

Development of IoT Curricula for Children (Age 10 to 13)

Naohiko Shimizu

Tokai University, Japan

March 6, 2020



1 / 40

ラズベリーパイによる小中学生から学べる本格的プログラミング教室

子どもIT未来塾

主催：青梅市教育委員会・羽村市教育委員会・公益財団法人青梅佐藤財団
後援：福生市教育委員会

「子どもの能力には限界が無い」の理念のもと基礎から最新技術まで学べる本格的プログラミング教室。「自分でゲームをつくりたい!」「家電製品制御をしたい!」「WEB データ活用をしたい!」等々……将来ITの道へ進みたい君のための未来塾!

■開催日時■ (詳細日程は裏面参照)

2020年6月～10月 (各月土日各1回・全10回: 13時～17時)

■開催会場■

青梅市文化交流センター・羽村市産業福祉センター他

■講師■

- ・東海大学教授 清水尚彦 ・会津大学准教授 奥山祐市
- ・ゲームプログラマー おにたま先生(武田寧)



Agenda

- ▶ Motivation
- ▶ Outline of the course 2019
- ▶ Changes between 2018 and 2019
- ▶ Developing materials
- ▶ Outcome of children
- ▶ Conclusion
- ▶ Appendix

Motivation

- ▶ We need talents for next generation engineering
- ▶ Elementary schools lack their facility of computing
- ▶ Ome-sato foundation want to support education of young people
- ▶ Hypotheses : Children will learn quickly on IoT with a small knock.



Outline of 2019 course

- ▶ Total 40 hours lectures/exercise (5 months, one weekend/month)
- ▶ 20 children + 5 adults attended out of 40 or more requests
- ▶ Fee: 7,000 yen/attendee (50 euros) includes:
Raspberry Pi 3, Power adapter, Sensor board, Keyboard, Mouse, HDMI cable, microSD card, WEBCAM, Headset, Textbook.
- ▶ Teaching staff: 3 lecturers, 8 teaching assistants
- ▶ Develop original materials (examples, quiz) and textbooks with 2,400,000 yen budget (20,000 euros) for both 2018 and 2019 courses. 6 students worked on it. 1,500 hours in total.
- ▶ We will not hide the technical details of IoT to children. Let children use the same tools as engineers.

2019 Course Contents

No.	theme	pages
1	Introduction, installation, basic operations, WEBCAM, HTML, GIMP	117
2	Modify video games source codes	42
3	Use sensors for temperature, humidity, illuminance, etc.	37
4	Programs for moving objects on screen	40
5	Sensors, Actuators, Infrared remote control programming	44
6	Synthesize voices, Speech recognition	19
7	Basics of Internet: TCP/IP, NAT, DNS, HTML, CGI	66
8	WEB scraping, programming with online data, Double byte character encodings	78
9	Making original program, Question and Answers	-
10	Presentation and peer evaluation	-
TOTAL textbook pages:		443

Major Changes from 2018 to 2019

2018	2019	Reason
Command line operations	GUI for file operations	Children did not accustom to keyboard and directory structures.
Typing lesson in a lecture	Online typing lesson for homework	same as above
No homework but suggest review	Settle quiz to complete	Make the requirement more obviously.
NA	Completion seal sheet for encouragement	We need some indication of the children's progress.
Theory first (50% talk, 50% exercise)	Experience first (20% explanation, 80% exercise)	They lose their concentration on long talk.
Mixed age group	Group by age	Clarify the effectiveness of the course against younger children

Developing Materials

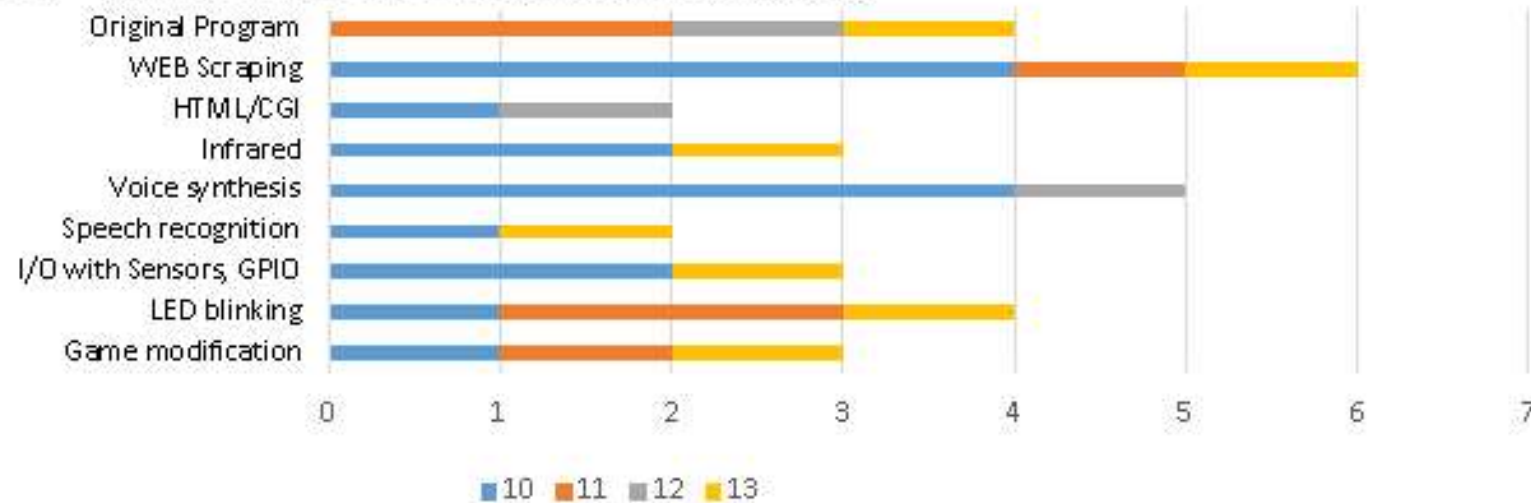
- ▶ Distributed developing with 8 developers: located 300km apart
- ▶ Tools: Gitlab, Skype, Slack, Google Documents, Wiki, (Docker)
- ▶ Pay students (masters and undergraduates) for their developing efforts
- ▶ Common Raspberry PI system images for all developers.
- ▶ Cross compiling environment for Raspberry PI including kernel development.
- ▶ Languages: C, Python, OpenHSP(Hot Soup Processor)
- ▶ Problems: Bugs in the official distribution of Raspbian (Audio output for USB / HDMI switch, LIRC driver buffer length, etc.), Discontinued sensor forced us to change our programs, MP3 unsupported in some programs, wireless mouse prevent HSP to use keyboard, interaction between different programs fails occasionally, etc.

Outcome of children

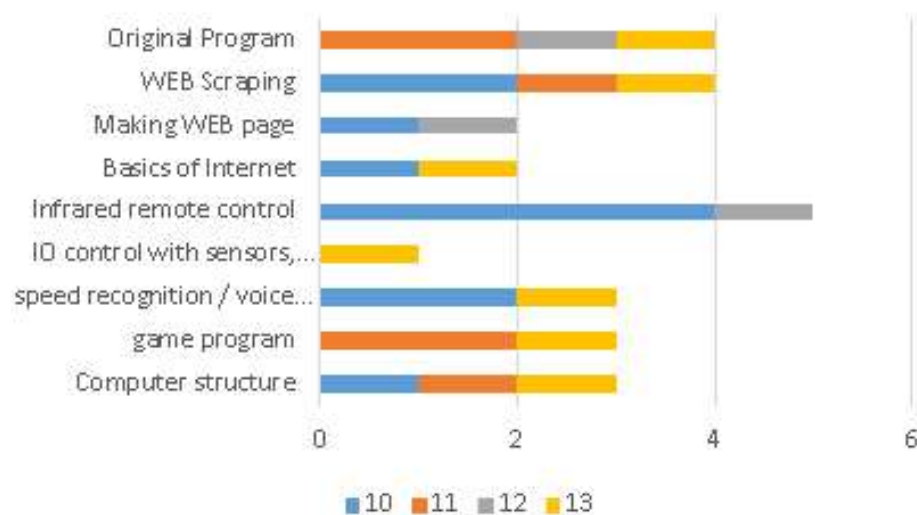
- ▶ Every children build their own program
- ▶ They made oral presentations
- ▶ Also they made poster sessions with demo to evaluate each other
- ▶ They got the diploma

Enquete-1

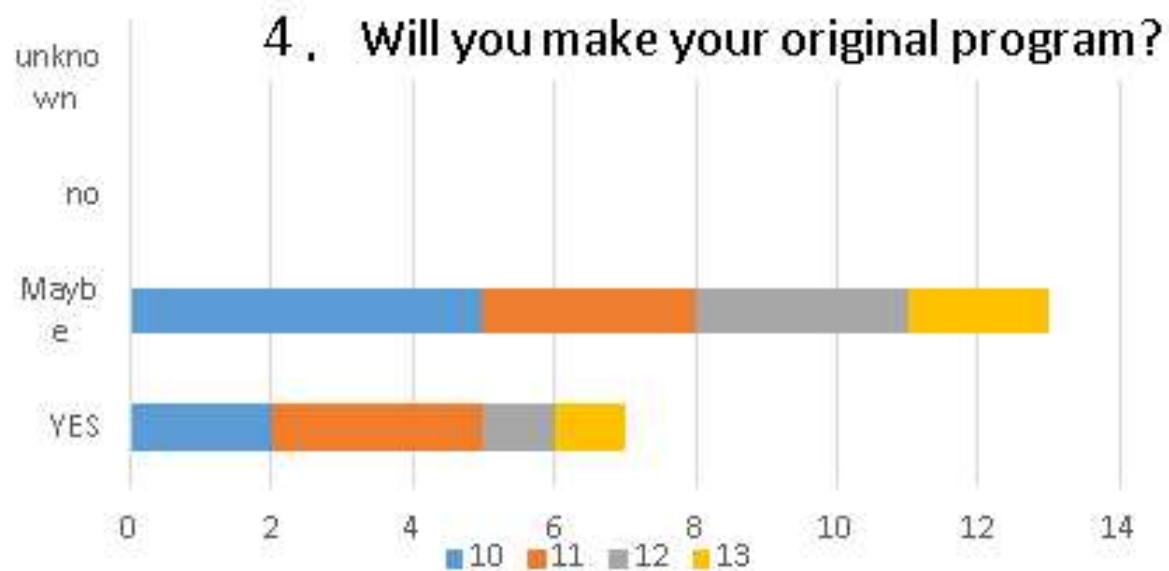
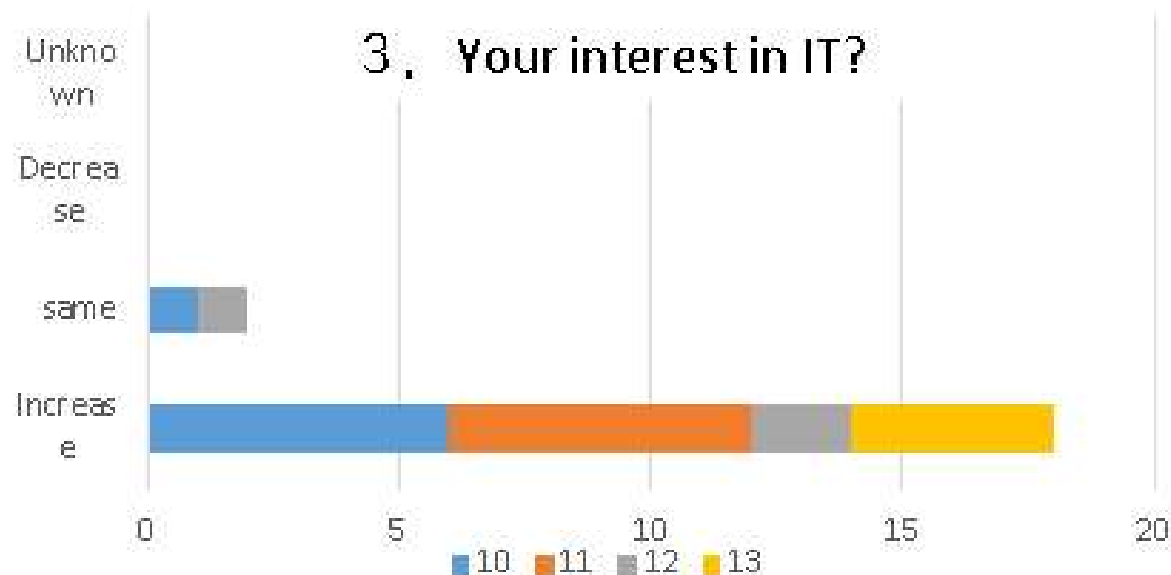
1. Interested Programs (Multiple answers)



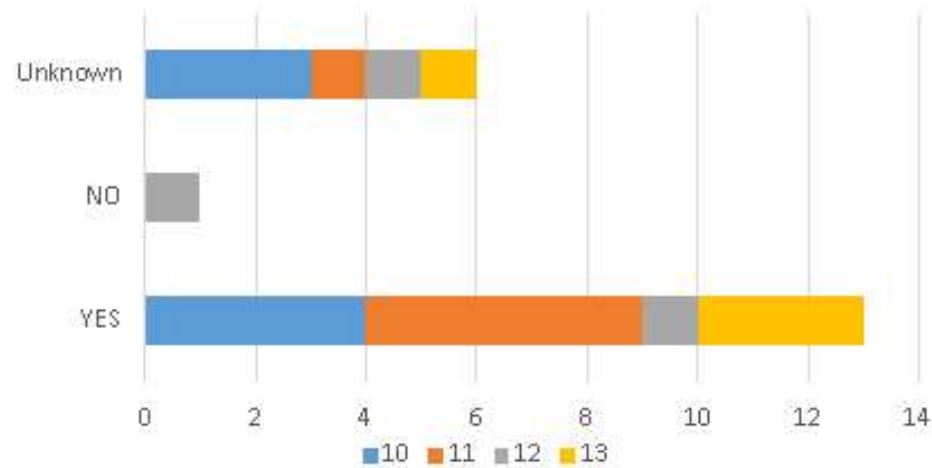
2. Not understand well



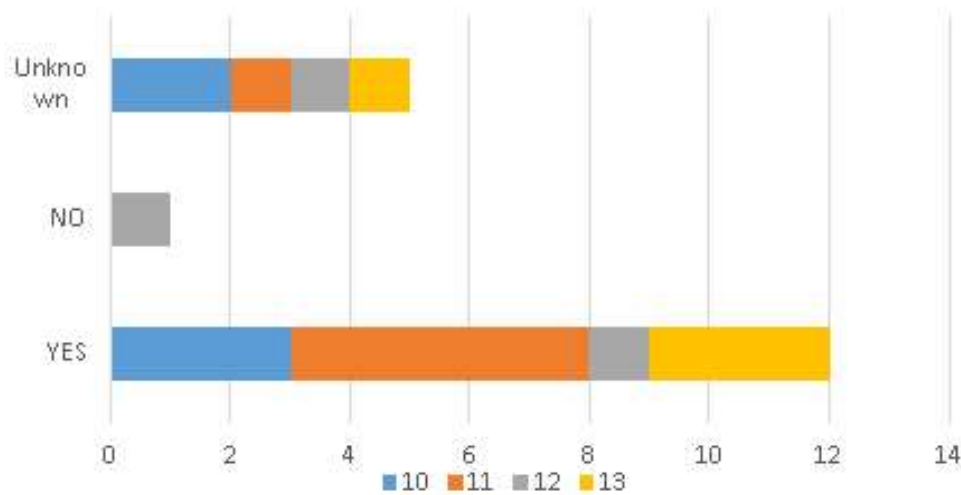
Enquete-2



Enquete-3



5. Do you want a course of computer structure?



6. Do you want advanced course?

Conclusion

- ▶ In year 2018 and 2019, we developed courses to teach IoT against children
- ▶ Each year, we had 20 attendee and we operated 40 hours program
- ▶ All children got their diploma
- ▶ Many of them increase their interest in IT
- ▶ We will have another program in 2020
- ▶ Demands for advanced course (FPGA/HW design)

Appendix: Lesson 1 installation, basic operations



Appendix: Lesson 1 Part of Textbook: File and Folders

2 ラズベリーパイになれよう (1)

2.1 ファイル、フォルダ

コンピュータやラズベリーパイの中にあるデータ（音楽、動画、写真、テキストなど）はすべてファイルとして扱われています。しばしばファイルには(ドット)から始まる「拡張子」がついており、ファイルがどんな種類なのかを表します。例えば画像の場合は.jpgや動画ファイルの場合は.mp4、テキストファイルは.txt、ホームページの場合は.htmlがついています。

ファイルの他にフォルダというものがあります。フォルダはファイルをまとめたものです。ファイルを書類としてフォルダはバインダと考えるとわかりやすいでしょう。フォルダのなかにフォルダをいれることもできます。

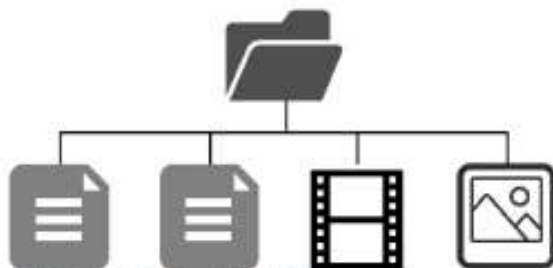


Figure 15: フォルダ、ファイルの関係

例題 1-1 フォルダを作成しよう

自身のフォルダを開き、Documents フォルダ直下に taro, hanako, saburo という名前のフォルダを作成しよう

考え方

1 ディスプレイの左上のアイコンをクリック

2 上の図にファイルマネージャーが表示されます

3 左側の空白部分で右クリックし、新規作成からフォルダを選んでクリック

4 フォルダ名の入力を求められるので、付けたい名前を入力しよう

Appendix: Lesson 1 Part of Textbook: GIMP

例題 1-11 画像に絵をかこう

“GIMP” ソフトを立ち上げて、色付きの筆で自分の顔写真に絵をかこう
 画像を開いて、色を選択、お絵かきツールの筆を選んで書こう

考え方



本格的な画像編集、加工ソフトの GIMP を使って自分の顔写真などの画像を編集してみよう

- 1 編集したい画像ファイルを右クリックし GIMP をクリック
- 2 画像編集モードになるよ
 棒で囲まれた中にある十字の交点と水色で囲まれた中にある黒のバーを動かすことによって鉛筆やスプレーアートの色を変えることができるよ戻ったら OK をクリックしよう
- 3 赤い四角で囲まれている部分をクリック
- 4 色を変更しよう
好きな色を選んで OK を押します
- 5 次に絵かきツールを選びます。
青色で囲われているところをダブルクリックしてください。これで色付きの筆が使えるようになります。
- 4 今度は色を赤にしました。赤く囲んであるところの色が赤に変わっていることを確認してください。

考え方(続き)



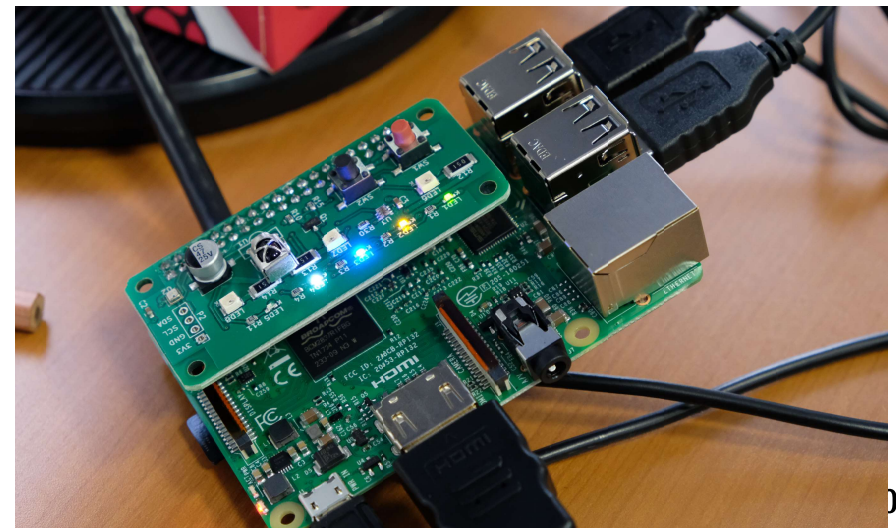
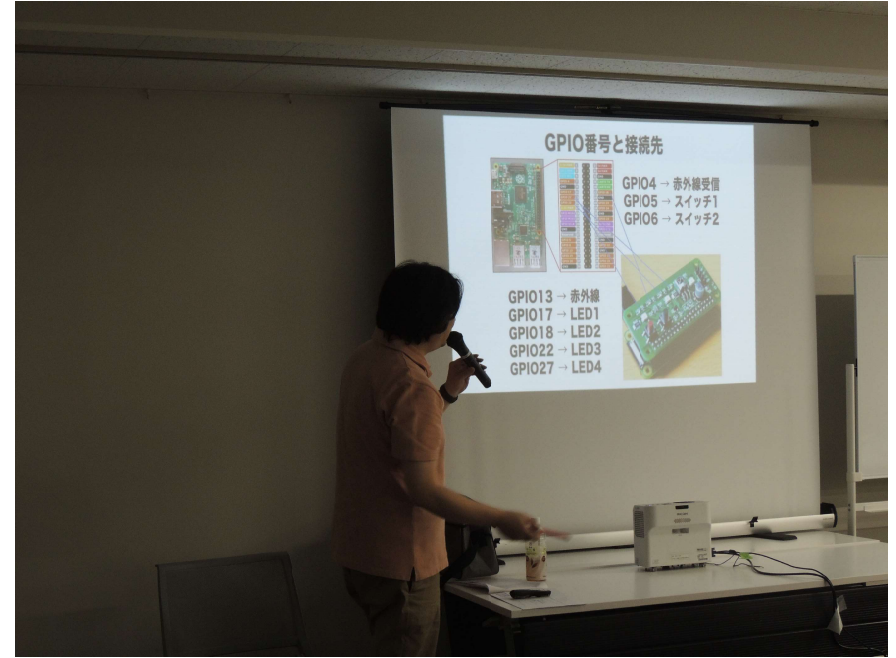
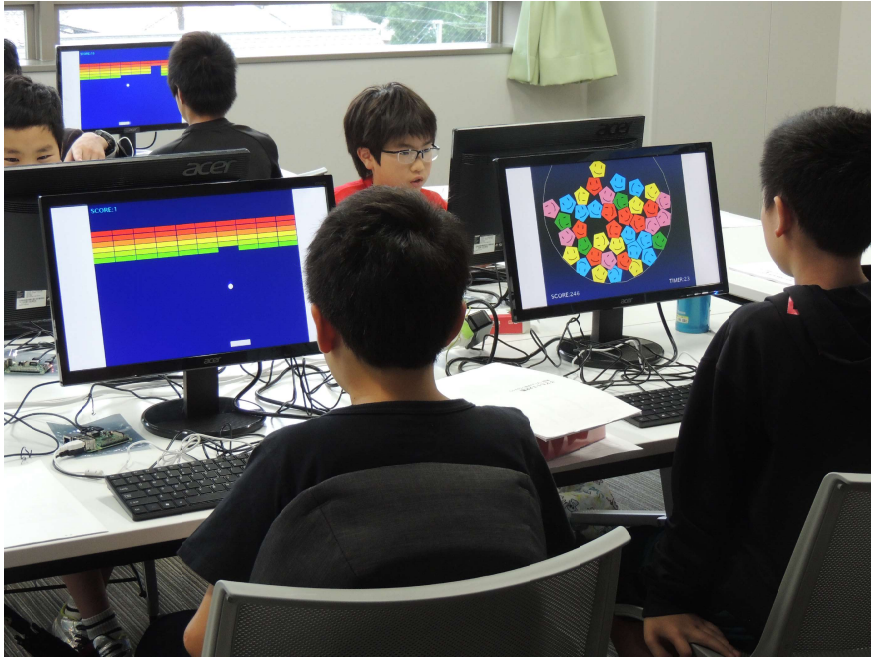
- 6 ダブルクリックしたツールは少し色が変わります。これで筆が使えるようになりました。
- 7 左クリックをおしながら画像の上を移動させると筆でお絵かきができます。いたずら書きをしてみてください
- 7 間違えてしまって、もとに戻したいと思ったらメニューの編集からもとに戻すをクリックします。
- 8 他にもたくさんツールはあります。スプレー 鉛筆を試してみよう。書き方は一筆で左クリックを押しながら移動だよ

鉛筆マークは画像の上で文字を書けるよ。加工したい画像の上で左クリックをしてみよう

スプレーマークはスプレーで画像にペイントすることができるよ。スプレーマークを選択して加工したい画像の上で左クリックをしてみよう

ここはスプレーアートなどの色の変更することができるよ。自分の好きな色を探してみよう

Appendix: Lesson 2 Modify video games



Appendix: Lesson 2 Part of Textbook: Color

例題2-6 文字に色を付けてみる

考え方

mes 命令で表示する文字に色を付けるには color 命令を使います。

color 命令

color 赤の明るさ, 緑の明るさ, 青の明るさ

(色を変えるにはcolor命令を使います)
(colorの後にスペースに続けて3つの数字を指定します)
(3つの数字は「,」で区切ること)
(0は暗い光、255は明るい光になります)

明るさとは、その成分がどのくらい明るいかということを示している数値です。
明るさは、0がもっとも暗くて、255が最も明るい状態になります。
これらを組み合わせると、1670万以上の色を出すことができます。この時に指定する色のことを RGB と呼んでいます。

RGBとは

RGB

光の3原色は赤、緑、青紫ですが、赤と緑を混ぜるとイエローに、緑と青紫を混ぜるとシアンに、青紫と赤を混ぜるとマゼンタになります。

光の3原色を等分に混合すると明るくなって、白色光になります。これを「加法混色」といいます。

RGB カラーはカラーテレビやモニタなどの「光り」で色を表現する方式です。



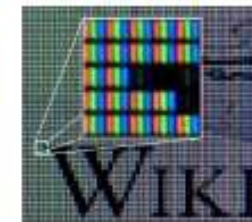
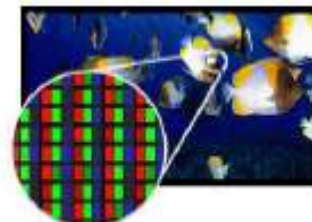
RGB の意味は R:レッド(赤) G:グリーン(緑) B:ブルー(青)の頭文字です。

赤・緑・青それぞれの光の強さを0から255までの数値で表わします。

黒は、赤・緑・青のどれも光らせない(強さ0)ことで表現します。

白は赤・緑・青すべてを光らせた時(強さ255)の色になります。

モニターで色がついているのも、すべての点で3つの色(RGB)が光っているからです。
コンピューターが、それぞれの点(ドット)を制御して光らせているのです。



mes 命令の前に書かれている color 命令の数値を変更することで、「OK!」の文字に使われている色を変えることができます。
自分で色を選んで、好きな色で文字を出してみましょう。

例題2-6 答え

必ず、mes 命令が実行されるよりも前(上の行)に color 命令を書く必要があります。
代表的な色は表のようになります。参考にして思った通りの色になっているか確認してみましょう。

```
color 0,0,0; --黒  
color 0,0,255; --青  
color 255,0,0; --赤  
color 255,0,255; --紫  
color 0,255,0; --緑  
color 0,255,255; --水色  
color 255,255,0; --黄色  
color 255,255,255; --白
```

例題2-7 文字のサイズを変えてみる

考え方

mes 命令で表示する文字のサイズを変えるには font 命令を使います。

(例)
font "",100

font 命令の後にスペースに続けて「"」、さらに「」と大きさが書かれています。
たとえば、この「100」をもっと大きな数字にしてみよう。文字が大きくなるのを確認しましょう。

Appendix: Lesson 2 Part of Textbook: Drop game

2-14 ドロップパズルの改造

ファイル→「開く」メニューから/home/pi/02のフォルダに移動してから、「drop.bmp」を読み込んでみましょう。



ドロップパズルゲームの画面

ドロップパズルゲームを改造して、自分だけのゲームに変えてみます。スクリプトは、簡単に修正できることを学びました。実は、ゲームの中の絵も自由に修正することができます。

2-15 GIMP を使ってみよう

皆さんの Raspberry Pi は、パソコンやスマホと同じように自由にプログラムを入れたり、作ったりすることができます。好きなプログラムを入れて便利にすることができるのは、コンピュータの特徴の一つです。ここでは、Raspberry Pi で絵を描くことのできるプログラムを使ってみます。Raspberry Pi メニューからグラフィックツール「GNU Image Manipulation Program」をクリックすることでツールを起動させることができます。



Gimp を起動したところ

2-16 ドロップパズルの中身を変えてみよう

Gimp を使って、ドロップパズルのゲームで使われた絵を改造してみましょう。Gimp は、絵のデータを読み込んで、自分で好きに変更することができます。変更した絵は、ふたたび保存することができます。Gimp のファイル→「開く」メニューから、「ome02フォルダ」にある「koma.png」を読み込んで下絵とします。

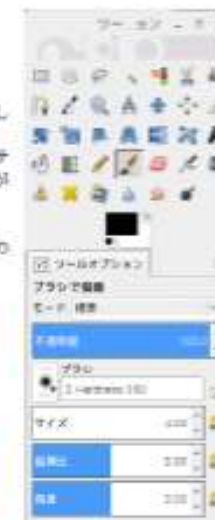
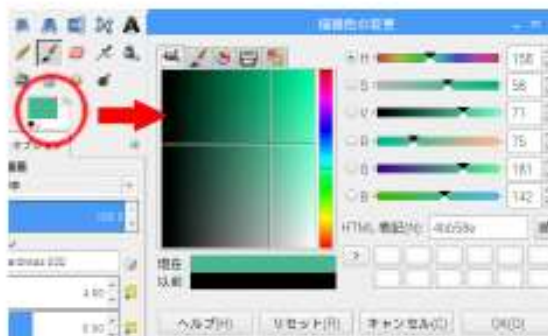


koma.png を修正している画面

このキャラクターを別な絵に書き換えればゲームの絵も変わります。パレットで色を選んで、鉛筆のツールで絵を書いていきます。元の絵があった部分を書き換えて他の絵柄にしてみましょう。

画像が小さい場合は、画像の下にある「100%」の右側にあるボタンを押して、「400%」「800%」などを選ぶと拡大されます。絵を修正する時には、ツールアイコンの中から「ブラシ」を選択して、サイズを4~6くらいの数字に変えてから絵の上で、マウスボタンを押しながら動かして色を置きます。

色を変更する場合は、ツールパレットの描画色をクリックして、「描画色の変更」ダイアログを出してから好きな色を選びます。



Appendix: Lesson 3 Use sensors



Appendix: Lesson 3 Part of Textbook: GPIO

3.2 きほんの HSP プログラム

3.2.1 LED を光らせよう

まずはセンサーボードにある LED を光らせてみましょう。HSP スクリプトエディタから LED を光らせるプログラム(led.hsp)を読み込んで実行してみましょう。led.hsp は、/home/pi/ome/03 のディレクトリにあります。

```
#include "hsp3dish.as" ;スクリプトの設定を読み込む
#include "rpz-gpio.as" ;スクリプトの設定を読み込む

redraw 0 ;画面更新 (仮想画面に描画)
font "",30 ;文字のフォント、サイズを決める
pos 20,20 ;文字の場所を決める
mes "LEDが光るよ" ;文字を決める
redraw 1 ;画面更新 (実際の画面に描画)

*led
gpio 17, 1 ;GPIO17を点灯させる
await 100 ;0.1秒待つ
goto *led ;*ledまで戻る

gpio 17, 0
```

かならずいれましょう。

ディスプレイに表示するものをきめています。

LED を光らせています。

下の写真のように LED が光りましたか？



それではプログラムを解析してみましょう。font 命令や pos 命令、mes 命令がありますね。これらがどんな命令だったかを前回の教科書を読んで復習しましょう。gpio 命令も前回勉強しました。17 という数字は上の接続図の GPIO17 に対応していて、1 という数字は ON をあらわすのでしたね。

問題 3-2-1 (第 2 回の教科書を見ながらやりましょう)

1. LED を光らせる命令を書きましょう(ヒント: 教科書第 2 回 2-9 LED が点滅するスクリプトを参考にしましょう)。

答え、_____

2. LED を消す命令を書きましょう(ヒント: 教科書第 2 回 2-9 LED が点滅するスクリプトを参考にしましょう)。

答え、_____

3. ターミナルを使って led.hsp のコピーを、321.hsp という名前で作りましょう。

← できたらチェックしましょう。

4. 他の色の LED が光るように 321.hsp のプログラムを変えてみましょう。

← できたらチェックしましょう。

Appendix: Lesson 3 Part of Textbook: Humidity, etc.

3.2.3 温度、湿度（しつど）、気圧、明るさを調べてみよう

次はセンサーボードに繋がっている BME280（湿湿度気圧センサー）と、TSL2572（照度センサー）を使ってみましょう。HSP スクリプトエディタで /home/pi/ome/03/sensors.hsp を読み込み実行してください。

```
#include "hsp3dish.as"
#include "rpz-gpio.as"

i2c_ch_bme = 0 ; bme280 に使う I2C チャンネルを 0 に設定
i2c_ch_tsl = 1 ; tsl2572 に使う I2C チャンネルを 1 に設定

fail = init_bme(i2c_ch_bme) ; 湿湿度気圧センサ bme280 を初期化する
if fail { ; 初期化成功チェック
  redraw 0
  mes "failed to init bme: " + fail
  redraw 1
  stop
}

init_lux i2c_ch_tsl ; 照度センサ tsl2572 を初期化する

*main
  redraw 0 ; 仮画面に描画
  pos 60, 60 ; 表示位置を設定

  temp = get_temp(i2c_ch_bme) ; 温度取得
  hum = get_humidity(i2c_ch_bme) ; 湿度取得
  press = get_pressure(i2c_ch_bme) ; 気圧取得
  lux = get_lux(i2c_ch_tsl) ; 照度取得

  ; 取得したデータの表示
  mes "温度: " + temp + " [C]"
  mes "湿度: " + hum + " [%]"
  mes "気圧: " + press + " [hPa]"
  mes "照度: " + lux + " [lx]"

  redraw 1 ; 実際の画面に描画
  wait 100
  goto *main ; 繰り返し
```

湿湿度気圧照度センサーを使うときはおなじように書きましょう。

センサーの値をゲットしています。

センサーの値を表示するための文字を決めています。

BME280 と TSL2572 は I2C (アイスクエアドジー、と読みます) と呼ばれる規格でラズベリーパイと通信します。HSP は内部で I2C のための通信経路 (チャンネルと呼びます) を 32 個持ちますが、経路 1 つに対して 1 つのセンサーとしか通信できません。従って、これらの経路はプログラマが管理しなければなりません。

```
i2c_ch_bme = 0
i2c_ch_tsl = 1
```

という 2 行は、0 番目の経路を BME280 に、1 番目の経路を TSL2572 に割り当てるための変数を宣言しています (プログラミングの世界では、しばしば順番号が 0 から始まります)。

まずは BME280 に関する行について説明します。

init_bme

という命令で BME280 を初期化しています。実際にはキャリブレーション (補正) のパラメータ取得をしたり、動作モードを決定したりしています。気になる方はデータシート (英語) を読んでみましょう。

この命令の値が変数 fail に代入されています。初期化に失敗すると success は 0 以外の値が入るの

で、次の行文で確認しています。

init_bme によって、get_temp、get_humidity、get_pressure が使えるようになりました。これらはチャンネルの番号 i2c_ch_bme を引数にとり、そのチャンネルが BME280 に繋がっているとみなして通信を行います。

```
temp = get_temp(i2c_ch_bme) ; 温度取得
hum = get_humidity(i2c_ch_bme) ; 湿度取得
press = get_pressure(i2c_ch_bme) ; 気圧取得
```

のようにすることによって、変数 temp に気温 (temperature)、変数 hum に湿度 (humidity)、変数 press に気圧 (pressure) が入ります。

次は TSL2572 に関する行を説明します。

init_lux

によって TSL2572 が初期化されます。BME280 のときと違って返り値を取らず、エラーチェックをする必要が無いことに気をつけましょう。実際には感度の設定等が行われます。詳しく知りたい方はデータシート (英語) を読んでみましょう。

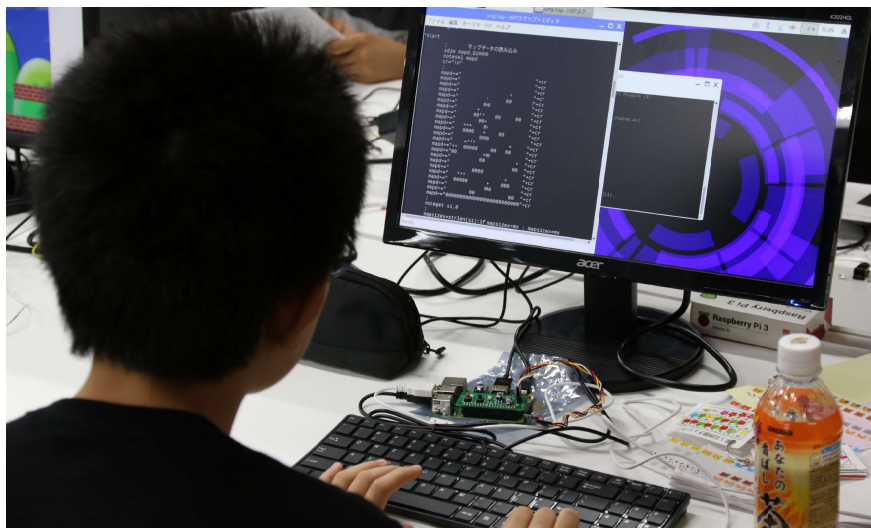
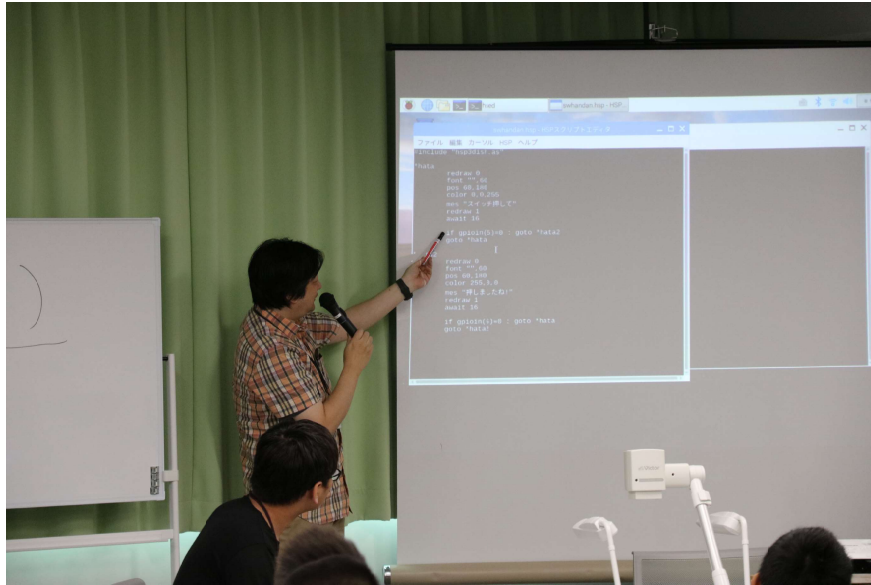
初期化によって、get_lux が使えるようになりました。これはチャンネルの番号 i2c_ch_tsl を引数にとり、そのチャンネルが TSL2572 に繋がっているとみなして通信を行います。

```
lux = get_lux(i2c_ch_tsl) ; 照度取得
```

とすることで、変数 lux に照度が入ります。

これらによって得られた temp、hum、press、lux を mes 命令を使って画面に表示しています。

Appendix: Lesson 4 Programs for moving objects



Appendix: Lesson 4 Part of Textbook: Moving Pictures

4-7 絵を動かしてみよう

もう少し別な絵を表示してみましょう。
スクリプトエディタの、ファイル→「開く」メニューから ome/04ディレクトリの中にある「apple.hsp」を読み込んで実行してみてください。



apple.hsp の実行画面

今度は、画面の中央に「りんご」が表示されていることに気付いたでしょうか。

実は、このスクリプトでは背景と「りんご」の2つの画像を使っています。
背景の上に「りんご」を表示しているのです。



リンゴの画像 (apple.png)

```
#include "hsp3dish.as"

cellload "sozal4.jpg",1 ; 背景
cellload "apple.png",2 ; りんご

x=300:y=200

*main
  redraw 0
  pos 0,0
  celput 1 ; 背景を表示
  amode 2 ; 重ね合わせる
  pos x,y ; りんごを表示
  celput 2
  redraw 1
  await 1000/30
  goto *main
```

13

このように複数の番号を登録して、celput 命令を使うことで絵を重ねて表示することができます。
そのために、gmode という新しい命令が使われています。

(HSPのルール)

画像を重ねる設定をするには gmode 命令を使います
gmode の後、スペースに続けてコピーモードを示す数字を指定します
コピーモードは、以下のような意味があります

- 0のとき : 画像のまま表示
- 1か2のとき : 透明な部分は背景を映して表示
- 3のとき : 透明度を変えて表示

「gmode 2」を指定することで、リンゴの透明な部分は背景が表示されて、きれいに重なっているように見えています。試しに、「gmode 0」にしてリンゴの絵がどのように表示されるか確認してみましょう。リンゴのように透明な部分がある絵は、以前にも使った GIMP ツールで作ったり、確認することができます。GIMP は、Raspberry Pi メニューからグラフィックス→「GNU Image Manipulation Program」をクリックすることで起動させることができます。



Gimp を起動したところ

celput 命令には、実はもっと多くのパラメーターがあります。

celput 絵の番号, 分割画像 No., 横方向の表示倍率, 縦方向の表示倍率, 回転角度

横方向の表示倍率=0.0~(1.0): 横方向の表示倍率(実数)
縦方向の表示倍率=0.0~(1.0): 縦方向の表示倍率(実数)
回転角度=0.0~(0.0): 回転角度(実数)

絵を自由な大きさ、角度で出すことができるようになっています。
通常は、倍率1、角度0の状態が表示されます。
分割画像 No. は、1つの画像を複数の小さなブロックに分けて表示するための機能です。
(これについては、また別の機会に説明します。)

試しに、表示倍率や角度を変えてみて、どのように表示が変わるか確認してみましょう。
また、余裕がある人は、文字を重ねてさらに複雑な画面を作ることに挑戦してください。

4-8 絵を動かすには

今度は、表示した絵を動かしてみることにしましょう。
絵が動くことで、ゲームやアニメーションなどさらに応用が広がります。
そのために覚えておかなければならないのが、変数です。

14

Appendix: Lesson 4 Part of Textbook: Make Original Screen

4-11 ジャンプアクションゲームに挑戦

ジャンプアップゲームでキャラクターを動かして遊ぶ本格的なゲームを体験してみましょう。

ファイル→「開く」メニューから「jump.hsp」を開き込んでください。
作業は、すべて「/home/palomae/04」ディレクトリで行ないます。
見つからない場合は、まわりの友達か、近くの先生に聞いてみてください。

[F5]キーで実行すると、タイトル画面が出ます。[Enter]キーを押して始めてください。



タイトル画面

ジャンプアップゲームは、キャラクターを操作してコインを取るゲームです。左右の移動とジャンプをさせることができます。



例題4-8 オリジナルの面を作ってみよう

考え方

ジャンプアップゲーム(jump.hsp)のプログラムを改造してゲームで使われている面(マップ)を変えてみましょう。
プログラムを編集することで、面のデータを自分で変えることができます。
以下の場所を探して編集します。

```
*start
; マップデータの読み込み
sdim mapd,$10000
notesel mapd
cr="\n"
;
mapd+="                               "+cr
mapd+="                               "+cr
mapd+="                               "+cr
mapd+="                               "+cr
mapd+="                               *   *   "+cr
mapd+="                               000  "+cr
mapd+="                               00** 00 00 "+cr
mapd+="                               06    00 "+cr
mapd+="          +++++ 00    "+cr
mapd+="          0000  00    "+cr
mapd+="          0000  "+cr
mapd+="          ***    "+cr
mapd+="          00000  00 00 "+cr
mapd+="          00    00    "+cr
mapd+="          00    *   "+cr
mapd+="          00    00 "+cr
mapd+="          0000  "+cr
mapd+="          ***    "+cr
mapd+="          00000  000  "+cr
mapd+="          00    000  "+cr
mapd+="          000000000000000000000000000000 "+cr
```

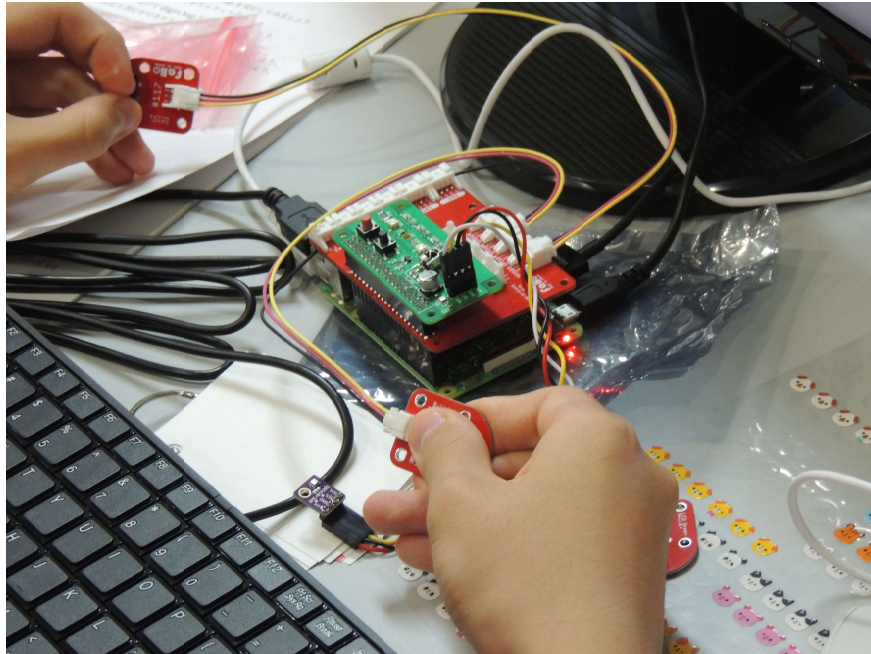
「0」はレンガ(壁)になります。
「*」はコイン、「+」は矢になります。
マップデータが書かれている場所を探して、自分だけの面を作ってみましょう。

でたために書き直してもエラーが出るだけです。必ず、

```
mapd+="                               "+cr
```

という形になるようにしてください。
文字は「」で囲み、最後に「+cr」を入れます。
面のデータは、縦・横方向に広げることができます。

Appendix: Lesson 5 Sensors, Infrared Red remote control programming



Appendix: Lesson 5 Part of Textbook: Analog Sensors

3.4 アナログ入力そうちを使ってみよう

アナログ入力そうちを HSP で使ってみましょう。アナログ入力そうちを A0 につなげてください。



HSP スクリプトエディタで `/home/pi/ome/05/anain.hsp` を開いて実行してみましょう。

```
#include "hsp3dish.as"
#include "rpi-gpio.as"

spiopen 0 ;SPIチャンネルを開く

*main
  data = spiget(0,0) ;SPIを使ってデータを受け取る
  res = "結果: "+data+"\n"

  redraw 0
  pos 20,20
  font "",30
  mes res
  redraw 1

  wait 10
  goto *main

spiclose 0 ;SPIチャンネルを閉じる
リスト 3: anain.hsp
```

`anain.hsp` はアナログ入力そうち用のプログラムです。アナログ入力には A0, A1, A2, A3 のピンを使います。感圧センサー、ボリューム、距離センサー、湿度センサーを使うことができます。

アナログ入力そうちから入力を受け取るときに SPI (Serial Peripheral Interface) という方法を使っています。 `spiopen` 命令を使って使う SPI デバイスのチャンネルを設定しています。 `spiopen 0` で Fabo のシールドに 0 チャンネルを設定します。チャンネルは変えないようにしましょう。

アナログ入力は `spiget` (ピン番号, チャンネル) で値を受け取ります。ピン番号はブリックがつながっている番号です。チャンネルは 0 にしておきましょう。入力される値は 0-1023 までの数字になります。詳しいアナログ入力そうちの使い方は 39 ページから見てみましょう。

3.5 距離センサーを使ってみよう

距離センサーを HSP で使ってみましょう。距離センサー(番号は書いてありません。41 ページを参考にしてください)を A0 につなげましょう。HSP スクリプトエディタ `/home/pi/ome/05/kyori.hsp` を開いて実行してください。



図 3.5: 距離センサー



図 3.6: A0

```
#include "hsp3dish.as"
#include "rpi-gpio.as"

spiopen 0

*main
  data = spiget(0,0) ;SPIを使ってデータを受け取る
  kyori = -1*(data*5000/1023)/36+845/9 ; data を距離に変換しています
  res = "距離: "+kyori

  redraw 0
  font "",20
  pos 30,30
  mes res
  redraw 1

  wait 100
  goto *main

spiclose 0
リスト 4: kyori.hsp
```

データは `spiget` を使って受け取ることができます。値は 0-1023 になります。これでは距離がわからないので、受け取った値を距離に変換する必要があります。変換するための式は 41 ページに書いてあります。今回は入力は `data` という変数にしているので、式の `x` を `data` にします。 `kyori = -1*(data*5000/1023)/36+845/9` で入力を距離に変換し、 `kyori` 変数に代入しています。単位は cm です。

問題 5-3-5

1. 距離センサー(番号は書いてありません。41 ページを参考にしてください)を A0 につなげて `anain.hsp` を実行してみましょう。距離はだいたい 5-80cm です。
 → 距離を測ることができたらチェック

Appendix: Lesson 5 Part of Textbook: Infrared Explanation

4 赤外線

4.1 赤外線ってなんだろう？

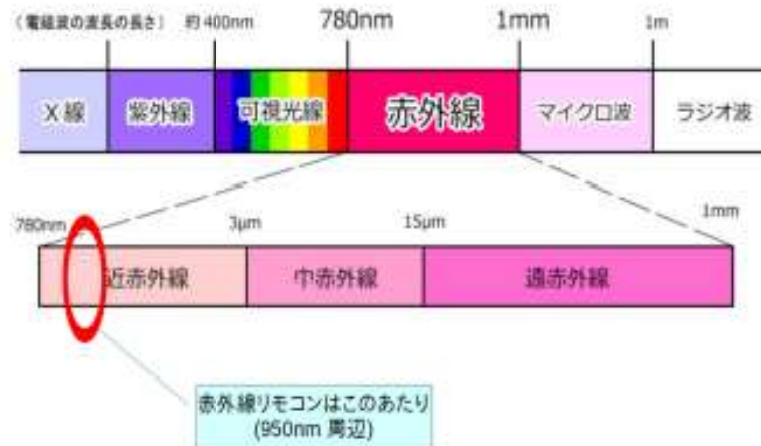


図 4.1: 電磁波。出典: 家電 Watch <https://kaden.watch.impress.co.jp/cda/word/2008/11/21/3196.html>

物質や生物はそれぞれ電磁波を発しています。X線、紫外線、光、マイクロ波などは電磁波の一種です。電磁波は波であり、どれだけ振動しているかで種類をわけることができます。わたしたちは可視光線と呼ばれる光が目に入ることで、色などを目で感知することができます。マイクロ波は電子レンジに使われています。マイクロ波によって物質を振動させることで、物質の温度をあげています。

赤外線は人の目には見えませんが、ほとんどすべての物体が赤外線を発しています。特に温度の高いものは赤外線を強く発しています。Raspberry Pi は発熱しながら動いています。Raspberry Pi に触れなくても、手を近づけるだけで熱を感じるのには、赤外線が Raspberry Pi から発せられていて、その赤外線を手で感知しているからです。

赤外線を使って信号を送ったり、受け取ったりすることもできます。例えば赤外線が送られているときは電気がつける、赤外線が送られていないときは電気を消すことができます。しかし送られているかどうかだけではつけない、消すしかできません。ロボットのように歩く、止まる、右に曲がる、左に曲がるなど多くの動きを制御したいときは、信号を組み合わせて送ります。



数百 - 数千 μs μs (マイクロ秒): $1/1000000$ 秒

図 4.2:

図 4.2: を見てみましょう。電球が点いているときは赤外線を送り、電球が点いていないときは赤外線を送っていないという意味です。例えば $1/1000000$ 秒の間隔で赤外線を送ります。つけたり消したりを組み合わせて、いろいろなパターンを作ることができます。そのパターンによって制御を決めています。送る、送らない、送らない、送らない、送る、送る、送らないと信号が来たときは「歩く」と制御します。

4.2 赤外線をつかった家電

もっと身近なものと考えてみましょう。赤外線を使った家電はなにがあるのでしょうか？



エアコン、テレビ、照明などはリモコンを使って動作を制御しています。ボタンを押せば電源が点いたり、風量や音量を調節できます。これは赤外線を使って家電を制御しています。電源を入れるときはリモコンから電源を入れるための赤外線信号が送られます。家電はそれを受け取り、信号で決められた動作をします。

Appendix: Lesson 5 Part of Textbook: Infrared Remote Controller

例題 5-4-3 リモコンの信号をコピーしてみよう

ターミナルを開いて、コマンドを入力しましょう。

1. リモコンの信号を受け取り、ファイルに記録します。
 1. 違う信号を受け取らないように、赤外線受信を止めます。
`sudo service lircd stop`
 2. 赤外線を受信し、`onoff.txt`に記録します。コマンドを実行してから、赤外線受信ユニットに向かって、リモコンのボタンを押しましょう。記録が終わったら `Ctrl+c` (`Ctrl` キーを押しながら `c` を押す) で終了します。
`mode2 -d /dev/lirc1 | tee onoff.txt`
 3. 記録した信号を、送信用に交換します。
`convert_pattern onoff.txt > onoff.pattern`
 4. 設定ファイルを作ります。/`home/pi/one/05/template.lircd.conf` を元で作ります。まずはこのテンプレートを `05.lircd.conf` という名前でコピーします。そのあと、`leafpad` で開きましょう。
`cd /home/pi/one/05/`
`cp template.lircd.conf 05.lircd.conf`
`leafpad 05.lircd.conf`
- リスト 8: `template.lircd.conf` の赤字の部分を書き換えます。
- ① リモコンの名前を決めます。使う家電の名前 (`fan`, `robot`, `TV`...) にしましょう。
 - ② 信号の名前を決めます。動作 (`digital_onoff`, `walk`...) の名前にしましょう。
 - ③ “ここに信号をペーストする” と書いてある行を消します。 `onoff.pattern` に書かれた信号をコピーして貼り付けます。 `leafpad` を使って `onoff.pattern` を表示しましょう。
`leafpad /home/pi/one/05/onoff.pattern`
- `onoff.pattern` に書かれた信号は、たとえば `f1207 588 447 838 447 823 475 812 1218` のような文字列です。ただし、ここに書くことができる数字の数は 255 以下にしないといけないことに注意してください。書き換えた後に数を数えて、255 個以下であることを確認しましょう。
- 書き換えたら、保存しましょう (名前が `05.lircd.conf` になっていることを確認しましょう)。

```
begin remote
name fan
flags RAW_CODES
eps 30
aeps 100

gap 200000
toggle_bit_mask 0x0

begin raw_codes
name onoff
ここに信号をペーストする
end raw_codes
end remote
```

リスト 8: `template.lircd.conf`

① リモコンの名前

② 信号の名前

③ 信号

4.4 HSP を使って赤外線を出力するプログラムを作ってみよう

HSP スクリプトエディタで `/home/pi/one/05/ir.hsp` を開いてみましょう。HSP では `exec` 命令を使ってターミナルで使うコマンドを実行することができます。

```
exec "lrsend SEND_ONCE fan onoff"
```

`exec` “コマンド” と使います。コマンドは先ほどターミナルで実行した `lrsend SEND_ONCE fan onoff` です。 `fan` はリモコンの名前、 `onoff` は信号の名前でしたね、 `fan` と `onoff` は自分で設定ファイルに書いた名前に変えましょう。

```
#include "hsp3dish.as"
#include "rpz-gpio.as"

BUTTON_PIN = 24
prev = 0
status=0

*main
  gpmos *ボタンPUSHで赤外線出力!
  goto *edge

*edge
  current = gpioin(BUTTON_PIN)
  if (prev=0) & (current=1) { ;ボタンが押されたら
    exec "lrsend SEND_ONCE fan onoff" ;lrsendコマンドを実行する
  }
  prev = current
  update 1
  goto *edge
```

テキスト 1: `ir.hsp`

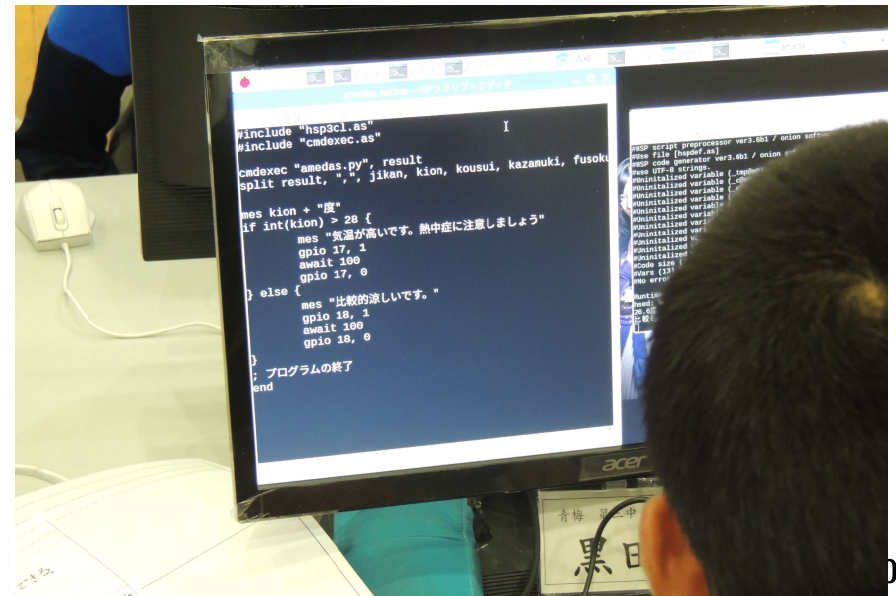
問題 4-4

1. ボタンを GPIO24 につなげ、自分でコピーした赤外線を出力してみましょう。 `fan` と `onoff` は自分で設定ファイルに書いた名前に変えましょう。
 赤外線を送信し、ロボットや扇風機を動作させたらチェックしましょう。

4.5 赤外線送受信のまとめ

1. `sudo service lircd stop`
2. `mode2 -d /dev/lirc1 | tee ファイル名 1.txt`
3. `convert_pattern` ファイル名 1.txt > ファイル名 2.pattern
4. `template.lircd.conf` を書きかえてファイル名 3.lircd.conf を作成
5. `sudo cp` ファイル名 3.lircd.conf /etc/lirc/lircd.conf.d/
6. `sudo service lircd restart`
7. `lrsend SEND_ONCE` <リモコンの名前> <信号の名前>

Appendix: Lesson 6-8 Voice, WEB programming



Appendix: Lesson 6 Part of Textbook: Synthesize voice, Speech recognition

1 音声合成

音声合成とは何でしょう。それは、人間の声を人工的に機械で作出すことです。みなさんが目にするもので言えば、例えばソフトバンクが制作する pepper(図 1)というロボットがあります。



図 1: pepper (softbank)

これは声を出して利用者とコミュニケーションを取ります。pepperから声が出るのは、中に人間が入っているわけではありません。読み上げたい文章があったとき、それにマッチした音声を生成し、スピーカーから発しているのです。

今回は音声を合成するために、OpenJTalkと呼ばれる音声合成のためのソフトウェアを使います。OpenJTalkは、読み上げたい文章と、ひらがなごとの音(厳密にはもう少し多種の音)を入力し、対応する音声ファイルを出力します(図 2)。

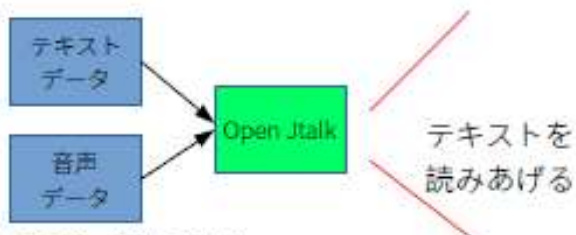


図 2: OpenJTalk の入出力

2 音声認識

音声認識は、音声合成とは逆に、文章を読み上げている音声ファイルからその文章を予想するものです。例えば Google の Google Home はまさにこれです。「1 時間後に起こして」と声を発すると、内部でそれが文字列に変換されます。その後その文字列を解析してアラームを設定します。

音声認識のためのソフトウェアとして Julius というものを使います。入力は音声で出力は文字列になります(図 3)。



図 3: Julius の入出力

問題 6-2

1. 音声認識とはなんですか。50 字以内で説明してください。

答え: _____

3 OpenJTalk

ここでは音声合成ソフトウェア OpenJTalk の紹介をします。まずは使ってみましょう。

3.1 文を読み上げてもらう

文を読み上げる HSP プログラムを紹介します。openjtalk.hsp を HSP スクリプトエディタで開いてください。openjtalk.hsp は /home/pi/ome/06/ の中にあります。

```
#include "hsp3dish.as"
#include "jtalk.as"

redraw 0
mes "電話開始"
mes "「子ども IT 未来塾」"
redraw 1

jtlload "子ども IT 未来塾", 0 ; 音声ファイルを出力し、読み込む
mplay 0 ; 音声ファイルを再生する

stop
リスト 1: openjtalk.hsp
```

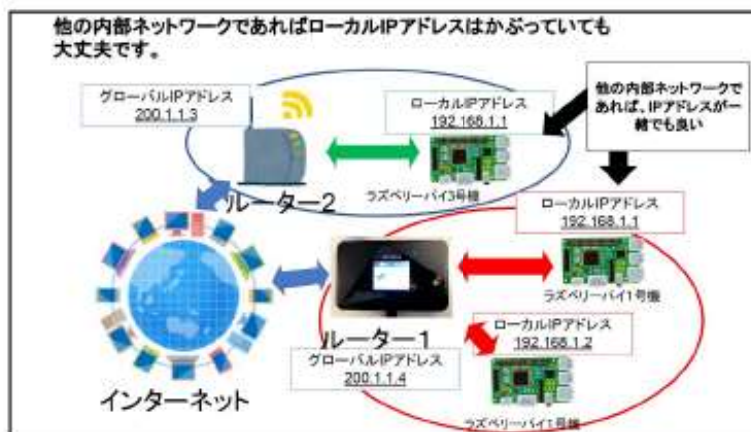
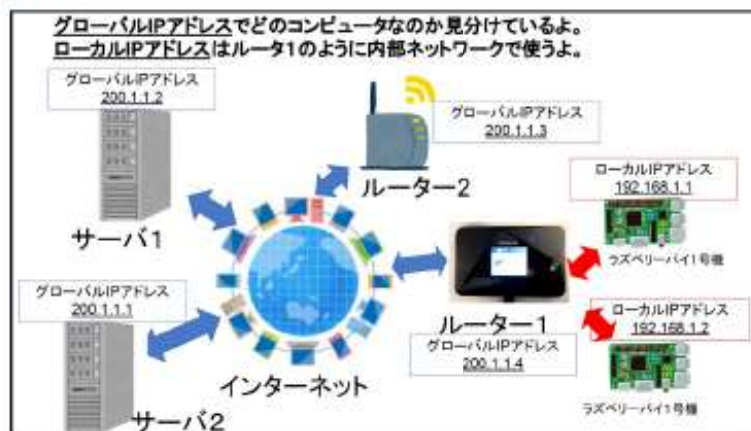
2 行目

```
#include "jtalk.as"
```

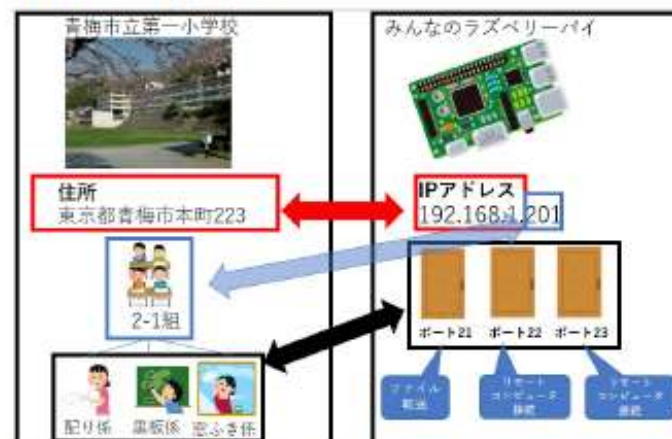

Appendix: Lesson 7 Part of Textbook: Internet Basics

1-3 グローバル IP アドレスとローカル IP アドレス

IP アドレスにはグローバル IP アドレスとローカル（プライベート）IP アドレスという二つに分けられます。



1-5 ポート番号とは



前のページではみんなのラズベリーパイがどのようにインターネットにつながっているの理解してくれたと思います。ここではさらにポート番号について知識を深めましょう。IP アドレスでは通信相手となる別のネットワークのコンピュータを特定することができます。しかし、コンピュータでは通常、多くのプログラム（機能）が動作していて、IP アドレスを指定するだけでは、どのプログラムと接続するかを区別できません。そこでコンピュータ上の、どのプログラムと接続するかを指定するためにポート番号が用いられます。下の図は小学校を例にしています。

青梅市立第一小学校の住所がみんなのラズベリーパイの IP アドレスに対応しており、IP アドレスの末尾の 201（青で囲まれている）が小学校のクラスに対応しています。皆さんの学校でも生徒にはそれぞれ役割が与えられていますね。例えば、プリントなどの配布物を配る配り係、黒板を使い終わったらきれいにする黒板係、教室の窓をきれいにする窓ふき係などこれらの係に相当するものがラズベリーパイにもあり、それがポートです。ポート 21 はファイル転送用の働きをする担当で、ポート 22 と 23 は他のコンピュータに接続する担当です。ポートには役割がありインターネットの通信はファイル転送であったり、他のコンピュータに接続して遠隔操作したり、web ページにアクセスしたりといういろいろあります。なので、その分のポートを用意しておき、役割を与え、対応している役割のときに頑張ってもらう仕組みになっています。みんなのラズベリーパイやパソコン、サーバにはポートは何個あるのか調べてみるのも面白いですね。

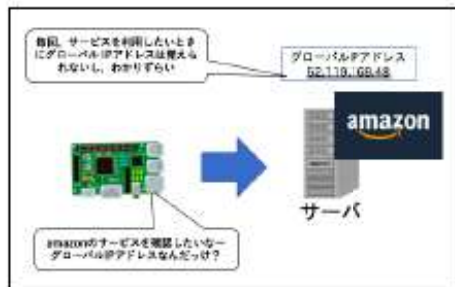
問7-4 ポート番号はどのようなことに使われますか。

答え

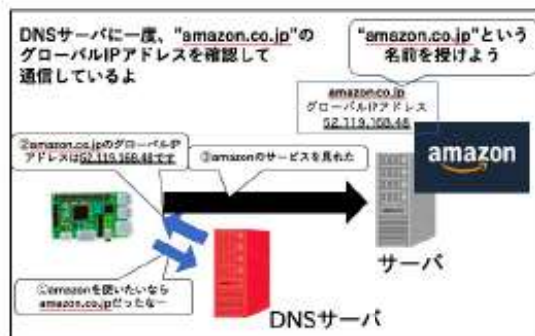
Appendix: Lesson 7 Part of Textbook: DNS, NAT

1-6 DNS について

IP アドレスは住所であり、それがわからないとサーバのサービスを受けることはできません。ですが以下の図のようなことが起こることがあります。

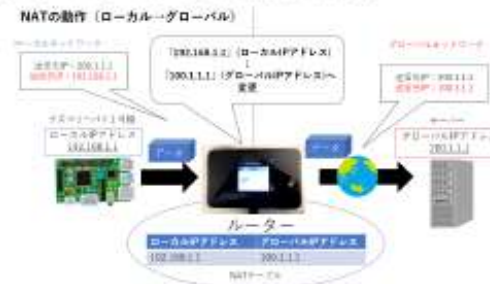


そこで、DNS サーバと言うものを作り、一度そこに "amazon.co.jp" のグローバル IP アドレスを覚えてもらってサーバに接続しています。



コラム NAT について

教科書の前のページでローカル IP アドレスとグローバル IP アドレスが存在することを皆さんは知りましたね。ここではもっと細かく IP アドレスを用いたインターネット接続を勉強しましょう。インターネットに接続するコンピュータは自分自身を示す IP アドレスとして世界で唯一のグローバル IP アドレスを使わなければならないのです。みなさんのラズベリーパイにはローカル IP アドレスしか割り当てられていませんので、このままではインターネットをすることはできません。そこで NAT (Network Address Translation) というものを用います。この NAT はルーターが行っていて、送信元のローカル IP アドレスをグローバル IP アドレスに変換して通信を行っています。具体的にはどのようにしているのか下の図をみて確認してみましょう。



ラズベリーパイ1号機からサーバにデータの送信先 IP アドレスは 200.1.1.1 (グローバル IP アドレス)、送信元 IP アドレスは 192.168.1.1 (ローカル IP アドレス) です。NAT を行うルーターは、プライベートアドレスとグローバルアドレスの境界に位置しています。ここでルーターは、ラズベリーパイから送信されたデータの送信元 IP アドレス 192.168.1.1 を、グローバル IP アドレスの「100.1.1.1」に変換して転送します。このときルーターは、その変換情報を NAT テーブルに記録しています。この記録が、「返ってくるデータ」、つまりサーバからラズベリーパイへ送られるデータの転送時に役立ちます。次の図を見てください。



Appendix: Lesson 7 Part of Textbook: WEB server

```
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi:/$ cd /home/pi/ome/07/www/
pi@raspberrypi:~/ome/07/www $ ./webserver.py
```

3. サーバーが動いているかどうかを確認してみましょう。

別のターミナルを立ち上げ、次のコマンドを打ちましょう。

```
nmmap localhost
```

今回使っているポート番号は:3000です。STATE (状態) が open (開く) になっています。

これでサーバーが動いていることが分かります。

```
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi:~$ nmap localhost
Starting Nmap 7.40 ( https://nmap.org ) at 2019-09-26 00:54 JST
Nmap scan report for localhost [127.0.0.1]
Host is up (0.00065s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
3000/tcp   open  pop
3900/tcp   open  vnc
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
pi@raspberrypi:~$
```

4. みなさんが作ったウェブページを表示してみましょう。

みなさんが作った index.html の場所は webserver.py を起動したディレクトリと同じでしたね。「2」で説明したドキュメントルートです。

これは webserver.py を起動したディレクトリ

2-3 例題 7-6 友だちのホームページを見てみよう



1. 左隣の友達 IP アドレスを確認しよう。

友達のターミナルで次のコマンドをうってもらいましょう。

```
hostname -I
```

その IP アドレスをメモしましょう。

[_____ さんの IP: _____]

2. 友達のウェブページを見て見ましょう。

自分のサーバーを見るときは

```
http://localhost:3000/
```

でした。localhost というのが自分のサーバーという意味でした。

今回は友達のサーバーを見たいので localhost の代わりに友達の IP アドレスを打ちましょう。

```
http://xxx.xxx.xxx.xxx:3000/
```

```
http://xxx.xxx.xxx.xxx:3000/index.html
```

をブラウザのアドレスバーに打ってみましょう。

もし左隣の友達の IP アドレスが 192.162.33.5 だったら

```
http://192.162.33.5:3000/index.html
```

となります。

友達のウェブページが表示されたら成功です!

Appendix: Lesson 7 Part of Textbook: Query String

例題 7-8 URL を使って CGI に情報を渡す

考え方

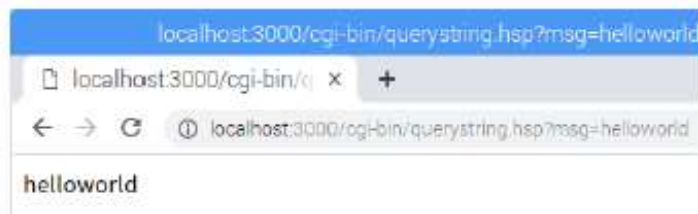
CGI のプログラムに情報を渡すことができればより便利なプログラムが作れます。ここでは単純な URL を用いた方法を使います。URL に埋め込まれた情報はクエリストリングと呼ばれます。クエリストリングはまずウェブサーバへ送られ、ウェブサーバが CGI プログラムへクエリストリングを渡します。これにより CGI プログラムは URL に埋め込まれた情報を渡うことができます。

まずは、ブラウザを開いて、

`localhost:3000/cgi-bin/querystring.hsp?msg=helloworld`

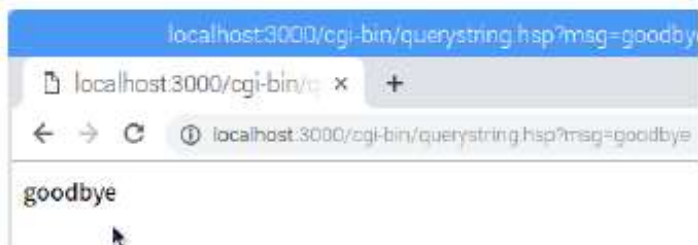
を入力してみましょう。

画面に `helloworld` と表示されたと思います。



`localhost:3000/cgi-bin/querystring.hsp?msg=goodbye`

に変更してみてください。



次は `goodbye` と表示されたと思います。

2019年9月29日

子どもIT未来塾 第7回

Page 46 of 66

例題 7-9 クエリストリングを使って LED を操作する

この例では、クエリストリングを使って LED を点灯消灯させてみましょう。

まずは、ブラウザを開いて、

`localhost:3000/cgi-bin/qsled.hsp?led=17&val=1`

を入力してみましょう。



実行すると LED17 が光ります。 `val=1` を `val=0` にすると消えます。URL に埋め込まれた情報をもとに CGI プログラムは LED を操作していることがわかります。URL に埋め込まれた情報はクエリストリングと呼ばれます。

クエリストリングを詳しく見てみましょう。

クエリストリングとは URL の最後に `?` をつけて始めます。 `?` のあとには

名前(変数名みたいなもの) = 値

の形を続けます。例えば、

`localhost:3000/cgi-bin/qsled.hsp?led=17`

のようにかけます。この場合は、

名前が `led`、値が `17` となります。

複数の情報を渡したい場合は `&` を使います。

`localhost:3000/cgi-bin/qsled.hsp?led=17&val=1`

のようになります。

名前1が `led`、値1が `17` となり

名前2が `val`、値2が `1` となります。

このクエリストリングをプログラムから扱う方法を次に学びましょう。

2019年9月29日

子どもIT未来塾 第7回

Page 49 of 66

Appendix: Lesson 8 Part of Textbook: HTML Fetch with curl

ウェブページから情報を取り出す(準備)

ウェブページから自分の必要な情報のみ取り出す方法を学びます。まずはプログラムから自動で情報を取り出すのではなく、スクレイピングの仕組みを理解するために、一つずつ手動で行います。

手順

0. グループの席を円と考えましょう。左となりの友達を確認します



1. 左となりの人の自己紹介ページをダウンロードします
2. ダウンロードしたページを調べてみましょう。

練習として、左となりの友達のウェブページから情報を取り出します。

そのための準備をしましょう。

1 冊目にした自己紹介ページを index.html としてコピーします

```
cp ~/ome/01/self_intro.html ~/ome/08/www/index.html
```

```
pi@raspberrypi:~$ cp ~/ome/01/self_intro.html ~/ome/08/www/index.html
pi@raspberrypi:~$
```

例題 8-1 コマンドラインからウェブページをダウンロードする

インターネットから情報を取ってくる際に使用するツールは curl (カール) と言います。curl はデータを送ったり受け取ったりするときに使います。

左となりの友達のウェブページを取ってきましょう。

保存するファイル名はローマ字で「友達の名前.html」にしましょう。

例: koyama.html

例題では friend.html として保存しています。

ターミナルを開いてください。

```
cd ~/ome/08
```

```
curl "左となりの友達の IP アドレス":3000/index.html -o friend.html
```

例: IP アドレスが 192.168.1.90 の友達の IP アドレス

```
→ curl 192.168.1.90:3000/index.html -o friend.html
```

を実行します。

```
pi@raspberrypi:~/ome/08
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
pi@raspberrypi:~$ cd ~/ome/08
pi@raspberrypi:~/ome/08$ curl 192.168.1.90:3000/index.html -o friend.html
% Total % Received % Xferd Average Speed Time Time Time Current
 0load  Upload Total Spent Left Speed
100 1720 100 1720 0 0 102k 0 0 0 0 0 0 0 0 0 0 112k
pi@raspberrypi:~/ome/08$
```

HTML ファイルをダウンロードして friend.html として保存しています。

コラム curl コマンドのオプション(機能)

curl コマンドを何もオプションをつけずに実行すると、ターミナルにダウンロードした情報を表示します。コマンドに "-o" オプションをつけることで、ターミナルに表示する代わりに、ファイルに保存することができます。

"-o" の後に保存するファイル名を指定します。

```
curl URL -o ファイル名
```

Appendix: Lesson 8 Part of Textbook: WEB Scraping

例題 8-7 アメダスのウェブページをスクレイピングする

考え方

まずは、ウェブブラウザでアメダスのページを見てみよう。検索で使ったホームページを開いてください。(/ome/08/links.html)

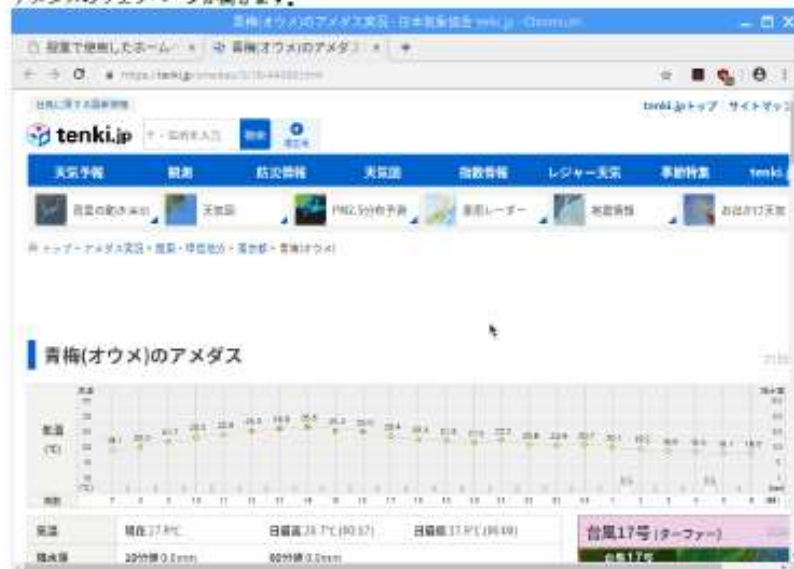
2.アメダスをダブルクリックします。



授業で使ったホームページ

1. ASGIコード書
2. アメダス
3. Python
4. Wotk
5. 気象情報
6. 気象コードを調べる
7. 気象データから気象を調べる
8. Wikipedia

アメダスのウェブページが開きます。



プログラム解説

```
1. #include "hsp3cl.h"
2. #include "cmdexec.h"
3.
4. cmdexec "amedas.py", result
5. split result, " ", jikan, kion, kouzui, kazamuki, fusoku, hizyou, sekisetsu
6.
7. ; 結果を出力
8. mez "時刻" + "気温" + "降水"
9. mez jikan + " " + kion + " " + kouzui
10.; プログラムの終了
11.end
```

この例題プログラムでは、amedas.py というコマンドを使って、ウェブページからアメダスの情報を取得しています。

4行目でHSPプログラム内から amedas.py コマンドを実行しています。

cmdexec 命令は、exec と基本的には同じですが、実行するコマンドの後に変数名をつけます。この変数にはコマンドの実行結果が文字列で入ります。

一度、amedas.py コマンドをターミナルで実行してみましょう。ターミナルを開きます。

amedas.py を実行します。

```
pi@raspberrypi:~/ome/08 $ amedas.py
21日 06:30, 27.8, 0.0, 北西, 0.4, 0.0, ---, pi@raspberrypi:~/ome/08 $
```

カンマ (,) で区切られて値が表示されています。

これらの値がなにを意味しているのか見てみましょう。

amedas.py --print-headers

を実行します。

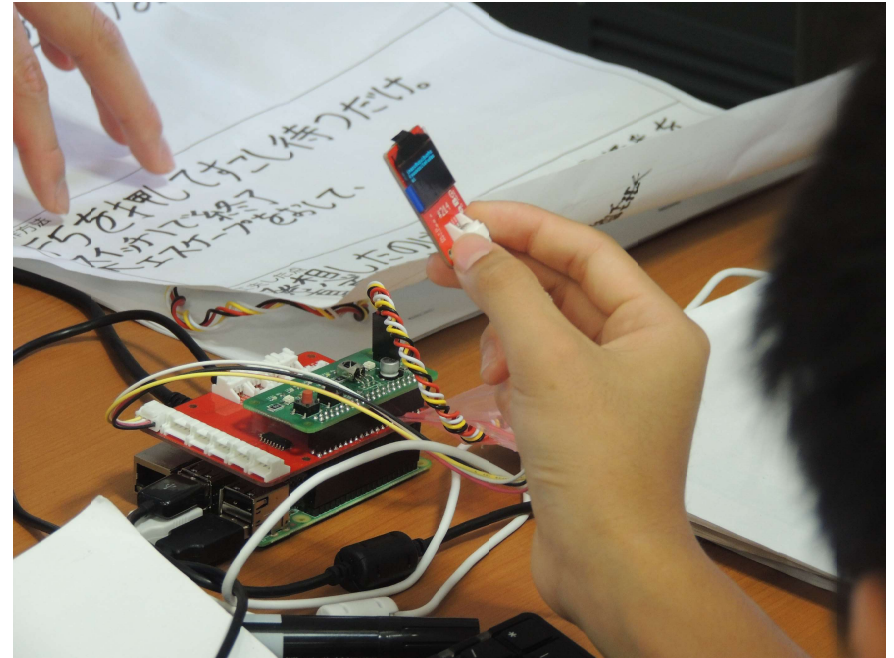
```
pi@raspberrypi:~/ome/08 $ amedas.py --print-headers
青梅市のアメダスの記録(10分間隔値)
日時: 21日 06:30
気温(°C): 27.8
降水量(mm): 0.0
風速(m/s): 0.4
風向(方位): 北西
日照時間(分): 0
積雪(cm): ---
```

日時、気温、降水量、風向、風速、日照時間、積雪深の順番で値と値の見出しが表示されています。

このカンマ (,) 区切りの数値も同じ並びで表示されています。

```
pi@raspberrypi:~/ome/08 $ amedas.py
21日 06:30, 27.8, 0.0, 北西, 0.4, 0.0, ---, pi@raspberrypi:~/ome/08 $
```

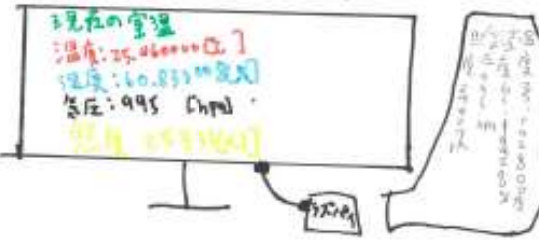

Appendix: Presentation, Demonstration, Finale



Naohiko Shimizu

Development of IoT Curricula for Children (Age 10 to 13)

Appendix: Lesson 10 Presentation

題名 しゃべる温度計	
グループ A	No. 3 名前 海藤 圭介
アピールしたい点 ラズベリーパイのセンサーボードで言った温度のことなどを音声でひょうじすることで、温度をかくとしゃべりなる。 	
操作方法 FSをおすだけ。(おしたら音声か流れる→情報がこらえられるのくらいがえし) LEDがつかうと、 ↑ 音がまじいので注意	
工夫した点 わけが時間をおろして ききやすした。 色を付けた。 音を付けた。	感想 無事終わらせる(完成) ことができた。

題名 百人一首ゲーム (vs julius) <small>初級及編</small>	
グループ E	No. 4 名前 松本 叶落
アピールしたい点 juliusと人がカレタで対戦できる。	
製作方法 読み手と対戦プレイヤー 1人ずつ必要。 読み手が読んだ句を julius が認識するのが先か、対戦プレイヤーが札をとるのが先かを競う。 今回は5枚のみなので、プレイヤーが3枚とるとプレイヤーが勝利。	
工夫した点 札をつくった。	感想 読みファイルにマイヒョウクするのが大変だった。