# STORYTELLER

kobayashikorio@gmail.com
2017/8/29

Storyteller shows long Japanese document on small display which has 128*64 dots. This program works on RaspberryPi Zero and OLE display driven by SSD1306 where display dots layout is 128*64=(16*8)*(8*8) means 16*8 Kanji characters can be shown simultaneously. To abort the display process, set pin7 to GND.

Objectives of this projects are:
1. To evaluate small display controlled by the Mathematica through i2c using Systm shell
2. To evaluate Japanese font ability and text management
3. To evaluate a GPIO interrupt switch
4. To evaluate a packaging method adopting 3D pen

Associating files
The program needs 2 files, Mathematica association style Kanji bitmap file and source text encoded by unicode file. Kanji bitmap is made by the Misaki font (8*8 very small faced Kanji font covers JIS 1st and 2nd level).

Hardware connection
RaspberryPi Zero and SSD1306 are connected with i2C. Mathematica drives SSD1306 through i2cset utility with the RunProcess/Writeline function.

OLE to RaspberryPi hardware pin connection
SDA.-> Pin3
SCL -> Pin5
VCC -> Pin1(3.3V)
GND -> Pin6

Interrupt switch connection (pin7 is initially pull-upped)
Pin7 - tactswitch - GND

Standalone setting
To start the program automatically after the booting process, all code should be written in text and add a following line at the begnning,

```
#! /opt/Wolfram/WolframEngine/10.3/Executables/math -script
```

then put this file to directory /home/pi, chmod +x the script file, and add the file name to rc.local. When code maintenace is required, set pin7 to GND and the booting process to escape shutdown.

Reference( in Japanese )
1. Kanji bitmap was constructed from "misaki_mincho.bdf"(X11 BDF fornat) and JIS code Kanji1, Kanji2 table, because misaki_mincho has JIS code but RapberryPi uses Unicode.
http://www.geocities.jp/littlimi/misaki.htm
http://ash.jp/code/unitbl21.htm
http://ash.jp/code/unitbl22.htm
2. GPIO pins initial state
http://d.hatena.ne.jp/hnw/20150607
3. SSD1306 data sheet
https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf
4. Ilias of Aozora Bunko
Text was downloaded and edited

http://www.aozora.gr.jp/cards/001099/card46996.html

## Open shell process for the interrupt

```
(* set interrupt service to GPIO Pin7 *)
process = StartProcess[$SystemShell];
WriteLine[process, "echo '7' > /sys/class/gpio/export"];
WriteLine[process, "cat /sys/class/gpio/gpio7/value"];
(* interrupt escape line *)
If[ReadLine[process] == "0", Exit[]];
```

## Kanji bitmap  ~association Misaki font and Unicode~

```
SetDirectory[NotebookDirectory[]];
kanji = Get["misakiFont.txt"];
```

## Sorce text to display ~ Ilias ~

```
fileStr = Import["ilias.txt"];
txt = StringSplit[fileStr, "\n"];
```

## Display set up

```
(* kanji face transformer *)
kanji2bp[face_] := (
  rbp = kanji[face];
  r2 = Transpose[Map[IntegerDigits[FromDigits[#, 16], 2, 8] &, rbp]];
  r3 = Map[FromDigits[#, 2] &, r2];
  r4 = IntegerString[r3, 16] /. "0" → "00";
  Map[StringJoin["0x", #] &, r4]
 )
(* SSD1306 i2c template *)
comStr = {"i2cset", "-y", "1", "0x3c", "0x00"};
dataStr = {"i2cset", "-y", "1", "0x3c", "0x40"};
rSet[reg_] := RunProcess[Join[comStr, {reg}]];
rDSet[reg_, adrm_] := RunProcess[Join[comStr, {reg}, {adrm}, {"0x00"}, {"i"}]];
rTSet[reg_, p1_, p2_] := RunProcess[Join[comStr, {reg}, {p1}, {p2}, {"0x00"}, {"i"}]];
dSet[bp_] := RunProcess[Join[dataStr, bp, {"i"}]];
(* page addressing commands *)
pagePos = 1;
page[1] = "0xb7"; page[2] = "0xb6"; page[3] = "0xb5"; page[4] = "0xb4";
page[5] = "0xb3"; page[6] = "0xb2"; page[7] = "0xb1"; page[8] = "0xb0";
```

```
(* display eraser *)
clear[] := (
   rSet["0x02"];
   rSet["0x10"];
   clrstr = "i2cset -y 1 0x3c 0x40 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 i";
   (* StartProcess/WriteLine mehod is significanly faster than RunProcess *)
   Do[WriteLine[process, clrstr], {4}];
  );
clearDisplay[] := Table[
   rSet[page[i]]; clear[];, {i, 8}];
(* increment page pointer and clear page for kanji drawing *)
drawline[str_ /; str == {}] := (
   rSet[page[Mod[pagePos++, 8, 1]]];
   clear[];
  );
(* kanji drawing *)
drawline[str_] := (
   (* interrupt execution and shutdown RaspberryPi *)
   WriteLine[process, "cat /sys/class/gpio/gpio7/value"];
   If[ReadLine[process] == "0",
    clearDisplay[];
    Pause[0.5]; $Epilog := WriteLine[process, "sudo shutdown -h now"]; Exit[]];
   (* end of interrupt signal  pickup *)
   rSet["0x02"];
   rSet["0x10"];
   bb = Map[kanji2bp[#] &, str];
   dcom = Map[Join[{"i2cset", "-y", "1", "0x3c", "0x40"}, #, {"i"}] &, bb];
   Map[RunProcess, dcom];
   rSet[page[Mod[pagePos++, 8, 1]]];
   clear[];
  );
(* nibble 16 kanji chatacters *)
nibl[str_ /; Length[str] ≤ 16] := drawline[str];
nibl[str_ /; Length[str] > 16] := (
   drawline[Take[str, 16]];
   nibl[Drop[str, 16]];
  );
```

## Text auto incremental viewer

```
(* initialize display *)
clearDisplay[];
rSet[page[1]];
(* power-on display *)
rSet["0x8d"];(*charge pump on*)
rSet["0x14"];(*enable charge pump*)
rSet["0xaf"];(*display on in normal mode*)
rSet["0xa1"];(*set segment remap to reverse *)


(* show text line *)
line = 1;
Do[dc = Characters[txt[[line++]]];
 nibl[dc];, {Length[txt]}]
(* end of story telling then shutdown *)
clearDisplay[];
Pause[0.5];(*$Epilog:=WriteLine[process,"sudo shutdown -h now"];Exit[];*)


$Aborted
```