い(´・ω・｀)

Vim は歴史が長く利用者が非常に多いので、Vim に関する情報や便利なプラグインなどを簡単に見つけることができます。日本の利用者も多いので日本語で情報が得られるのもかなりメリットだと思います。困ったことがあってもググれば問題が解決できるので、初めて Vim に触るときも案外困らないはずです(´・ω・｀)

□Vim ライフ (´・ω・｀)

Vim に興味が沸いたら、まずは自分にあった Vim をダウンロードしましょう。Vim を入手する方法はいろいろありますが、ここでお勧めするのは "Kaoriya" 版の Vim です。普通の Vim と違う点は、初心者向けにチューニングされている点、日本語に特化している点などが挙げられます。Vim を初めて使うにはぴったりだと思うのでオススメです(´・ω・｀) 基本的には Windows 版ですが、Mac にも移植されているので探してみてください。

ダウンロードが済んだら、vimtutor という付属のチュートリアルを実行してみましょう(´・ω・｀) 30 分程度で Vim の基本的な機能をおさらいすることができます。このチュートリアルも結構軽妙で、Vim の面白さがダイレクトに伝わってくるはずです

(´・ω・｀)

□あとかぎ

はじめて Vim を使ってみようと思ったのは、Vim のチュートリアルをしているときでした。書かれた文章から楽しさが伝わってきて、しばらく使ってみようと思ううちに気づくと Vim の操作でないとダメな体になってしまいました。最初は特殊な操作に驚きましたが、案外慣れてしまうものですね(´・ω・｀)

この記事を読んで Vim に興味がある人が増えたらいいなと思います(´・ω・｀)

```

1  #include <stdio.h>
2
3  int main(int argc, char const *argv[])
4  {
5      printf("Hello World.");
6      return 0;
7  }

```

Simple Programming

□はじめに

はじめまして、Adelie です。布団が好きです。

最近どこかで聞いた台詞で「**そのジャンルに興味はないけど、そのジャンルが好き**な人が語る話は好き」というのがあって、ストーンと心に落ちる思いでした。これ僕だけじゃなくて、結構多くの人に共通の見解だと思っているのですが、どうでしょうね？

という 言い訳 前置きを踏まえて、興味がない人が大半であろう「**プログラミング**」が今回の話題です！

以下諸注意をば。

- 「大学の講義でC言語をやったことがある」くらいの知識を前提に書いています(ごめんなさい)
- 記事に掲載したコードは、少なくとも学校の PC の環境では動作を確認していますが、環境によっては動かない可能性が**十分考えられる**ことを先に断っておきます
- 僕の持てる知識の範囲で正しいことを書いているつもりですが、間違い

等見つけましたら是非雑誌部まで

- 「工学的」で「役に立つ」内容ではないことを保証します

□工学的な発想

僕の所属する某学類にはC言語の講義がありまして、あるときこんな課題が出たわけです。

コマンドライン引数に文字列がいくつか渡される。ただし文字列の個数は最大で30であるとする。

出現回数が同じ文字列を 1 行にまとめて、以下の例にならって出力せよ。

```

% ./a.out ab cd ab cd ef ef ab gh
3 ab
2 cd ef
1 gh

```

ちょっと一部を都合よく改変したり端折ったりしていますが、だいたいこんなような課題でした。問題設定が若干曖昧な感じがしますが、そのあたりは心の目で見ていただければと……

★★☆ Topic6 Simple Programming

で、例えば先の課題はこんなコードを書くことができます。このコードの中身自体は今回の話にあまり関係がないので、適当に読み飛ばしてもらって結構ですよ。

```
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[])
{
    int count[30] = { 0 };
    int dontprint[30] = { 0 };

    for (int i = 1; i < argc; ++i)
    {
        for (int j = i; j < argc; ++j)
        {
            if (strcmp(argv[i], argv[j]) == 0)
            {
                count[i]++;
                if (i != j) dontprint[j] = 1;
            }
        }
    }

    for (int i = 1; i < argc; ++i)
    {
        int flag = 1;

        for (int j = 1; j < argc; ++j)
        {
            if (count[j] == i && dontprint[j] == 0) {
                if (flag == 1)
                {
                    printf("%d", i);
                    flag = 0;
                }
                printf(" %s", argv[j]);
            }
        }

        if (flag == 0) printf("¥n");
    }

    return 0;
}
```

ところで、この講義を担当する某先生はどうやら板書派であるらしく、課題の解答例なんかも全部その場で板書します。

でも、上の解答例を見ていただければ分

かると思うのですが、これ板書するの結構大変だと思うんですよ。だって長いじゃないですか。

さて、こういう小さな「不便」を見つけたらより「便利」にしたいと考えるのが工学の心です。

学類生とはいえ工学を志すものの端くれとしては、放っておくわけにはいきません。

板書には板書のいいところがありますから、あくまで黒板とチョークを用いるという前提で、いかに先生の負担を軽減できるか……。そうして考えぬいた結果、僕は思いついてしまったのです。

そうだ！解答が短ければいいのでは!?

まごうことなき天才の発想と言えるでしょう。

そうと決まれば早速**ありとあらゆる手段**を講じて、解答を短くしてやりましょう。

□include が長い

まず目につくのが冒頭の include ディレクティブ。よく「おまじない」と称されるアレです。

```
#include <stdio.h>
#include <string.h>
```

これだけで 37 文字(改行文字を含めるなら 39 文字)もあるんですよ。削りたくないですか？

そこで「本当にただの『おまじない』だ

★★☆ Topic6 Simple Programming

ったら別に書く必要ない¹のでは～～～？」とか言いながら include を全部削除して学校の PC の gcc を叩いたら**本当に通ってしまいました**²。やったね。

たちまちのうちに 39 文字が消え失せました。幸先がいいですね。

□変数宣言が長い

例えば整数の変数が欲しかったら普通は `int x` って書きますし、初期化の必要があれば `int x = 0` とか書きますね。

ところが、変数(と関数)はグローバルで宣言したときに限って、**型の指定も初期化も省くことができます**。型の指定を省いた場合 `int` と見なされ、初期化を省いた場合は 0 で初期化されます。ですから次のコードは合法です。短くてすっきりです。

```
x;           //int x = 0; と同じ
y[10];       //int y[10] = {0}; と同じ

main()       //int main() と同じ
{
  //中略
  //ちなみに「return 0」も省略できる
}
```

□if 文が長い

突然ですが、高校数学の内容をちょっと思い出してみてください。

例えば「A または B」という命題 P があったとして、これの真偽が知りたいとします。

もし A が真なら、B の真偽を確認するま

でもなく P が真であると分かるので、B を読む必要がありません。

C 言語は横着者なので、先の例と同じように、読まなくて良い条件は読み飛ばそうとします。これを短絡評価といいます。

短絡評価は論理演算子の `&&` か `||` が用いられた式において、演算子の左側だけ見て結果がわかってしまうときに発生します。

短絡評価の動作する例を参考までに載せておきます。以下の例で画面に出力されるのは 2 ではなくて 1 です。

```
int x = 1;

int a = 5;
int b = 10;

if (a < b || x++) { //x++は読まれないので...
    printf("%d\n", x); //1が出力される
}

if (a > b && x++) { //このx++も読まれないので...
    printf("ここは実行されないよ");
}
else {
    printf("%d\n", x); //やっぱり1が出力される
}
```

ところでこの機能、逆手に取れば if 文の代わりになります。

「A が偽なら『A && B』の B は読まれない」ということは、取りも直さず「**A が真でなければ B が実行されない**」ということです。それって if じゃん！

「条件 X が真なら do_something() を実行する」という内容を書きたければ、if 文なら

```
if(X) do_something();
```

¹ そんなわけではない

² もちろん環境依存ですので注意してください

と書きますが、短絡評価を利用してこれを書けば

```
X && do_something();
```

で済んでしまいます！

少なくとも `if()` の 4 文字が必要なところを、`&&` の 2 文字で済むので 2 文字分短く書けますね。2 文字も書く時間があれば宅通の僕でも家まで帰れちゃいますよ³。

「if 文はいらない⁴」ということが分かっていただけたかと思います。

□ while 文が長い

「最初に I をして、X が真である限り Y を繰り返す」という処理は while を使っても for を使っても書くことができますが、次を見れば分かるように for を使ったほうが 1 文字短くなります。

```
I;while(X){Y;}
for(I;X;Y){;}
```

while 文もいらないということが分かりましたね！

□ ブロックが長い

ブロックとは波括弧で囲われた部分のことです。

if や for などの制御構文は、条件が真のと

きに直後の文またはブロックを実行します。(正確には、ブロックで囲われた部分がひとつの文と見なされている、の方がより実態に近いのですが、まあどういう認識でも実用上問題はなからうと思います)

でも考えてみてください。ブロックを使うと、処理とは関係のない文字が 2 文字(波括弧の開きと閉じ)も増えてしまうんです。そんなにたくさん板書してたら講義終わっちゃいますよ。

つまりブロックさえ使わなきゃいいってことなんですけど、じゃあ複数の文じゃなきゃ処理できないような中身を書きたいときゃどうするんだよという話になります。

ご安心ください、コンマ演算子という便利なものがあるんです。

コンマ演算子は式を連結させることのできる演算子です。

ざっくりと説明するなら、式とは要するに算数の計算式と同じで「計算すれば一つの値になる意味のかたまり」です。

例えば「 $2+3$ 」は計算すれば「5」になる式で、「1」は計算すれば「1」になる式です。また、「 $1<2$ 」は計算すれば「0 以外」になる式です。なぜなら論理演算は真であれば「0 以外」、偽なら「0」を導く「計算」だからです。あるいは、関数も戻り値という計算結果がありますから、`printf("ぽわ")` も式です⁵。

この辺になってくるともう訳がわからない感がありますが、戻り値がない関数も void という「何もない値」を導く式であると

³ 2 時間かかります

⁴ そんなわけではない

⁵ printf は出力した文字数を返します

★★☆ Topic6 Simple Programming

いうことができます。

これらは全てコンマ演算子で繋ぐことが可能です。コンマ演算子も演算子ですから、きちんと"計算"をして、ひとつの値を導きます。計算「A,B」の演算結果は「B」です。つまり、演算子の左側の計算結果はすべて無視されることになります。

```
//「ぼわ」が出力されたのち、x には-1 が代入されます
int x;
x = (printf("ぼわ"), 1+2+3, -1);
```

えーっと、何の話をしていたんでしたっけ……ああそうそう、ブロックを使わずに複数の文を if とか for に入れる話でしたね。

コンマ演算子を用いれば、おおかたの文は一つにまとめてしまうことができます。単一の文ならブロックが不要ですから、例えば次のような書き換えが可能です。

```
//書き換え前
for (int i = 1; i<3; i++) {
    printf("ぼわ");
    printf("ぼげ");
}

//書き換え後
for (int i = 1; i<3; i++)
    printf("ぼわ"), printf("ぼげ");
```

めでたく 2 文字削れました。2 文字削るための説明に一体何文字かけたことでしょう。

□ その他の長いもの

⁶ とかいつつコードを短くするのに一番貢献してるのは多分こいつら。特に後ろ 2 つ

いろいろ述べてきましたが、短くする方法は他にも星の数ほどありますから、ここで全てを網羅することは難しいです。今回僕が用いた手法には他に「char* v[]」を「char**v」に書き換える」とか「改行を全部省く」とか「変数名を全部 1 文字にする」なんてものがありますけど、この辺は説明するようなものでもない⁶ので触れるのみにとどめます。

もうひとつ特筆すべきものがあるとするなら「1 つの変数に 2 つ以上の意味を持たせる」とかになりますけど、これは説明しだすとキリがないのでやはり触れるだけにします。

□ できあがったもの

今まで説明してきた様々な手段を用いて、冒頭の課題を解き直したものがこちらになります。

```
i;j;n[99];main(int c,char**v){for(i=c;--i;)for(j=i;!strcmp(v[i],v[j]))&&(n[i]++,i-j)&&(n[j]=-99),--j);for(;--c;)for(j=i=48;--i;)n[i]-c||(j&&putchar(c+j),j=!printf(" %s",v[i]),j-i+1||puts(""))};}
```

わあ、気持ち悪い。

それでも一応このプログラムはちゃんと課題の要求をクリアしています。詳しい挙動の解説は省略しますので、興味のある方は各自手元の環境で動かすなどして⁷考えてみてください。コード中に出てくる「48」

⁷ 僕は VisualStudio2015 で実行しましたが、警告を黙らせる設定をしないと上手く動作しないかもしれません

はなんで 48 じゃなければならないのか、とか。

上のコードは 193 文字ですが、この程度なら先生も楽々板書できますし、僕たちも楽にノートが取れますし、いいことづくめです。ばんざい！

□ 結論

プログラムは短ければいいというものではないということが身にしみて分かりました。二度とやらない。

□ 最後に

コードをできるだけ短く書く取り組みには「コードゴルフ」という名前がついていて、一種のパズルのように親しまれています。検索するとその手の記事がたくさん出てくると思うので、これを読んで興味を持った方は少しコードゴルフの世界を覗いてみてはいかがでしょうか。


かくいう僕は、コードゴルフに関してはこれが初挑戦で、前述のコードについても「これで最短！」という自信は実はなかったりします。「もっと短いのを書いてやったぞ！」という方がいらっしやいましたらどうぞ雑誌部まで！

□ 追記

とか言ってたら早速はいもくたん(5 章書いた人)にもっと短いのが書かれました。173 文字です。

```
i;j;n;m;f;g;main(int c,char**v){for(;--n;g=0)for(i=c;--i;m--n||printf(g?" %s":"%c%d %s",g++?v[i]:f++?10:0,n,v[i]),m=0)for(j=c;--j;strcmp(v[i],v[j])||(i>j?m=0:m++));puts("");}
```

実行速度がえらいことになりますが、そもそも黒板に書かれたコードが実行されることなんてないので、文量が短いほうが正義です。



Java でピアノを作る

こんにちは。寝そだです。工学システム学類にもついに学類誌(「が狂い死」に変換されがち)が出るという噂を聞いて、「いつ出るんだろう...いつ出るんだろう...」とそわそわしてたところ「記事を書いてくれ」と頼まれた。「なるほど! こう来たか!」と驚愕しながらも私は首を縦に振った。

さて、工学システム学類の学類誌ということで何を書こうかと考えた。まあ自分の趣味に近いことを正直に書いたほうが良いだろうと考えた。私はコンピューターで曲を作ること、俗にいう DTM を趣味としており、web 上にちょくちょく作曲したものをあげたりしている。ということで今回はそういうことを書いていこうと考えた。

□MIDI について

突然だがみなさんは MIDI というものをご存じだろうか。

MIDI 検定のホームページの「MIDI とは」(<http://www.midilicense.com/hajimete/midikentei2015.html>)の最初に

Musical Instrument Digital Interface の頭文字を組み合わせた言葉で、電子楽器

やコンピュータ等のメーカーや機種に関わらず音楽の演奏情報を効率良く伝達するための統一規格です。

と書いてある。なるほどわからん。その後にも色々書いてあるが要約すると、音の大きさ、高さ、長さや楽器の種類などをまとめた演奏情報のことだ。wav ファイルではなく、あくまで楽譜のようなものである。データ量が小さく、また録音されたデータとは違い一度作ったあとでもデータのミを変えたりとか、ピアノで鳴らしていた音をトランペットに変えるといった大胆な編集が可能である。昔のホームページに入るとだらしのない音源の BGM が流れる経験があると思うが、それが midi で作られた音楽である。

私が趣味としている DTM では、DAW という midi の編集ソフトのようなもので演奏データを作り、良い感じの音源を用いて wav ファイルに変換することで曲を作ることが出来る。

まあそういうわけなので(????)、今回の記事ではこの MIDI を使ってピアノを作ってみたいと思う。

□用意するもの

- Java のコンパイル出来る環境
- 適当なエディタ(vim でも Emacs でもメモ帳でも良い。)
- ターミナル(コマンドプロンプト)

□Java で MIDI を扱う

DTM の始まりはパソコンにソフトをインストールしてから音を鳴らすところである。つまりは MIDI をプログラミングで取り扱うのも音を鳴らすところからである。Java の HelloWorld は初見では正直意味わからないので Java が未経験な方でも是非やってみて欲しい。なお、私自身もつい最近 java を勉強し始めた程度なので拙い点があるかもしれないがどうかご了承頂きたい。

ではさっそく音を鳴らすコードを書いていこう。私は最初に色んなものを参考に以下のようなコードを書いた。

このプログラムについて軽く解説する。まず `import javax.sound.midi.*;` を一番上に書くと Java の MIDI のライブラリを読み込む MIDI を扱うことが出来る様になる。Re

`ceiver receiver= MidiSystem.getReceiver();` で MIDI 信号のレシーバーを宣言している。Receiver 型のメソッドは MIDI 信号を受け取るための `send()` メソッドを定義しており、これを使うことで MIDI 信号から音を鳴らすことが出来る。

`ShortMessage message = new ShortMessage();` で、Receiver が受け取るための信号を作るための変数を宣言している。

`message.setMessage(ShortMessage.NOTE_ON, 60, 127);` では、先ほど宣言した信号の変数に、NOTE_ON という音を鳴らしますという信号と、音の高さ(60=ド、61=ド#、62=レ...)と、音の大きさ(0 から 127 で設定出来る)を入れている。今回は、「ド(60)の音を 127 の音量で鳴らします。」という信号を変数 message に入れた。

最後に `receiver.send(message, -1);` で、message に設定された音を鳴らしている。

さて、このプログラムを `sample.java` という名前で保存して実行したいと思う。ターミナルで

```
javac sample.java
java sample
```

```
import java.io.*;
import javax.sound.midi.*;

class sample{
    public static void main(String[] args) throws
        Exception {
        Receiver receiver = MidiSystem.getReceiver();
        ShortMessage message = new ShortMessage();

        message.setMessage(ShortMessage.NOTE_ON, 60, 127);
        receiver.send(message, -1);
    }
}
```

★★★ Topic1 Java でピアノを作る

というコマンドでプログラムを起動した。
……音は鳴らなかった。

なぜこうなるのかよくわからなくて結構悩んだが、結論としてはこのプログラムだと midi 信号を Receiver に送った直後に終了してしまうからである。これを解決するには、とりあえず `Thread.sleep(1000);` というコード音を鳴らす信号のあとに挿入すると、ドが 1000 ミリ秒(1 秒)だけ鳴ってプログラムが終了することがわかった。

□ ピアノを作ろう

先ほど、なんと我々は苦勞に苦勞を積み重ね、そしてついに MIDI を使ってドの音を 1 秒の間鳴らすことが出来た！なんと素晴らしいことであろうか。

と、心の中では思っていたがよくよく考えると先ほどのプログラムはドの音が鳴るだけである。これだけではクソつまらないプログラムだと我ながら思う。これでは編集長に「お前締め切りギリギリまで粘ってこんなゴミのような記事書いてきて、タダで済むと思うなよ。」と言われてしまうので、とりあえず好きな音を鳴らせるようにしたいと思う。

音の鳴らし方がわかればあとどうにでもなると思うので、パソコンのキーボードをピアノに見立てて、さっそくピアノプログラムを作っていこう。以下のような方法で実装したいと思う。

- ①キーボードから文字を入力する
- ②文字に対応した音階を代入

- ③信号を send し音を鳴らす
- ④最初に戻る。

きっとこんな感じで良いはずである。今のところ完成している部分は③なのでそこ以外を書いていく。ぶっちゃけ①②④は midi とか全く関係ないのでググったりググったりググったりすれば出来るだろうという甘い考えを抱きながらプログラムを書いていくこと n*10 分ほど経ち...出来た。

以下のような 35 行のプログラムになった(編集注: ソースコードは次ページに掲載しました)。

□ このプログラムについて

このプログラムは最初にしたプログラムを基に①②④の機能を入れたものである。

プログラムの遊び方であるが、キーボードの a=ド, s=レ, d=ミ, f=ファ, g=ソ, h=ラ, j=シ, k=ドに対応していて、q を押すとプログラムを終了するというものである。ついでに上記以外のキーを入れたら注意する文を吐いてくれるようにした。

プログラムを実行したら a を押して enter キーを押すとドの音が鳴り、s を押して enter キーを押すとレの音が鳴る。どう考えてもこれはピアノである。

□ 最後に

最終的に作ったピアノプログラムは enter キーを押さないと音が鳴らない、和音には対応しない、ピアノの音しか鳴らせないな

★★★ Topic1 Java でピアノを作る

どガバガバな点は多々あるが、ピアノとして必要な最低限の機能を実装できたのでは、と我ながら思っている。そもそも、これ以上機能を追加すると私が色々な意味で死んでしまうのでどうか許して欲しい。(執筆当時試験期間中の顔をしながら)

今回はピアノで、しかも単音限定にしたが、MIDI のプログラムでは和音を鳴らしたり、for 文を使って音階を上げ(下げ)ながら音を鳴らす、乱数で音を鳴らす、音源をピ

アノ以外の音に変える、GUI の実装でユーザーが使いやすいようにするなどまだまだ無限に可能性を秘めている。

この記事を読んで万が一、Java で MIDI の音を鳴らすことに興味を持ってくれた人がいたら是非、MIDI を使って様々なプログラムを作って遊んでみて欲しい。プログラミングなんてやってられるか！俺は音楽がやりてえ！って人は是非 DTM に手を出して欲しい。

```
import java.io.*;
import javax.sound.midi.*;

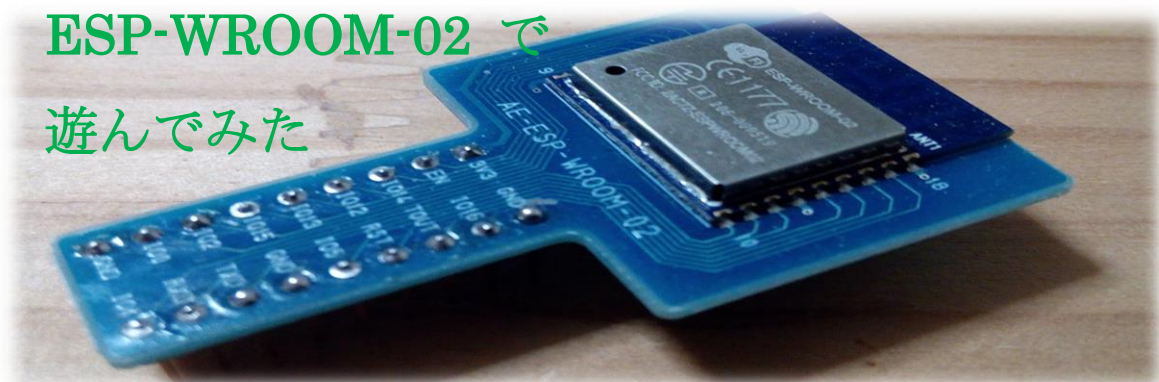
class MidiPiano{
    public static void main(String[] args) throws Exception {
        System.out.println("a=ド, s=レ, d=ミ, f=ファ, g=ソ, h=ラ, j=シ, k=ド");
        System.out.println("quit=q");
        int midiSignal = 0;

        while(midiSignal!='q'){
            InputStreamReader reader = new InputStreamReader(System.in);
            midiSignal = reader.read();

            if(midiSignal=='a'){midiSignal = 60;}
            else if(midiSignal=='s'){midiSignal = 62;}
            else if(midiSignal=='d'){midiSignal = 64;}
            else if(midiSignal=='f'){midiSignal = 65;}
            else if(midiSignal=='g'){midiSignal = 67;}
            else if(midiSignal=='h'){midiSignal = 69;}
            else if(midiSignal=='j'){midiSignal = 71;}
            else if(midiSignal=='k'){midiSignal = 72;}
            else if(midiSignal=='q'){break;}
            else {System.out.println("Input a,s,d,f,g,h,j,k,or q");
                midiSignal = 0;}
            sound(midiSignal);
        }
    }

    public static void sound(int midi) throws Exception{
        Reciever reciever = MidiSystem.getReciever();
        ShortMessage message = new ShortMessage();
        message.setMessage(ShortMessage.NOTE_ON, midi, 127);
        reciever.send(message, -1);
    }
}
```

ESP-WROOM-02 で 遊んでみた



9月の末、某有名電子部品販売店「秋月電子通商」で面白いマイコンが発売した。ESP-WROOM-02 というマイコンだ。

電源電圧、入出力電圧ともに 3.3V で、IO として一般的なデジタル入出力は勿論、シリアルや SPI、I2C に、知っている人は知っているデジタルオーディオ用の I2S やアナログ入力も持っている。ここまでなら最近の便利なマイコンには付いていたりするが、こいつはさらに Wi-Fi モジュールも搭載している。

秋月電子通商では変換基板付きで¥650 とお手頃価格で発売しているので、ここまで読んで興味を持ってくれた方は是非購入して遊んでみてほしい。

しかしながらこれだけでは良く分からないという人が大部分だと思われるのでもう少し詳しく。このマイコン、中身のチップが ESP8266 という海外ではそこそこ遊ばれているもので、Wi-Fi に繋げて周辺を I/O でごにょごにょできるので簡単に IoT(Internet of Things)できる便利なものである。

この度、Wi-Fi の技適が通ったため国内で

も発売するようになったという経緯があるのだが、発売から約 2 ヶ月、ようやく日本でも話題になり始め、電子工作を嗜んでいる人々が最近注目している。

10 月頭にこれを購入し、日本語での情報が少なく、また周りに弄っている人も居なかったために英語サイトとわずかな日本語サイトを頼りに奮闘していた私としては大変嬉しい。

この記事は約 2 ヶ月一人でこのマイコンと戦っていた私のこのマイコンのレビューのようなものだと思っていただきたい。

□ 何故 ESP-WROOM-02 か

さて、早速だが、先ほどこのマイコンが注目されていると記したが、注目されている理由としては 2 つある。

まず 1 つ目はその価格だ。しかしながら、あまり電子工作に詳しくない人には¥650 という価格だけではピンとこないかもしれない。そこで、比較的知名度が高いもので「Wi-Fi に繋げて IoT で周辺機器を操れるプログラマブルなデバイス」を作る方法と

そのコストを考えてみる。

まずはラップトップ PC を用いてみよう。目的に対してあまりにも過剰なスペックになるが気にしないで考えると、現在私が使っている ThinkPad X220 が中古で¥25k 程度だった。確かにプログラマブルで Wi-Fi に繋げる上、入出力インターフェイスも豊富だが、何かを作る度に¥25k の出費はあまりにもモッタйнаイし、サイズも無駄に大きい。

では少しスペックを落として、Raspberry Pi に Wi-Fi ドングルを付けるという方法を考える。Raspberry Pi で約¥4k、Wi-Fi ドングルで約¥1k 程だろうか(執筆段階では\$5 の Raspberry Pi zero というものがあるが、あまりリサーチしていないので考慮していない)。サイズもラップトップ PC の半分以下になり、価格も¥5k なので、作るものの仕様やそれにかける情熱次第では良さそうである。

一応さらにスペックを落として、Arduino と Wi-Fi シールドの組み合わせを考える。

実はこの組み合わせ、コストが約 10k になってしまい、Raspberry Pi を用いた場合より高くなってしまう。では Raspberry Pi だけでいいのではと思われるかもしれないが、利点もある。

Raspberry Pi は一般的には Linux が走っており、その上でプログラムを動かすため、プログラミングの知識に加え多少は Linux の知識も必要となる。しかしながら Arduino は C++ 風 Arduino 言語でプログラムを書き USB 経由(正しくは USB シリアル経由)で直接書き込むことで IO を弄るので、

Arduino 言語だけを学べば良く学習コストが低いのである。これは好みによるだろう。しかしながら¥650 が格安というのが分かっていたただけだろう。

他にも法律に触れる恐れがあるがファームウェアを書き換えた無線 LAN ルーターを用いる方法や、Raspberry Pi より安い SBC(Single Board Computer)を用いるなどが考えられるがここまでは安くないと思われる。

もう一つの理由としては、Arduino 言語でプログラムを書き、Arduino IDE から書き込みができるという点だ。

Arduino IDE は書き込むボードを選択することができるが、ESP8266 用のボード情報があるのでそれを IDE にインストールすれば簡単に Arduino 言語で ESP-WROOM-02 で遊ぶことができる。

さらにありがたいことに、Wi-Fi の SSID のスキャンやちょっとした Web サーバーのサンプルプログラムもあり、Wi-Fi への接続などで苦労することはまずない。

勿論、Serial や wire といった Arduino で使うクラスも使えるので、デバックや外部回路との接続もまず困らないだろう。

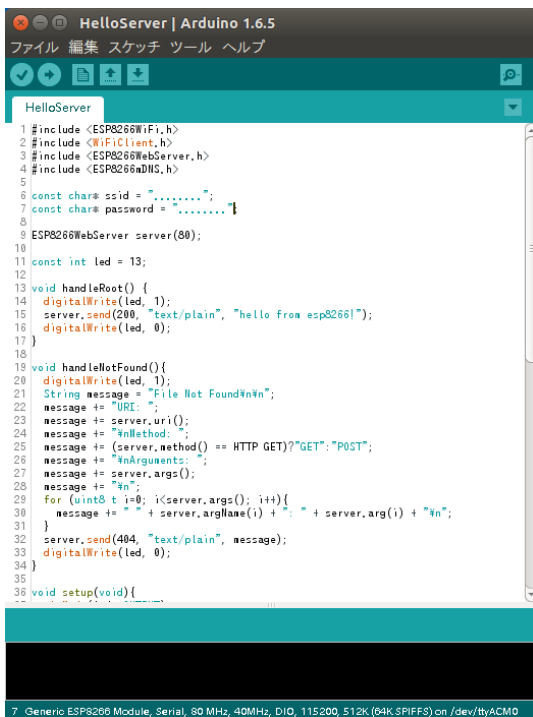
こういった理由で注目されているのだ。Wi-Fi に繋げる Arduino が¥650 なのだから注目されるのは必然だろう。

□ サンプルプログラムでちょろっと

★★★ Topic8 ESP-WROOM-02 で遊んでみた

Arduino IDE で ESP8266 のプログラミングをできるようにする方法は Qiita などにあるのでこの場では割愛する。

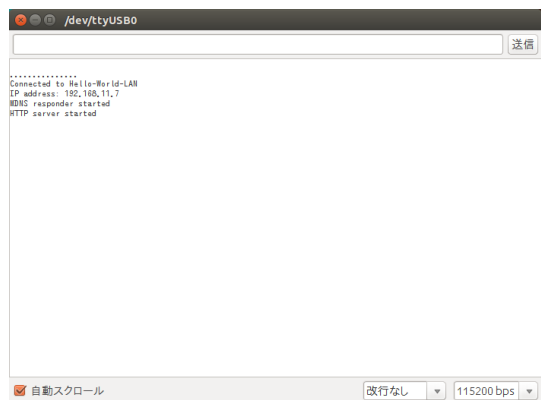
ちょっとしたWebサーバー「HelloServer」のサンプルプログラムがこちらになる(編集注: 次ページに文字起こしを掲載しました)。(サンプルなので全体が気になる方は各自で確認して欲しい)



```
1 #include <ESP8266WiFi.h>
2 #include <WiFiClient.h>
3 #include <ESP8266WebServer.h>
4 #include <ESP8266DNS.h>
5
6 const char ssid = ".....";
7 const char password = ".....";
8
9 ESP8266WebServer server(80);
10
11 const int led = 13;
12
13 void handleRoot() {
14   digitalWrite(led, 1);
15   server.send(200, "text/plain", "hello from esp8266!");
16   digitalWrite(led, 0);
17 }
18
19 void handleNotFound() {
20   digitalWrite(led, 1);
21   String message = "File Not Found\n\n";
22   message += "URI: ";
23   message += server.uri();
24   message += "\nMethod: ";
25   message += (server.method() == HTTP_GET) ? "GET" : "POST";
26   message += "\nArguments: ";
27   message += server.args();
28   message += "\n";
29   for (uint8_t i=0; i<server.args(); i++){
30     message += "  " + server.argName(i) + ": " + server.arg(i) + "\n";
31   }
32   server.send(404, "text/plain", message);
33   digitalWrite(led, 0);
34 }
35
36 void setup(void) {
37   // ...
38 }
```

ご覧の通り、グローバル領域で定数変数として Wi-Fi の SSID とパスワードを宣言するととりあえず動くようになっている。

実際に動かしてみると、シリアルモニタに色々と出力されてサーバーがスタートしたことが分かる。

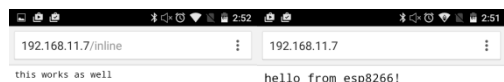


```
.....
Connected to Hello-World-LAN
IP address: 192.168.11.7
MDNS responder started
HTTP server started
```

```
.....
Connected to Hello-World-LAN
IP address: 192.168.11.7
MDNS responder started
HTTP server started
```

起動が確認できたので、ブラウザからアクセスしてみる。

アクセスするにはシリアルモニタに表示されている IP アドレスを直接用いる。



```
192.168.11.7/inline 192.168.11.7
this works as well hello from esp8266!
```



全てを把握したわけではないが、基本的には String に HTML 文を格納し、Server.send()で String をブラウザからのリクエストに対して返している。

★★★ Topic8 ESP-WROOM-02 で遊んでみた

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = " ... ";
const char* password = " ... ";

ESP8266WebServer server(80);

const int led = 13;

void handleRoot() {
    digitalWrite(led, 1);
    server.send(200, "text/plain", "hello from esp8266!");
    digitalWrite(led, 0);
}

void handleNotFound() {
    digitalWrite(led, 1);
    String message = "File Not Found\r\n\r\n";
    message += "URI: ";
    message += server.uri();
    message += "\r\nMethod: ";
    message += (server.method() == HTTP_GET) ? "GET" : "POST";
    message += "\r\nArguments: ";
    message += server.args();
    message += "\r\n";
    for (uint8 t i = 0; i < server.args(); i++) {
        message += " " + server.argName(i) + ": " + server.arg(i) + "\r\n";
    }
    server.send(404, "text/plain", message);
    digitalWrite(led, 0);
}

void setup(void) {
    //以下略
```

逆に言えば、HTML（と JavaScript）でできることはだいたいできるというわけだ。

メモリには余裕があるので多少規模の大きい HTML も投げ返すことができる。

センサの値を取得して簡単にブラウザから確認したり、逆にブラウザから簡単に IO をいじったりができるわけだ。

□ 良い点だけではない

今回は ESP8266 のちょっとした紹介程度だったが、興味を持っていただければ幸いだ。

そのため良い点をできるだけ伝えようとしてきたが、もちろん欠点もある。

まず IO が少ない。

Arduino UNO がデジタル・アナログ合わ

★★★ Topic8 ESP-WROOM-02 で遊んでみた

せて計 20 本あるのに対して、ESP-WROOM-02 は 12 本しかない。I2C と SPI で競合するピンも存在する。

数多くのスイッチングをしたい場合などは歯がゆい思いをするだろう。

また消費電流が大きいことが挙げられる。

Wi-Fi にアクセスするため仕方がないとは言え、通常時 200mA 以上は平均して消費するらしい。

IoT デバイスとしては大きなバッテリーを積むなどは避けたい場合があるので目的によってはESP-WROOM-02は適さない可能性が有る。

そして我々学生としては辛いことに、WPA-enterprise (utwlan-x) に繋がらない。

少なくとも私が調べた結果、公式 Espressif が未対応の旨を言っている。

utwlan-w は接続できると思われるが、認証を通す部分でメモリを消費してしまい手間もかかってしまうのであまり好ましくない。

しかしながら公式が検討はするとも言っているので、もしかしたら対応するようになるかもしれない。

また発生条件があいまいながら、Stack Overflow が発生して WDT(WatchDog Timer)機能が働き再びブートし以下ブートループという現象が発生した。

これは自分で書いたプログラムを書き込み遊んでいたときなのだが、それまでは何事もなく動いていたのに突然正常動作しなくなったのだ。

何度同じプログラムを書き込んでも変わらないため壊れたのかと諦めかけたが、複

数のサンプルプログラムを何度か書き込むことで解消されることが分かった(私は根性焼きすると言っている。)

□最後に

長くなったが、要するに「ESP-WROOM-02 楽しいからみんな遊ぼうよ!!」ということである。

まだまだ情報が少ないため手が出しづらいかもしれないが、新しい物は新しい可能性に気づかせてくれる。

是非挑戦してみて欲しい。

Column 大手キャリアから格安 SIM に乗り換えて留年する(2/3)

○メリット

- 1, 利用料金が安い
- 2, 大手キャリアに存在する契約期間に縛られない

○デメリット

- 1, スマホを買い換えなければならないことがある。(必ずではない)
- 2, 通話代がかかる
- 3, キャリアメールが使用できなくなる
- 4, 時間によりデータ送受信の速度にバラつきが出る

コラムですので、要点だけかいつまんで説明しますと

メリットの 1, 利用料金が安い については

私は今月(2016 年 1 月)、au 社に約 11000 円の利用料金を支払いましたが、

所謂格安 SIM に乗り換えると、月々2000 円以下にまでその支出を抑えることが可能

になります。

ここで、格安 SIM を月々1000 円で契約した場合を考えると

月々10000 円 × 12 ヶ月 × 4 年 = 約 50 万円

と、大学在学中に約 50 万円もの出費を抑えることが可能になるわけです。

私の場合、月々約 10000 円強の支払いと、やや高めの契約をしていたのでこのような極端な数字が現れたということもありますが、それでも数十万円の出費を抑えることが可能になるのは確かです。

大学院に進学し、6 年間大学に在学することを考えると、確実に一回留年できる程度の出費を抑えることが可能になります。(タイトル回収)

(3 に続く)

```
1 var vid = new Whammy.Video(frames);
2 vid.add(canvas);
3 vid.compile();
```

Webブラウザで動画を作ろう！

工学システム学類 2 年次のグリです。最近、Ubuntu を使い始めたのですが、exe 形式のファイルが完全には実行できないため、Windows のツールを使うことが出来ず不便だな、と感じることがあります。そこで、この頃、OS に依存しないで動くソフトウェアで作業すればいいのではと考え始めたので、今回は Web ブラウザ上で動画を作ってみようと思います。

□ 動画を作るためのライブラリ

今回は Javascript を使ってクライアントの中で完結する動画作成を行ってみようと思います。

そこで、今回は whammy.js というライブラリを使います。

このライブラリは javascript で描画した canvas の内容を動画の 1 フレームとして複数枚つなぎ合わせて Webm 形式の動画ファイルを作るといえるものです。

よって、普通の動画編集ソフトよりも自由度の高い描画を行うことができるようになっています。

しかし音声をつける機能は無いので、音声をつけるなら、別途に音声ファイルを用

意して、他のソフトウェアで動画ファイルと音声ファイルを統合する必要があります。

□ ライブラリの使い方

このライブラリは以下のように使うことができます。

```
var vid = new Whammy.Video(frames);
vid.add(canvas)
vid.compile()
```

frame には動画の 1 秒間に再生されるフレーム数を代入します。

canvas には描画した canvas のコンテキストを代入しておきます。

最初の一行目の命令で動画を作成するための入れ物を作成します。

二行目の命令を 1 回実行するたびに canvas の内容が 1 つのフレームに記録されます。この命令はフレームに canvas を代入するたびに実行する必要があります。そして、三行目の命令で動画ファイルとしてデータを扱うことができるようにします。かなり簡単に作成されているため、便利ですが、音声を入れることが出来ないのは残念です。

★★★ Topic9 Web ブラウザ上で動画を作ろう!

□実際に作ってみる

ではライブラリの使い方がわかったところで実際に使ってみます。

今回は円をぐるっと一周書くアニメーションを作ってみます。

ソースコードは以下の様になっています。

```
<!--index.html-->
<canvas id="canvas" width="150"
height="150"></canvas>
<video id="awesome" width="150"
height="150" controls
autoplay></video>
<script src="whammy.js"></script>
<script src="main.js"></script>
```

index.html にフレームを描画する canvas 要素と動画を再生するための video 要素を作成します。あと、whammy.js と main.js をインクルードしておきます。

```
//main.js
//秒間20フレームの動画入れ物を作成
var video = new Whammy.Video(20);

//canvasに円を一周書くアニメーションを描画する
var ctx =
document.getElementById('canvas').getContext('2d');
ctx.beginPath();
ctx.fillStyle = "rgb(255,255,255)";
ctx.fillRect(0,0,150,150);
ctx.fill();
```

```
for (var i = 0; i <= 180; i++) {
  ctx.beginPath();
  ctx.fillStyle = 'rgb(0,0,0)';
  ctx.arc(70, 45, 35, 0, i * Math.PI
* 2 / 180, false);
  ctx.stroke();
  //1フレーム追加する。
  video.add(ctx);
}
```

//動画ファイルとして扱えるようにする。

```
var output = video.compile();
```

//動画ファイルのデータのurlを作成

```
var url =
webkitURL.createObjectURL(output);
```

//html内のvideo要素のソースを作成した動画ファイルに設定する。

```
document.getElementById('awesome').src
= url;
```

□プログラムがしていること

- ・ 1.png～7.png 貼り付け
- ・ 各図の名前は 0°、30°、60°、90°、120°、150°、180°

この様に角度の違う円弧を 1 フレームずつ canvas に描画して動画ファイルに追加しています。

実際に作成された動画は Web 上で見られるようにしてあります。

□まとめ

whammy.js を使うことで javascript の描画関数を使った動画を作成することが可能になっています。そのため、複雑なアルゴリズムの動きをグラフィカルな動画にして

★★★ Topic9 Web ブラウザ上で動画を作ろう!

見ることが簡単に出来ます。さらに、応用することで既存の動画に対しての編集を行うことも可能です。また、OpenCVなどを組み合わせることでもかなり高度な編集を行うことも可能となっています。このwhammy.jsというライブラリはプログラミングによって動画を行うことを非常に簡単なものになっています。プログラミング初心者の人でも簡単に扱うことが出来ます。ぜひ、プログラミング初心者の方も、このライブラリを使って動画を作ってみてください！

Column 大手キャリアから格安 SIM に乗り換えて留年する(3/3)

デメリットの 1, スマホを買い換えなければならないことがある。(必ずではない)

につきましては、iphone などの端末では実は格安 SIM が使えないことがあるということです。

厳密には使えなくはないというところですが、モバイル SIM 各社公式の手順を踏まないことで通信が可能になる、

という現状であり、今使用している端末によっては乗り換えを推奨することができない場合があります。

まあ、新しい端末を買うということであれば話は別です。

・・・っと、他にもここに書くべき内容は大いにありますが、このコラムを読んで格

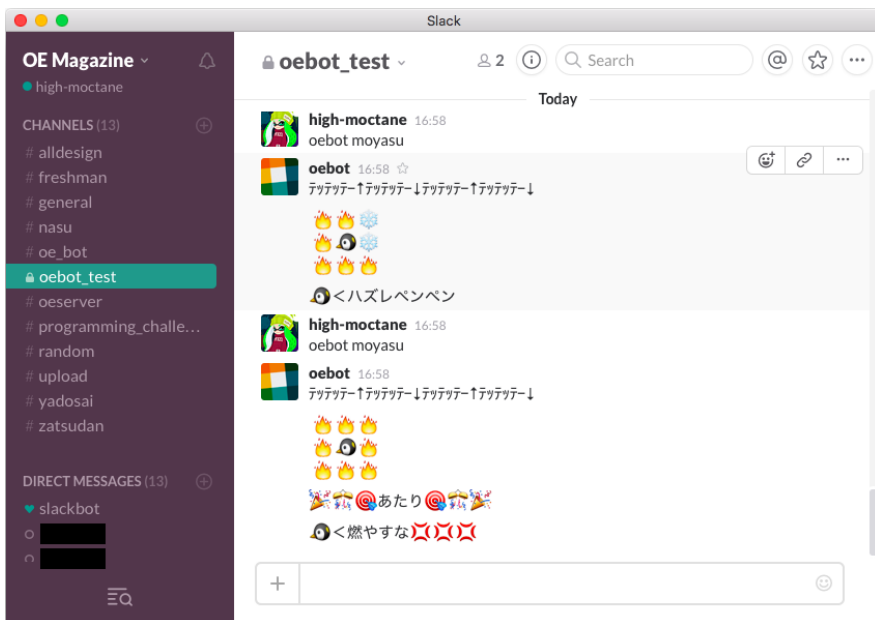
安 SIM に興味が湧いた！といった読者諸君の皆様には

ぜひ、検索エンジンに“格安 SIM”のワードを入力し、新たな知識を自分の力で身に付けていただきたいと思います。

無知の所為で人生知らぬ間に損をしている、ということは大いにあるので日頃から多くの知識を頭に入れておきましょう。

(決してコラムを書くのが面倒になった、というわけではございません。あしからず。)

Column Slack を使おう(´・ω・｀)!!!!!!



エシス2年次のはいもくたんです。

今回お勧めするのは Slack です！

サークルなどの団体で LINE やメールを使っている方は必見のツールです。今すぐ Slack を使いましょう！

Slack はグループで使える無料チャットツールのようなものです。Team と呼ばれるグループ内に Channel と呼ばれる小グループを作ることができます。掲示板でいうところのスレッドのように、話題ごとに作るのが一般的です(´・ω・｀)

この Channel が Slack の目玉機能のひとつです。メンバーは必要な Channel にだけ参加することができます。OpenEsys では例えば #freshman チャンネルでは新歓に関する話し合い、#yadosai では宿祭に関する用途として分けています(´・ω・｀)

LINE では話題ごとにグループを乱立させる必要がありました。皆さんの LINE のトークは複数の団体の目的別のグループで溢れかえっていませんか？

Slack を使えば一発解決！！

団体ごとに Team を切り替えることができますし、各 Team には話題別の Channel がまとまっています。しかも自分とは関係のない Channel には参加しなくていいのです(´・ω・｀)!!! これはもう乗り換えるしかない(´・ω・｀)!!!

LINE やメールでできることは Slack でできてしまうので、頼むから私の属している団体は今すぐメールと LINE をやめて Slack を使え(´・ω・｀) 皆さんも Slack の導入を検討してみてください(´・ω・｀)！

奥付

OpenEsys 新春号 初版

- ・ 発行元 OpenEsys
- ・ 連絡先 esys.magazie@gmail.com
- ・ 発行日 2016/02/18