トイレビアン**Project**

**１．目的**

**コンビニのトイレは沢山の問題を抱えている**

1. **長時間利用**
2. **トイレの破損、排水の詰まり**
3. **汚される**

**監視カメラではトイレの利用者からクレームとなるため設置できない、現状は無防備な状態になっている**

**それを改善できる手段をオーナーは待ち望んでいる**

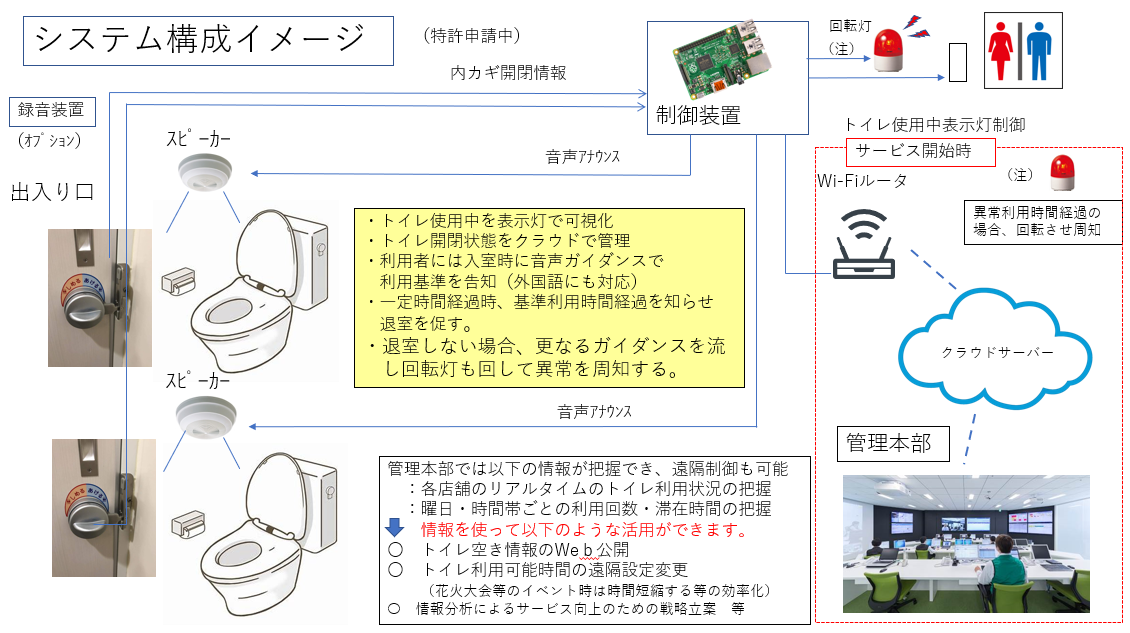
**その期待に応えるのが「トイレビアン」である**

**README**

This project is used for to know how many gentleman and lady use toilet for one day and one hour and average minute for one day and one hour with graph. That data we see in graph. And the light sign when the user using toilet .And then when the toilet user use 3minute announce occur and next 6 minute different announce occur and next 9 minute announce occur and next 12 minute also. We use Raspberry Pi 3 Model B Rev 1.2 and Python 2.7.

このプロジェクトは、1日1時間にトイレを使用している紳士と婦人の数と、1日と1時間の平均分をグラフで把握するために使用されます。グラフに表示されるデータ。そして、トイレを使用するユーザーがライトサインを表示します。トイレユーザーが3分間のアナウンスを使用すると、次の6分間のアナウンスが発生し、次の9分間のアナウンスが発生し、次の12分間もアナウンスします。 Raspberry Pi 3 Model B Rev 1.2とPython 2.7を使用しています。

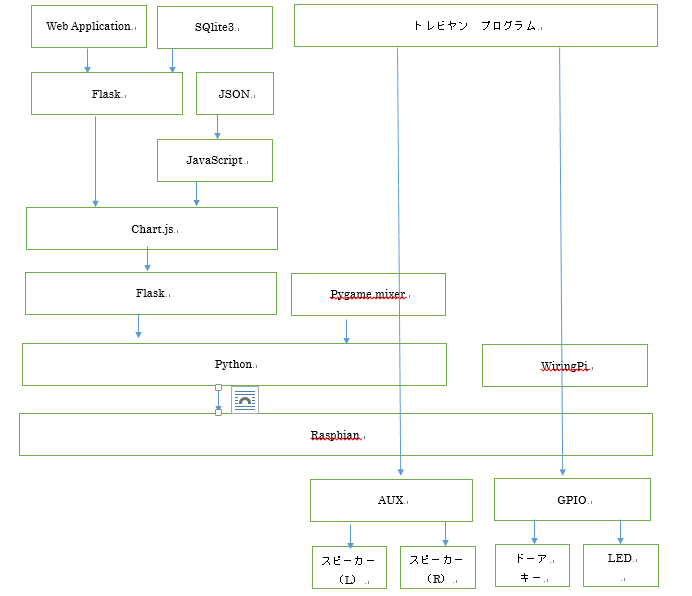
２．システム構成



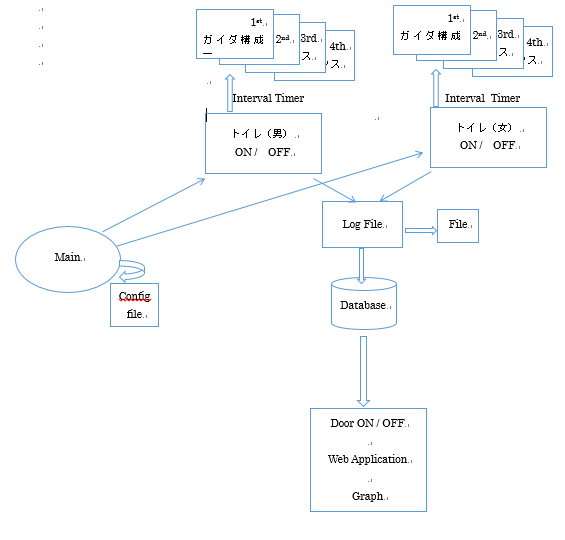
３．ハードウェアインタフェース

「BH3テクニカルノート」参照

４．ソフトウェアデザイン



４－１．ソフトウェア構成図



５．機能一覧

５－１．起動時

|  |  |
| --- | --- |
| セルフテスト | ・LEDを全て点灯して消灯  ・ウォッチドックタイマー有効 |
|  |  |
|  |  |
|  |  |
|  |  |

５－２．通常動作

|  |  |
| --- | --- |
| トイレのドアON/OFF確認 | 【OFF->ON】の場合  ・男性／女性で別々に検知する  ・入室ログをとる  ・ガイダンスのタイマーを起動  ・WEB画面で「使用中」にする  ・短時間でON->OFFとされた場合は退出とはみなさずタイマーは継続させる（時間はConfigファイルにて定義）  【ON->OFF】の場合 |
| ガイダンス | ・Configファイルで指定されているインターバル時間を指定  ・ガイダンスはNo1～No4までをタイマー起動  ・No4を再生後、リピート再生（間隔はConfigファイルにて定義） |
|  |  |
|  |  |

**Prerequisites**

* Wiringpi
* Time
* Datetime
* Threading
* Pygame.mixer
* Json
* ftp

**How to Install Flask**

sudo apt-get install python-virtualenv

sudo apt-get install python-virtualenv

mkdir simple\_flask (Making Folder)

mkdir template (creating folder)

mkdir LOG (creating folder)

cd simple\_flask

python2 -m virtualenv venv

. venv/bin/activate (Activate Virtual env)

sudo apt-get install python-pip

sudo pip install flask  (To Install flask,It can work python2.7 and newer)

**Used**

Wiringthread (Gentleman), SWITCH = GPIO 26 , 1( SWITCH is ON)

LED = GPIO 18, 0 (LED is ON)

Wiringthread 2(Lady), SWITCH = GPIO 17, 1 (SWITCH is ON)

LED = GPIO 18, 0 (LED is ON)

Database = sqlite3 ToData.db

Table = DATA

Column = id, kinds, date, time, mode, message, duration

home.py

wiringthread.py

wiringthread2.py

home.html

logTable.py

power.py

[shutdown.py](ftp://ftp.py)

ftp.py

**home.py**

- Call wiringthread and wiringthread2 simultaneously by using threading.

- The data updated in status file read and send data to home.html to light for boy button

- The data updated in status file read and send data to home.html to light for girl button;

- The data in database send to see graph for boy in web browser the data

- The data in database send to see graph for girl in web browser the data.

- To see what time raspberry is opening login our page’s log file and send each 5minutes.

- To see what time raspberry is shut down page’s shut down log each 5 minutes

- スレッディングを使用して、wireringthreadとwirethread2を同時に呼び出します。

- ステータスファイルで更新されたデータを読み取り、home.htmlにデータを送信して、男の子ボタンを点灯させます

- ステータスファイルで更新されたデータを読み取り、home.htmlにデータを送信して、女の子のボタンを点灯させます。

- データベース内のデータは、ウェブブラウザで男の子のグラフを表示するために送信されます

- データベース内のデータは、女の子のグラフをWebブラウザーで表示するために送信されます。

- ラズベリーが開いている時間を確認するには、ページのログファイルにログインし、5分ごとに送信します。

- ラズベリーがシャットダウンされる時間を確認するには、5分ごとにページのシャットダウング

**wiringthread.py**

* Switch = GPIO 26 , 1( SWITCH is ON)
* LED = GPIO 12, 0 (LED is ON)
* To Check switch is ON/OFF
* When switch is ON ,light is ON
* Save the number of people and Boy busy log in log.txt file and update the status in status.txt file
* When the time that is busy is over 3 minute the announce occur, after 6 minute other announce occur , after 9 minute the next and after 12 minutes the other occur.
* To store at database.
* When switch is OFF, light is OFF
* Save the boy free log in log.txt file and update the status in status.txt file
* After Time sleep 0.1
* Check again

- チェックスイッチがオン/オフになっている

* スイッチがオンのとき、ライトはオンです -log.txtファイルに人数とBoy busyログを保存し、status.txtファイルのステータスを更新します
* -ビジー状態の時間が3分を超えるとアナウンスが発生し、6分後に他のアナウンスが発生し、9分後に次のアナウンスが発生し、12分後に他のアナウンスが発生します。
* -データベースに保存します。
* -スイッチがオフの場合、ライトはオフです -Boy.freeログをlog.txtファイルに保存し、status.txtファイルのステータスを更新します –
* -アフタータイムスリープ0.1

**wiringthread2.py**

* Switch = GPIO 17,1 ( SWITCH is ON)
* LED = GPIO 18,0 (LED is ON)
* To Check switch is ON/OFF
* When switch is ON ,light is ON
* Save the number of people and Girl busy log in log.txt file and update the status in status.txt file
* When the time that is busy is over 3 minute the announce occur, after 6 minute other announce occur , after 9 minute the next and after 12 minutes the other occur.
* To store at database.
* When switch is OFF, light is OFF
* Save the girl free log in log.txt file and update the status in status.txt file
* After Time sleep 0.1
* Check again.

- チェックスイッチがオン/オフになっている

- スイッチがオンのとき、ライトはオンです

- 人数とビジー状態のログをlog.txtファイルに保存し、status.txtファイルのステータスを更新しす

- ビジー状態の時間が3分を超えるとアナウンスが発生し、6分後に他のアナウンスが発生し、9分後に次のアナウンスが発生し、12分後に他のアナウンスが発生します。

- データベースに保存します。

- スイッチがオフの場合、ライトはオフです

- 女の子無料のログをlog.txtファイルに保存し、status.txtファイルのステータスを更新します

- アフタータイムスリープ0.1

- 再び確かめる。

**home.html**

* To see Boy Busy button and Girl Busy button in web browser
* When the switch is ON for boy the Boy Busy Button is red and switch is OFF the Boy Free button is red.
* When the switch is ON for boy the Boy Busy Button is red and switch is OFF the Boy Free button is red.
* To see the graph for one day how many boy and girls use toilet.
* To see the graph for one hour how many boy and girls use toilet.
* To see the average minute graph for one day per number of people use toilet.

-　　　　WebブラウザーでBoy BusyボタンとGirl Busyボタンを表示する

- BoyのスイッチがONの場合、Boy Busyボタンは赤色で、スイッチがOFFの場合、Boy Freeボタンは赤色です。

- BoyのスイッチがONの場合、Boy Busyボタンは赤色で、スイッチがOFFの場合、Boy Freeボタンは赤色です。

- トイレを使用している男の子と女の子の数を1日のグラフで確認します。

- トイレを使用している少年少女の数を1時間のグラフで確認する。

- トイレを使用している人数あたりの1日の平均分のグラフを表示する。

**logTable.py**

* Using slqite3 database
* Create database ToData.db
* Create table with id, kinks, date, time, mode, message, duration.
* Inserting data to that table.
* For one day select the kinds and count the date where mode is busy
* For one hour select the hour and count the hour
* Update user and duration

- slqite3データベースの使用

- データベースToData.dbを作成する

- ID、ねじれ、日付、時刻、モード、メッセージ、期間を含むテーブルを作成します。

- そのテーブルにデータを挿入します。

- 一日のために種類を選択し、モードが忙しい日付を数えます

- 1時間のために時間を選択し、時間を数えます

- ユーザーと期間を更新する

[**ftp.py**](ftp://ftp.py)

* Use ftp library, os, fileinput
* Write the file that want to sent
* Write output\_directory to logfile
* ftp connect to server
* And login with username and password
* FTP is used for file transfer from log file to server

- FTPライブラリ、OS、ファイル入力を使用する

- 送信したいファイルを書きます

- output\_directoryをログファイルに書き込む

- FTPでサーバーに接続

- そして、ユーザー名とパスワードでログイン

- FTPはログファイルからサーバーへのファイル転送に使用されます

**power.py**

* Import time
* Determine the current time
* Write the current time at powerlog text file

- インポート時間

- 現在の時刻を特定する

- 現在時刻をpowerlogテキストファイルに書き込む

**shutdown.py**

* Import time and datetime
* Determine asctime
* Write the current time at shutdown log text file

- インポート時刻と日付時刻

- asctimeを決定する

- シャットダウン時の現在の時刻をログテキストファイルに書き

wiringthread

wiringthread2

Switch2

Switch1

0.1 sec 0.1 sec

Off Off

On On

home.py

logTable.py

home.html

Database

DATA

Web Browse

Button see, Graph see

ftp.py

shutdown

power

Log2

Log1