Vol.2019-IOT-47 No.10
Vol.2019-SPT-35 No.10
2019/9/20

簡便な大型幅広デジタルサイネージシステムとその自動運用システム

山之上　卓[†], 田中 始男[†], 原 裕樹[†], 中川 裕貴[†]
矢嶋 辰伍[†], 増野 治哉[†], 森重 太智[†]

概要：簡便な大型幅広デジタルサイネージシステムとその自動運用システムについて述べる．このデジタルサイネージシステムはビルの多くの窓に障子紙を貼り，窓の部屋側から多数のプロジェクタで多数の画像を投影し，全体として大きな１枚の画像を構成して表示するものである．自動運用システムはこのサイネージシステムの多数のプロジェクタを，自動的に，予め設定された時刻において一斉に、点灯・消灯したり，管理者によって強制的に点灯・消灯したりするものである．この時刻設定や強制的な点灯・消灯は外部の遠隔地から行うことができるため，ビルに人がいない夜間でも臨機応変にプロジェクタの点灯・消灯時間を変更したり，強制的に点灯・消灯したりすることができる．

キーワード：Wiki, Bot, IoT, P2P, Digital Signage

A Casual Big and Wide Digital Signage System and its Automatic Operation System

Takashi YAMANOUE[†]

Abstract: This paper discusses a casual, big and wide digital signage system and its automatic operation system. The image of the signage is partitioned into several parts of the image and projected to the screen using several projectors. The screen is realized by windows of a building which are putted “Shoji”, paper screens, on. The automatic operation system is used for turning on and off the several projectors at times of previously set. The system also can turn on and off all of the projectors forcibly by the administrator. The setting and operations can be performed from remote places. So, it is possible to perform such operations at night when no one in the building, flexibly.

Keywords: Wiki, Bot, IoT, P2P, Digital Signage

1. はじめに

人通りが多い場所で多くの人々に情報を伝える手段の一つとして，デジタルサイネージ[1]が良く利用されている．デジタルサイネージは内容の更新が容易であり，動画の表示も可能な場合もある．

夜間の建物に映像を投影する方法としてプロジェクションマッピングが良く使われている．プロジェクションマッピング[2]は建物の凹凸も考慮して映像を投影することにより大きな映像を効果的に投影することが可能である．しかしながら、これは建物の外部から投影することが必要となる．繁華街で建物の外部にプロジェクタ設置場所を確保することは，外部との交渉や費用が必要となり，気軽には実施できない．大きな画像の投影を行うにはそれなりの輝度を持ったプロジェクタが必要となり．この部分でも大きな費用が必要となる．

デジタルサイネージシステムとして大型 LED 表示装置も良く使われている．これは昼夜問わず表示が可能であり，プロジェクションマッピングのときのようなプロジェクタ設置場所の確保の問題もない．しかしながら窓をふさぐような場所には設置が難しく，また設置工事が必要となり，気軽に設置できない．

プロジェクションマッピングや大型 LED 表示装置と比べて，安価に気軽に設置可能なデジタルサイネージとして，建物の室内から窓に映像を投影し，窓全体に光を通すスクリーンを張ることで窓全体を使う「大型窓サイネージ」システムを開発している．ここで，横長の大きな窓全体に、明るい映像を投影するため，複数のプロジェクタを使う方法を採用した．

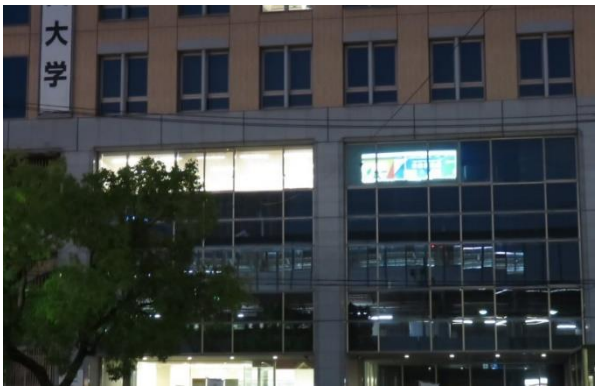

図 1. 試験的な「大型窓サイネージ」の実装

† 福山大学
Fukuyama University

Vol.2019-IOT-47 No.10
Vol.2019-SPT-35 No.10
2019/9/20

表示サイズを大きくするには，大型窓サイネージシステムで使用するプロジェクタとパソコンの数を増やし，設定変更するだけで実現できる．本システムは、一秒当たりの表示画面数が 10 枚程度になる制限があるが，動画も表示できる．試験的に、二台のプロジェクタを使って横長の画面表示をおこなった(図 1)が，画面に変化があったとき，２つの画面間の同期についても，概ね、問題なく表示がおこなわれていた．

ビルの窓の内側から夜間にデジタルサイネージのコンテンツを投影する場合，表示したい時間にプロジェクタを点灯し，表示する時間が終了するときにプロジェクタを消灯する必要がある．適切な時間にプロジェクタを点灯しないと，コンテンツがサイネージ近くを通過する人々の目に映る時間が減少する．また，適切な時間にプロジェクタを消灯しないと街の人々に迷惑をかける可能性が生じ，誰もコンテンツを見てくれない時間に点灯することによる無駄な電気料金も必要になる．また長時間プロジェクタを点灯したままにするとプロジェクタの障害発生の可能性も高くなる．

プロジェクタの電源の On/Off は，市販のタイマーと，プロジェクタが備えている内臓タイマーを組み合わせて，決まった時間に行うことが可能である．しかしながら，この方法では，日によって表示時間を長くしたり短くしたりすることをプロジェクタの設定変更なしに行うことはできない．また，市販のタイマーの設定時間を変更するためには，人間がそのタイマーの場所に行って直接操作することが必要になる．デジタルサイネージが実際に運用される夜間はサイネージが設置されるビルが施錠され，人間が直接操作することが不可能になる場合があるので，映像が表示されている夜間に，臨機応変にプロジェクタの表示時間を変更可能にするためには，プロジェクタの内臓タイマーや市販のタイマーを利用することは不適切である．

本論文の「自動運用システム」はこのサイネージシステムの多数のプロジェクタを，自動的に，Web の上で予め設定された時刻において一斉に、点灯・消灯したり，管理者によって強制的に点灯･消灯したりするものである．この時刻設定や強制的な点灯・消灯は外部の遠隔地から行うことができるため，ビルに人がいない夜間でも臨機応変にプロジェクタの点灯･消灯時間を変更したり，強制的に点灯・消灯したりすることができる．

2. ポータブルクラウド

「大型窓サイネージ」システムのスクリーンに表示される画像は，画面送信用ノートパソコンの画面にいったん表示され，その画像はそのままプロジェクタに接続された画面受信用ノートパソコンそれぞれに送信される．それぞれの画面受信用ノートパソコンでは受信した画面のうち，対応するプロジェクタが表示する部分のみを切り出して表示される．

画面送信用ノートパソコンでインターネット場の Web ページを表示したり，画面送信用ノートパソコンの画面を複数の画面受信用ノートパソコンに送信したりするため，ポータブルクラウドの「分散 Web スクリーンシェア(DWSS)」を使っている．ポータブルクラウド[3] [4][5]は持ち運び可能な容器の中に小型ネットワークスイッチ，複数台の小型パソコン，高性能無線 LAN アクセスポイント，電源装置などを格納し，一般的な教室で BYOD 端末を利用した授業や会議を行うためのソフトウェアを導入したものである．

ポータブルクラウドは無線 LAN アクセスポイントも持っているので，ポータブルクラウドの外の環境はあまり気にせずに，ポータブルクラウドの機能を使うことができる．Web で使われている通信プロトコルの TCP の場合，遅延が長いことが単位時間あたりの通信容量低下に結びつくので，近くにサーバがあることは，どこにサーバがあるかわからない一般的なクラウドと比べて，通信容量の面でも有利になる．

図 2 にポータブルクラウドの概要を示す．従来，クラウドは，その中にある様々なサービスを実現するサーバが，利用する場所の外の，どこにあるかわからない所にあったのに対して，ポータブルクラウドでは，良く利用するサーバ類を，利用する場所の近くに置くことになる．
ポータブルクラウドは，その利用者間で画面共有を可能にする機能を持つ[6][7][8].

パソコンをポータブルクラウドに接続するとメニューが表示され、そこで “WebScreenShare”のボタンを選んでクリックすると，その利用者が使っているブラウザの画面が，他者が送信するデスクトップ画面の表示する画面（受信画面）になる．

受信画面のリンクをクリックして，画面送信に必要なプログラムをダウンロードして実行することにより，そのプログラムを実行しているパソコンのデスクトップ画面が，他の利用者の受信画面に，繰り返し，送信される．その結果，それぞれのブラウザの上で，送信者のデスクトップ画面が実時間で表示される．このとき，画面受信側は HTML5[9]が利用できる一般的な Web ブラウザがあれば，特にソフトウェアのインストール等は必要ない．送信側は，Java が使えれば画面を送信できる．

3. 「大型窓サイネージ」システム

図 3 に大型窓サイネージシステムの概要を示す．窓サイネージシステムは，ポータブルクラウド，複数のノートパソコン，複数のプロジェクタ，スクリーン，自動運用システムから構成される．

ポータブルクラウドのルータは，インターネットに接続される．

ノートパソコンは，画面送信用に利用するものと画面受信用に利用するものの２種類があり，それぞれ、ポータブルクラウドに，有線 LAN または無線 LAN で接続される．画面受信用のノートパソコンの台数は必要に応じて増やすことができる．図 2 では画面受信用ノートパソコンの台数は 2 である．

プロジェクタは画面受信用ノートパソコンに接続される．

Vol.2019-IOT-47 No.10
Vol.2019-SPT-35 No.10
2019/9/20

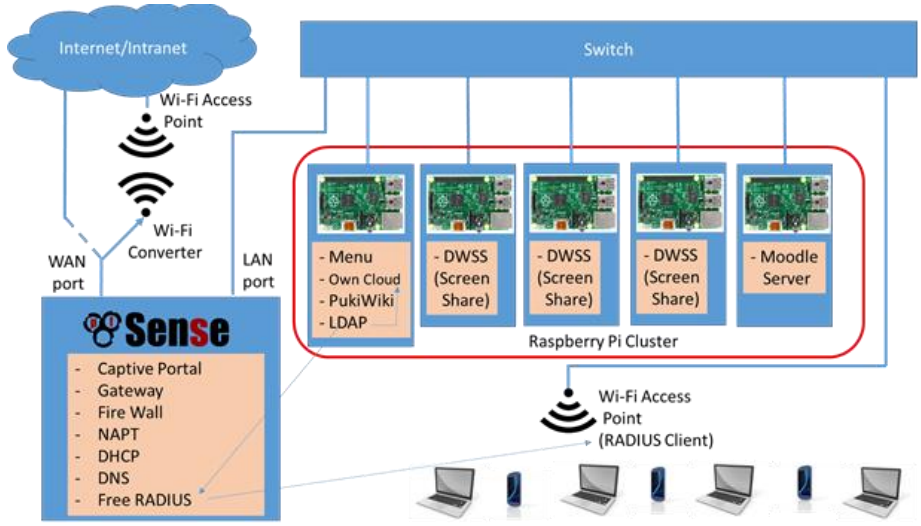

図 2. ポータブルクラウド 2016

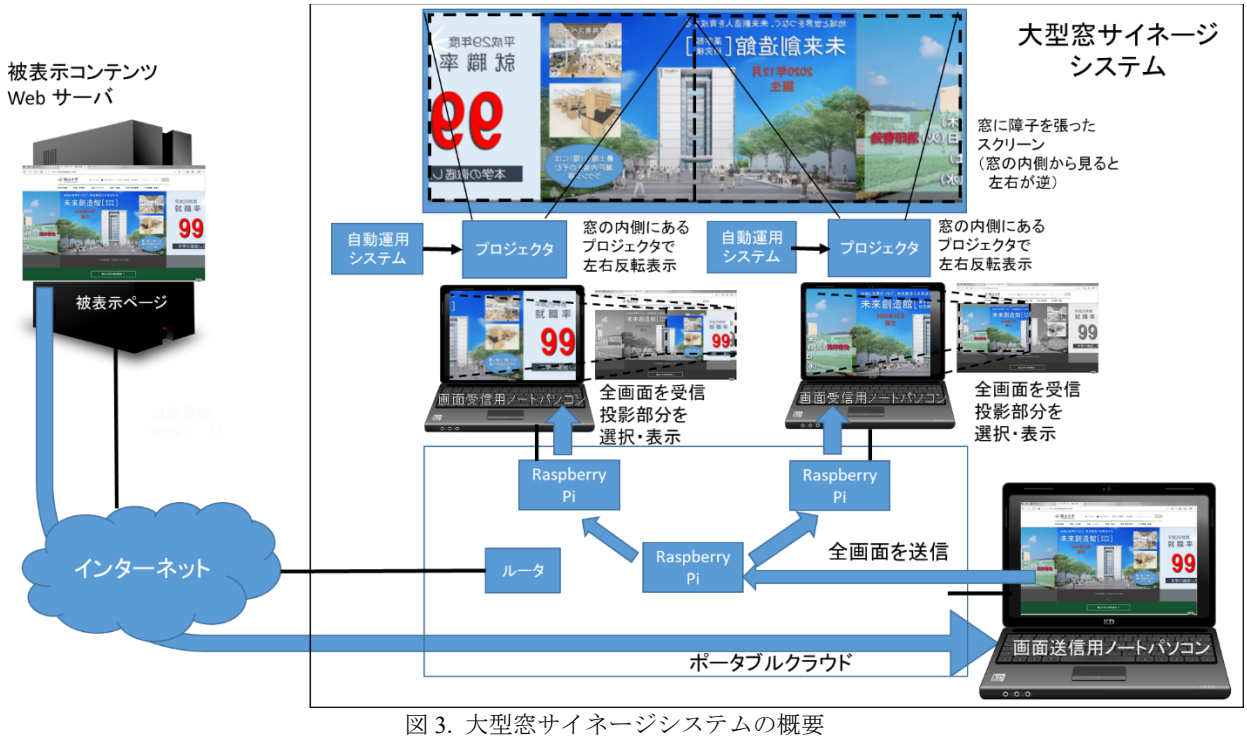

図 3. 大型窓サイネージシステムの概要

画面送信用ノートパソコンはポータブルクラウドのルータを通じて、表示したい Web ページをインターネットからダウンロードして、自分のディスプレイに表示する。自分のディスプレイに表示された画面の全体部分が，すべての画面受信用ノートパソコンに送信される。それぞれの画面受信用ノートパソコンでは画像送信用ノートパソコンのディスプレイに表示された画像の全体が受信されるが、プロジェクタで表示する部分のみを選んで、ディスプレイ全体に表示するように利用者が操作して表示する。ここで、窓の外側から見ると、内側の画像の左右が反転されて表示されるので、それに合わせて、表示する部分を選ぶ。プロジェクタでは、画面受信用ノートパソコンの画面に表示された画像の左右を反転して窓に張られたスクリーンに投影する。

4. Wiki IoT

プロジェクタの自動運用システムは Wiki IoT [10][11] を使って実現されている.

図 4 に, Wiki IoT の概要を示す. Wiki IoT は，インターネット上の Wiki ページと，センサが接続された Wiki Bot や，センサネットワークと通信が可能な Wiki Bot/Gateway や，センサやセンサネットワークが接続されていない Wiki Bot から構成されている．センサだけでなく，アクチュエータを使うこともできる.

Vol.2019-IOT-47 No.10
Vol.2019-SPT-35 No.10
2019/9/20

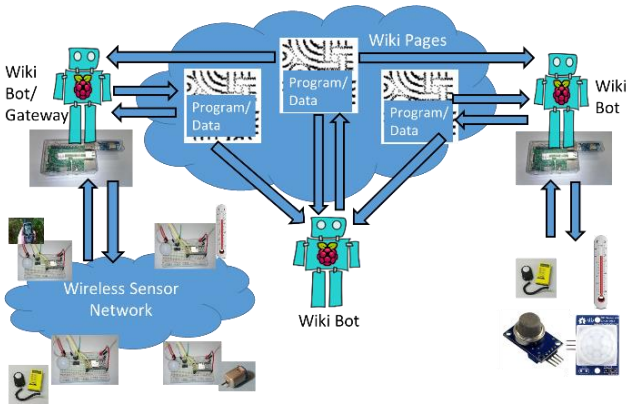

図 4. Wiki IoT の概要

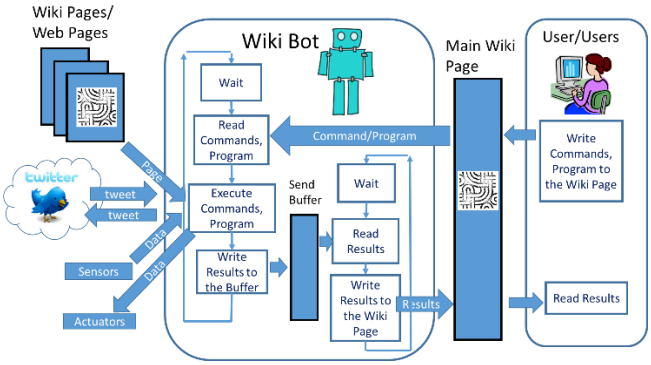

図 5. Wikii Bot の概要

```
objectPage http://[redacted]/index.php?Basi
device yamaRasPiDp9_1 or yamaRasPiDp9_2 start after no w
command: set readInterval=60000
command: set execInterval=0
command: program ex1
program: ex("service","clear sendBuffer")
program: s=0
program: for i=0 to 10
program:   s=s+i
program:   ex("service","putSendBuffer "+s)
program: next i
program: ex("service","sendResults.")
command: end ex1
command: run ex1
result:
0
1
3
6
10
15
21
28
36
45
55
currentDevice="yamaRasPiDp9_1",Date=2019/6/26/ 22:50:46
```

図 6. Wiki ページに書きこまれたスクリプトと
その実行結果の例

図 5 に, Wiki Bot の概要を示す. Wiki Bot はスクリプトインタープリタであり, Wiki Bot の振る舞いは, Wiki ページに記述されたスクリプトによって定まる. Wiki ページのスクリプトは, コマンドの列とプログラムで構成される.

Wiki Bot のハードウェアとして, Android 端末[12]にセンサやアクチュエータを付けた Arduino ボード[13]を接続したもの, Raspberry Pi の GPIO にセンサやアクチュエータを接続したもの, GPIO には何も接続していない Raspberry Pi, Windows パソコン, などが利用できる[14]. 本自動運転システムでは, Raspberry Pi に Arduino を接続して作成した赤外線リモコン送受信機を USB ケーブルで接続して Wiki Bot としている.

Wiki Bot は NIC や Wi-Fi を通じてインターネット上の Wiki ソフトウェアと通信を行う. 実行時には, Wiki Bot に接続されたセンサや, Wiki Bot に接続された無線センサネットワーク(WSN)を制御し, データを入力するが, センサや無線センサネットワークを持たない Wiki Bot も存在する. この Wiki Bot は, Wiki IoT の Wiki ページや, その他のページに書かれたデータを解析するために利用される.

Wiki Bot 側から定期的にそのスクリプトが記述された Wiki ページを読みに行くため, 多くの場合は, Wiki Bot と Wiki ページの間に NAT やファイヤーウォールが存在しても, Wiki ページに書かれたスクリプトによって, Wiki Bot を制御することができる. Wiki Bot が Wiki ページを読みに行く間隔も, スクリプトで変更することができる.

図 5 は, Wiki Bot の振る舞いも示している. Wiki Bot が起動した後, Wiki Bot は以下の手順を繰り返す.

1. 指定された Wiki ページに書かれたスクリプトを読み込む. この Wiki ページのことを, Object ページと呼ぶ. Object ページの URL は Wiki Bot の設定ファイルに書かれている. Wiki Bot の GUI でこの URL を設定することもできる.
2. スクリプトを実行する. このとき, 取り付けられたセンサや WSN の先にあるセンサの設定を行ったり, センサからデータを入力したり, Wiki ページや他の WWW のページのデータを入力したり, 入力したデータを処理したり, Twitter にアクセスしたりする.
3. スクリプトを実行後, 実行結果を指定された Object ページに書きこむ.

図 6 に Wiki ページに書きこまれたスクリプトとその実行結果の例を示す. このスクリプトは Wiki Bot に対して以下のような指示を行う. :

- “command: set readInterval=60000”の行により, 1 分間に一度, この Object ページを Wiki Bot が読み込むようにする.
- 以下部分でプログラムを定義する.
  - command: program ex1
  - program: …
  - …
  - program: …
  - command: end ex1

Vol.2019-IOT-47 No.10
Vol.2019-SPT-35 No.10
2019/9/20

・ “command: program ex1” と “command: end ex1” の間が定義されるプログラムで, ex1 と名付けられて Wiki Bot に格納されることを示す. ”program:”で始まる行が, その行がプログラムの一部であることを表す. このプログラムは 1 から 10 までの和を, 途中経過を表示しながら計算するものである.

・ “command: run ex1” によって, 格納されたプログラム ex1 が実行される.

このプログラムは, 内部では Lisp に変換して動作させているため再帰的な関数の定義と実行が可能である. 配列としてハッシュ表を利用できる. 表計算に関する関数もいくつか利用可能であり, CSV を表に読み込んで, データ解析や加工を行うことが可能である.

複数の Object Page で構成された Wiki IoT の場合, いくつかの Object Page 内に, 同じスクリプトを持つ場合がある. 同じ記述を多くの Object Page で行うのは無駄があるし, そのスクリプトを変更するとき, すべての Object Page の同じ部分を変更するのは大変であるので, オブジェクト指向プログラミング言語の場合と同様に, 同じ記述の部分を一か所の Wiki ページにまとめ, Object Page にその Wiki ページを挿入することができる. 挿入されるページのことを Class Page と呼ぶことにする. スクリプトの, 挿入したい行に, Class Page の URL を指定する“include” コマンドを書くことにより, Object Page に Class ページを挿入することができる.

5. プロジェクタの自動運用システム

自動運用システムのハードウェアは, インターネット上の Wiki サーバと, ポータブルクラウドに接続された複数の Raspberry Pi と, その Raspberry Pi に USB ケーブルで接続された赤外線リモコン送受信機で構成されている. この赤外線リモコン送受信機は Arduino を使って作成されている.

このハードウェアの上で「Wiki IoT」を動作させて自動運用システムを動作させている.

Wiki ページにかかれたスクリプトにより, 遠隔地からの全プロジェクタの強制 On, 強制 Off, スケジュールに基づいた自動運転が行われる.

図 7 にプロジェクタの自動運用システムの概要を示す.

それぞれのプロジェクタに対して, On/Off を行う赤外線信号を送る「Arduino 赤外線リモコン送受信機」が設置され, その送受信機の赤外線 LED がプロジェクタの赤外線受信窓に取り付けられている. 図 8 に Arduino 赤外線リモコン送受信機の回路を示す. この Arduino には, shirriff 氏が作成した IRremote ライブラリを使ったスケッチ(プログラム)が書き込まれている. このスケッチは外部の赤外線リモコンの信号を受信し, その信号を解読してシリアルモニタに表示する機能や, ボタンを押すことで決まった機種のプロジェクタの電源の On/Off を行う機能や, USB シリアルインターフェースからコマンドを受け取り, そのコマンドに従って, 赤外線リモコンのコマンドを送信する機能を持っている.

それぞれの Arduino 赤外線リモコン送受信機には Raspberry Pi が USB ケーブルで接続されていて, この 2 つが Wiki IoT の Wiki Bot になる. Wiki Bot は定期的にそれに対応した Wiki ページ(Object Page)を読む. その Object Page に, すべてのプロジェクタを同時に操作するための Class Page を include するコマンドが記述されており. この Class Page に書かれたスクリプトが, それぞれの Wiki Bot で実行され, プロジェクタの On/Off が制御される.

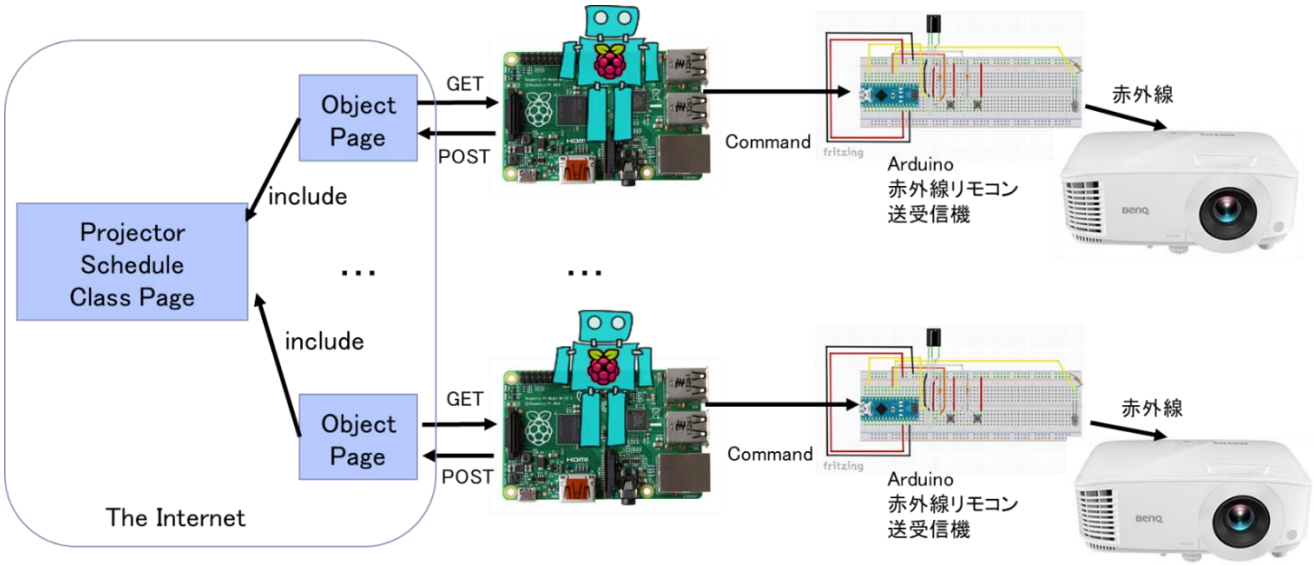

図 7. プロジェクタの自動運用システムの概要

Vol.2019-IOT-47 No.10
Vol.2019-SPT-35 No.10
2019/9/20

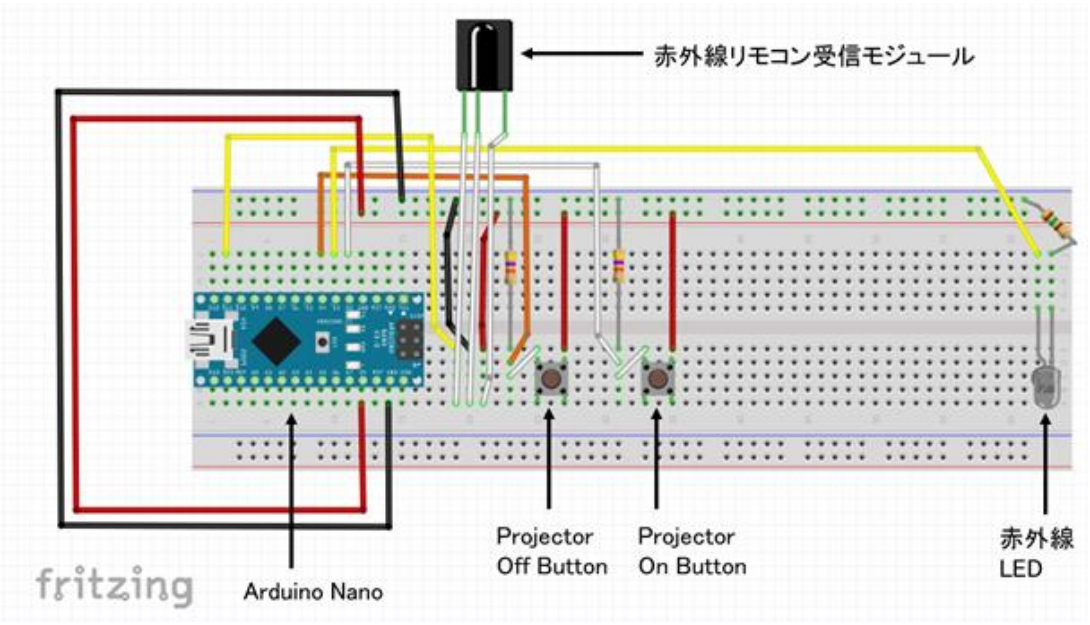

図 8. Arduino 赤外線リモコン送受信機

図 9 にプロジェクタを同時に操作するためのスクリプトを記述した Class Page (Projector_ScheduleClass)の一部を示す.

この記述の中で, def onPrj()={…} は Raspberry Pi から Arduino 赤外線リモコン送受信機に対して, プロジェクタの電源を On にする赤外線信号を発信するためのコマンドを送信する関数を定義している. この定義の中の関数 ex(“pi4j”, “serial send ¥”irCmd NEC 0xCF20D 32;¥”.”) がそれを行っている. “pi4j” は, Raspberry Pi の入出力を行うオブジェクトを表す. “serial send” は, そのオブジェクトから serial インターフェースを通じて, send 以降の文字列, “irCmd NEC 0xCF20D 32;”を送信することを表す. Arduino 赤外線リモコン送受信機はその文字列を受信し, 先頭の irCmd を見て, これが赤外線リモコンのコマンドであることを認識し, それ以降の “NEC 0xCF20D 32” をリモコンの信号に変換し, 赤外線信号としてプロジェクタに送る. NEC はこのコマンドが NEC の赤外線リモコンの形式であることを表す. 0xCF20D が赤外線リモコンのプロジェクタを On にするコマンドであり, 32 はコマンド長が 32 bit であることを表す. なお, この NEC 0xCF20D 32 は, Arduino 赤外線リモコン送受信機で, プロジェクタのリモコンの On ボタンを押したときに発生された赤外線信号を受信して得られたコマンドである.

def offPrj()={…} は Raspberry Pi から Arduino 赤外線リモコン送受信機に対して, プロジェクタの電源を Off にする赤外線信号を発信するためのコマンドを送信する関数を定義している. この定義の中の関数 ex(“pi4j”, “serial send ¥” irCmd NEC 0xC728D 32;¥” .”) がそれを行っている. ex 関数の中身については, 赤外線リモコンのコマンド以外は同じである. NEC 0xC728D は, Arduino 赤外線リモコン送受信機で, プロジェクタのリモコンの Off ボタンを押したときに発生された赤外線信号を受信して得られたコマンドである. 今回使用しているプロジェクタの場合, プロジェクタの電源を切る場合, Off ボタンを続けて 2 回おさないと, 電源が切れない. これを実現するために, 同じコマンドを, 2 秒の間隔を開けて送信している.

comTab(0,0)=… : comTab(0,1)= … の行は, 配列 comTab の要素 comTab(0,0)に代入された日時に, 配列 comTab の要素 comTab(0,1)の文字列に対応した関数が実行されることを表す. このように, 日時とその日時に行う動作を配列に記録しておくことにより, 設定した日時に設定した動作が行われる.

mode=”…”の行は運用モードの指定を行っている. 運用モードを表す“….”の中に入れる文字列によって, プロジェクタをすべて On にしたり, Off にしたり, comTab に代入された値により自動運用したりすることを切り替えることができる. なお, この Class Page はすべての Object Page で include されているので, この行の mode を書き換えることにより, すべてのプロジェクタを同時に制御できることになる.

また, すべてのプロジェクタが同じメーカーのプロジェクタであるとは限らない. このときに同時にプロジェクタを制御しようとすると、それぞれのメーカーのプロジェクタに適した On/Off コマンドを発行する必要が生じる. Wiki IoT は関数の Overriding の機能を持っており, 異なるコマンド体系を持ったプロジェクタを操作する Wiki Bot の Object Page において, 関数 onPrj や offPrj を定義しなおすことにより, この問題に対処可能である.

6. 関連研究

6.1 プロジェクションマッピング

第 1 章に記述したように、夜間の建物に映像を投影する方法として、プロジェクションマッピングが良く使われている。プロジェクションマッピングは、建物の凹凸

Vol.2019-IOT-47 No.10
Vol.2019-SPT-35 No.10
2019/9/20

も考慮して映像を投影することにより、大きな映像を効果的に投影することが可能である。しかしながら、建物の外部から投影することが必要となる。

```
command: set readInterval=60000
command: set execInterval=500
command: set sendInterval=300000
command: program ex1
program: def abs(x)= if x>0 then x else -x
program: def onPrj()=
program: {
program:   ex("pi4j", "serial send \"irCmd NEC 0xCF20D 32;\".")
program: }
program: def offPrj()=
program: {
program:   ex("pi4j", "serial send \"irCmd NEC 0xC728D 32;\".")
program:   delay(2000)
program:   ex("pi4j", "serial send \"irCmd NEC 0xC728D 32;\".")
program: }
program: dim comTab
program: ex("service","clear sendBuffer")
program: output=""
program: dx="2019/8/13/ 0:"
program: comTab(0,0)=dx+"5:00":comTab(0,1)="onPrj"
program: comTab(1,0)=dx+"10:00":comTab(1,1)="offPrj"
```

・・・

```
program: '
program: ' projector operation mode ... "allOn", "allOff", "schedule"
program: mode="schedule"
program: '
program: if mode="allOn" then { onPrj(): delay(2000) }
program: else
program: if mode="allOff" then {offPrj(): delay(2000)}
program: else
program: if mode="schedule" then
program: {
program:   for i=0 to 11
program:     t=comTab(i,0):lt=date2l(t)
program:     ct=ex("service","getCurrentDate."):lct=date2l(ct)
program:     dt=abs(lt-lct)
program:     if dt<505 then
program:     {
program:        cmd=comTab(i,1)
program:        if cmd="onPrj" then onPrj()
program:        if cmd="offPrj" then offPrj()
program:        delay(800)
program:     }
program: next i
program: }
program: 'ex("service","sendResults.")
command: end ex1
```

図 9. プロジェクタ自動運用システムの
Projector_ScheduleClass の一部

```
objectPage http://[redacted]/index.php?prj-1-01
device yamaRasPiDp9_1 or yamaRasPiDp9_2 start after no write for 10 min.
include http://[redacted]/index.php?ProjectorScheduleClass-1
command: run ex1
result:
currentDevice="yamaRasPiDp1_1",Date=2019/8/13/ 0:56:55
```

図 10. プロジェクタ自動運用システムの Object Page の例

街中で、建物の外部にプロジェクタ設置場所を確保することは、外部との交渉や費用が必要となり、気軽には実施できない。大きな画像の投影を行うにはそれなりの輝度を持ったプロジェクタが必要となり、この部分でも大きな費用が必要となる。

これに対して大型窓サイネージは、投影範囲は窓に限られるが、建物内部にプロジェクターを設置するため、建物外部の場所を確保する必要はなく、建物全体を投影する場合とくらべて、弱い照度のプロジェクタで、十分な明るさの映像投影を行うことが可能である。

6.2 大型 LED ディスプレイ

第 1 章に記述したように、大型 LED 表示装置を使ったデジタルサイネージもよく利用されている。この表示装置は昼間でも鮮明な表示を行うことができる。しかしながら、この表示装置を設置する場合、壁などに表示装置を固定するための工事が必要になる。大画面の表示を行う場合、窓をふさぐ必要も出てきて、人間が建物内部で活動するときに不具合が生じる場合もある。

これに対して大型窓サイネージは、昼間の投影を行うことはできないが、窓に障子を張ることで、昼間の室内の、ある程度の明るさを確保すると同時に、夜間の、使われていない部屋の窓を有効利用して、映像表示することができる

6.3 アミッド・スクリーン

アミッド・スクリーン技術は，空間投影の一般化 技術として青木敬士氏により開発されオープン化さ れた、家庭用網戸をプロジェクタ投影のスクリーンとして利用する技術である[16]. 本論文 7.1 節の実験で透過スクリーンを使った実験を行ったがこのスクリーンは非常に高価である。そこで，安価な技術として，アミッド・スクリーンが登場している。

我々の大型窓サイネージシステムは、当初、家庭用網戸を利用することを考えていた。しかしながら、第 7 章で示した実験により、障子紙を使って、宮地茂記念館における投影を行った。

網戸は、障子紙と比べると透過性に優れており、通常の部屋の利用に対する影響が少なく、また、空間上に映像が浮き上がるような投影効果もある。しかしながら、遮光性のある窓では、外部からはほとんど映像が映らなかった。これに対して障子紙は、夜間ではあるが、遮光性のある窓で投影された映像を室外から違和感なく認識することができた。網戸ほどは透過性はないが、室内の明るさを保ちながら同時にプライバシーを高める目的などのため、多くの家屋で利用されている。また、空間上に映像が浮き上がるような投影効果は、今回の宮地茂記念館における投影では要求していない。

6.4 スマートスピーカー

2015 年頃から Amazon Echo や Google Home などのスマートスピーカーが販売されている．スマートスピーカと，家電製品を赤外線で操作するスマートリモコンを組み合わせることにより，音声で家電製品を操作

することが可能になる．外部のサービスやスマホアプリを組み合わせることにより，外出先から家電製品を操作することも可能になる[17]．プロジェクタについても適切に設定することにより，外部から操作することは可能になると思われる．Google カレンダー[18]とIFTTT[19]など組み合わせることにより，スケジュールに沿った操作も可能だと思われる．しかしながら，多数の機器の同時制御が可能かどうか不明である．

本自動運用システムは，多数のプロジェクタを同時に操作することが可能である．

7. おわりに

大型幅広デジタルサイネージシステムとその自動運用システムについて述べた．このデジタルサイネージシステムはビルの多くの窓に障子紙を貼り，窓の部屋側から多数のプロジェクタで多数の画像を投影し，全体として大きな１枚の画像を構成して表示するものである．これは多数のパソコンに同じ画像を送信し，それぞれのパソコンでプロジェクタで表示すべき部分を選んで拡大表示し，それをプロジェクタで表示することで実現している．

同じ画像を多数のパソコンに送信するため，ポータブルクラウドの画面共有機能を利用している．利用するパソコンとプロジェクタの数を増減することにより，様々なサイズの，様々な縦横比の画像を表示することが可能である．

自動運用システムはこのサイネージシステムの多数のプロジェクタを，自動的に，予め設定された時刻において一斉に、点灯・消灯したり，管理者によって強制的に点灯・消灯したりするものである．この時刻設定や強制的な点灯・消灯は外部の遠隔地から行うことができるため，ビルに人がいない夜間でも臨機応変にプロジェクタの点灯・消灯時間を変更したり，強制的に点灯・消灯したりすることができる．この自動運用システムは，Wiki IoT を使って実現した．

今後は，2012 年に MIT で行われたような[20]建物全体の窓を使って，ゲームで遊ぶようなことも可能にしたい．

謝辞 本研究の一部は JSPS 科研費 16K00197, 15H03055 福山大学研究助成金の助成を受けて実施しました．

参考文献

[1]デジタルサイネージ: https://ja.wikipedia.org/wiki/デジタルサイネージ, as of 16 Dec. 2018.

[2] プロジェクションマッピング: https://ja.wikipedia.org/wiki/プロジェクションマッピング, as of 16 Dec. 2018.

[3]山之上卓, 樋髙想士, 小林幸司, 小荒田裕理, 片桐太樹, 小田謙太郎, 下園幸一, “ポータブルクラウドの試作”, 情報処理学会研究報告, Vol.2013-IOT-22, No.12, (2013).

[4]山之上 卓, 小田 謙太郎, 下園 幸一, 小荒田 裕理, “中大規模会議用携帯クラウドコンピューティング環境の概念”, 第１２回情報科学技術フォーラム講演論文集, L-026, pp.275-280 (2013).

[5]Takashi Yamanoue, Soshi Tetaka, Kentaro Oda, Kochi Shimozono, "Portable Cloud Computing System - A System which Makes Everywhere an ICT Enhanced Classroom", Proceedings of the 42th annual ACM SIGUCCS conference on User services, Salt Lake City, Utah, US, 4-7 (Nov. 2014).

[6]山之上 卓, 杉田 裕次郎, 小荒田 裕理, 小田 謙太郎, 下園 幸一, “デスクトップ画像共有システムのための、トーナメントアルゴリズムを使った負荷分散機構”, マルチメディア，分散，協調とモバイル(DICOMO2013)シンポジウム講演論文集, pp.429-434 (2013).

[7]Yamanoue, T., Koarata, Y., Oda, K., Shimozono, K, "A Technique to Assign an Appropriate Server to a Client, for a CDN Consists of Servers at the Global Internet and Hierarchical Private Networks", Proc. of The 38th Annual International Computer Software & Applications Conference (COMPSAC2013/ADMNET WS) , pp.90-95, Västerås, Sweden, 21-25 (Jul. 2014).

[8]Takashi Yamanoue, "Optimizing Data Partitioning at Broadcasting the Data on Balanced N-ary Tree Formed P2P Systems", 6th International Conference on E-Service and Knowledge Management (ESKM 2015), Okayama, Japan(2015).

[9]HTML5: https://ja.wikipedia.org/wiki/HTML5, as is Jan. 2017.

[10]山之上 卓, "インターネット上の Wiki ページにより NAT 背後のセンサ端末の設定変更や制御が可能な IoT システムによるサーバとサーバ室の監視", 情報処理学会シンポジウム，インターネットと運用技術シンポジウム 2016 論文集，情報処理学会シンポジウムシリーズ No. 2018, pp. 62-69, (2018-12)

[11]Takashi Yamanoue, "Bot Computing using the Power of Wiki Collaboration," 10th International Conference on E-Service and Knowledge Management (ESKM 2019), July 7 - 12, Toyama Japan.

[12]Android, https://www.android.com as of Jan. 30, 2017.

[13]Arduino, https://www.arduino.cc as of Jan. 30, 2017.

[14] Takashi Yamanoue, Kentaro Oda, Koichi Shimozono, A M2M system using Arduino, Android and Wiki Software, Proceedings of the 3rd IIAI International Conference on e-Services and Knowledge Management (IIAI ESKM 2012), pp.123-128, Fukuoka, Japan, 20-22 (Sep. 2012).

[15]Shirriff, IRremote, https://www.arduinolibraries.info/libraries/i-rremote, as of 17 Aug. 2018.

[16]青木敬士, ”アミッドスクリーン研究室” ,. (http://textalk.moe-nifty.com/amid/, 参照:2018 年 12 月 16 日)

[17] 鴻池賢三, “リモコンの進化形へ！ 家電操作の歴史から見る「スマートスピーカー」”, 価格.com マガジン, https://kakakumag.com/av-kaden/?id=11099, as of Aug. 18, 2019.

[18]Google カレンダー, https://ja.wikipedia.org/wiki/Google_カレンダー, as of Aug. 18, 2019.

[19]IFTTT, https://ifttt.com, as of Aug. 18, 2019.

[20]MIT: Playing Tetris ... on a Building!: https://www.youtube.com/watch?v=IAIPUGO1iko, as of 16 Dec. 2018.