
RollingWord

kobayashikorio@gmail.com
2017/9/17

RollingWord project is intended to evaluate following challenge.

1. display rotation mechanically
2. manipulate a servo-motor with RaspberryPi PWM function
3. evaluate power condition is safe for simultaneous usage of a OLE display and a servo-motor
4. 16 dot-height large character usage for SSD1306
5. preparing suitable bit-map font
6. manipulate large word dictionary
7. construction system package with moving parts using 3D pen

hardware connection

1. RaspberryPi to SSD1306-OLED
 - 1:3.3V -> Vcc
 - 3:SDA -> SDA
 - 5:SCL -> SCL
 - 6:GND -> GND
2. RaspberryPi to SG90 servo-motor
 - 2:5V -> Vcc (Red)
 - 9:GND -> GND (Brown)
 - 12:GPIO18 -> SG90 PWM (Orange)
3. shutdown tact switch
 - GPIO7 -> switch
 - GND -> switch

SG90 servo-motor setting

SG90 specification is defined the pulse width as follows where 1-cycle length is 20 ms because 50 Hz pulse series is used.

position 0 deg: 1.45 ms
position 90 deg: 2.4 ms
position -90 deg: 0.5 ms

To set pin output to 50 Hz for SG90, where system clock is 19.2 MHz, should set COUNTER is 192, and RESOLUTION is 2000 as, $19.2 \times 10^6 / 192 / 2000 \rightarrow 50$, hereafter the duty cycle is set by the equation,
 $\text{DutyCycle} = N / \text{RESOLUTION}$.

So, command parameter N of "gpio pwm 1 N" becomes as followings.

0 deg: $1.45 / 20 \rightarrow 145 / 2000 \rightarrow$ "gpio pwm 1 145"
90 deg: $2.4 / 20 \rightarrow 240 / 2000 \rightarrow$ "gpio pwm 1 240"
-90 deg: $0.5 / 20 \rightarrow 50 / 2000 \rightarrow$ "gpio pwm 1 50"

References

1. <http://www.denshi.club/make/2016/02/raspberry-pi2-3.html>
2. <https://raspberrypi.stackexchange.com/questions/4906/control-hardware-pwm-frequencyz>
3. <http://kutajin.blog101.fc2.com/blog-entry-279.html>
4. <https://projects.drogon.net/raspberry-pi/wiringpi/the-gpio-utility/>
5. <https://raspberrypi.stackexchange.com/questions/4906/control-hardware-pwm-frequency/9725#9725>

```

SetDirectory[NotebookDirectory[]];
fontTable = < | | >;
AssociateTo[fontTable, Get["rules.txt"]];
words = Get["words.txt"];
process = StartProcess[$SystemShell];
(*set interrupt service to GPIO Pin7*)
WriteLine[process, "echo '7' > /sys/class/gpio/export"];
(* setup SG90 servo motor *)
WriteLine[process, "gpio mode 1 pwm"];
WriteLine[process, "gpio pwm-ms"];
WriteLine[process, "gpio pwmrc 192"]; (* set counter *)
WriteLine[process, "gpio pwmr 2000"]; (* set resolution *)
WriteLine[process, "gpio pwm 1 60"]; (* set to home position *)
clock :=
  Do[WriteLine[process, "gpio pwm 1 " <> ToString[n]]; Pause[0.05], {n, 250, 60, -2}];
unlock := Do[WriteLine[process, "gpio pwm 1 " <> ToString[n]];
  Pause[0.05], {n, 60, 250, 2}];

```

Setup for SSD1306 and output data lines

```

chipAdr = "0x3c";
(* set blank code considering i2cset command for data has a limit length *)
blank = ConstantArray["0x00", 32];
blankStr = StringRiffle[Join[{"i2cset", "-y", "1", "0x3c", "0x40"}, blank, {"i"}]];
(* register set function *)
rSet[r_] := WriteLine[process, StringRiffle[{"i2cset -y 1", chipAdr, "0x00", r}]];
(* page set function *)
setPage[r_] := (
  rSet["0xb" <> ToString[r]]; (* set page address *)
  rSet["0x02"]; rSet["0x10"] (* column reset to 1 with lower & higher nibble *));
clearPage[r_] := (
  setPage[r];
  Table[WriteLine[process, blankStr], {4}];
  (* one page is divided by 4 partitions *)
  setPage[r]); (* reset column to 1 *)
(* output string *)
putHex[page_, hex_] := (
  clearPage[page];
  dcom =
    Map[StringRiffle[Join[{"i2cset", "-y", "1", "0x3c", "0x40"}, #, {"i"}]] &, hex];
  Map[WriteLine[process, #] &, dcom]);
(* display set up *)
Table[clearPage[i], {i, 7, 0, -1}];
rSet["0x8d"]; (* charge pump on *)
rSet["0x14"]; (* enable charge pump *)
rSet["0xaf"]; (* display on in normal mode *)
rSet["0xa1"]; (* set segment remap to reverse *);

```

```

While[True,
  (*interrupt execution and shutdown RaspberryPi*)
  WriteLine[process, "cat /sys/class/gpio/gpio7/value"];
  If[ReadLine[process] == "0",
    WriteLine[process, "gpio pwm 1 60"]; (* back to home position *)
    Pause[0.5];
    $Epilog := WriteLine[process, "sudo shutdown -h now"];
    Exit[]];
  (*end of interrupt signal pickup*)
  cstr = Characters[StringJoin["", RandomChoice[words]]];
  up = Map[fontTable[#][[1]] &, cstr];
  down = Map[fontTable[#][[2]] &, cstr];
  putHex[5, up];
  putHex[4, down];
  Pause[3.0];
  unlock;
  Pause[1.0];
  clock];
$Aborted
{unlock, Pause[0.8], clock};

```