SNSとストリーミングサーバを利用した気温・湿度 やカメラ画像を投稿するBot装置の開発

著者	新谷 聖, 若沢 卓道, 中山 滉平, 佐藤 孝哉, 柴田 幸司, 花田 一磨							
著者別名	ARAYA Syo, WAKASAWA Takumichi, NAKAYAMA							
	Kouhei, SATOU Takaya, SHIBATA Kouji, HANADA							
	Kazuma							
雑誌名	八戸工業大学紀要							
巻	37							
ページ	159-166							
発行年	2018-03-31							
URL	http://id.nii.ac.jp/1078/00003833/							



SNS とストリーミングサーバを利用した気温・湿度 やカメラ画像を投稿する Bot 装置の開発

新谷 聖[†]・若沢 卓道[†]・中山 滉平^{††}・佐藤 孝哉^{††}・柴田 幸司^{†††}・花田 一磨^{††††}

Development of a Bot device for sending a temperature, humidity and image using SNS and streaming server

Syo ARAYA[†], Takumichi WAKASAWA[†], Kouhei NAKAYAMA[†], Takaya SATOU[†], Kouji SHIBATA^{††} and Kazuma HANADA^{†††}

ABSTRACT

In this research, we used a microcomputer remote monitoring system using a microcomputer developed by the authors to process information measured by temperature and humidity sensors and camera image information with a Linux microcomputer, We built a system to post as a tweet. As a result, access to the Twitter server has been confirmed for over a month. As a concrete construction procedure, connect the web camera and temperature and humidity sensor (USBRH) to Raspberry Pi which is a Linux microcomputer, install necessary packages, edit and set the program files, etc. In the Python program, Processing of image and temperature and humidity information and authentication processing of account information were carried out. Temperature humidity information, camera image, and time information are written to the specified Twitter account. By constructing this system, I think that it can help to understand how computer / network education, IoT technology and SNS are used.

Key Words: SNS, remote monitoring, Linux microcomputer, Raspberry Pi, IoT technology キーワード: SNS, 遠隔監視, Linuxマイコン, Raspberry Pi, IoT技術

1.背景と目的

平成 30年 1月 9日受付

- † 工学部電気電子システム学科・卒業生
- †† 工学部電気電子システム学科・4年
- ††† 工学部電気電子システム学科・准教授
- †††† 工学部電気電子システム学科・講師

近年SNSやIoT(Internet of Things)が急速に発展し ており、これは世界中に張り巡らされたインタ ーネットや携帯電話の基地局といった情報伝送 路の発達と、大量のデータを送ることが可能に なったこと、パソコンやスマホといった携帯端 末の爆発的な普及が関係している^{1,2)}。技術の発 達によって私達の生活の利便性は発達したが、 効率良く活用するためにはシステムの仕組みや どのような場所でどのように利用されているか を理解し運用する必要がある。

一方、筆者らにより開発されたLinux搭載のマ イコンと携帯電話網に接続可能なUSBモデムを組 み合わせ、屋外の任意の場所に設置可能な超小 型かつ安価なテレメータシステムは、屋外設置 により温度・湿度だけでなく太陽光パネルによ る発電による発電状況や、バッテリー、負荷お よび消費電力の状況を確認できている³⁾⁻⁵⁾。そこ で本研究では、これらの研究を発展させ、気 温・湿度とカメラ画像をSNS(Twitter)に自動配 信するBot装置を構築し、インターネット上の Twitterサーバへの連続的な接続を確認した。具体 的には、LinuxマイコンであるRaspberry PiにWebカ メラと温湿度センサを接続し、Pythonプログラム でカメラ画像と温湿度の情報を処理し、アカウ ント等の認証処理を行いSNSであるTwitterにカメ ラ画像と温湿度などをつぶやきとして投稿する システムを構築した。一方、Raspberry Pi2は4コ アのCPUを有しており、複数のタスクを動作させ られる。そこで、動画をストリーミングサーバ であるUstreamへとアップロードするサービスを 実装し、動画の連続配信が可能か検討した。本 システムは、コンピュータ・ネットワーク教育 の応用として、実際のシステム構築によりIoT技 術やSNSがどのように利用されてシステムが作ら れているか、理解させる助けになると考える。

2. 開発したシステムの概要

今回開発したシステムのブロック図を図1に示 す。本システムでは、Linuxマイコンである Raspberry Pi2モデルBに接続したUSBカメラおよび 温湿度センサ(USBRH)から情報を取得する。 そして、Pythonにてプログラミング処理やTwitter サーバへのユーザ認証を行い、投稿した時間、 その時間の画像情報、気温・湿度情報をTwitterに 投稿する。これらのシステム一式を図2に示す様 に防塵防水用の小型ケースに収納して、公共施 設に設置できるようにした。



ライブラリであるTwythonを使用

図1 構築したシステムの概要



図2 構築したシステム

3. 構築手順

この Twitter Bot を構築するため、まずは Twitter アカウントの作成と Twitter アプリ開発者用のサ イトへのログインとセットアップを行う必要が ある。その手順は以下の図3の通り。

- ① Twitter アカウントと開発者用サイトのセット アップ
- A) Bot 用の twitter アカウントを作成する。

- B) アプリ開発者用のサイトにアクセスし先のア カウントでログイン(ttps://apps.twitter.com/)
- C) アプリケーションの新規作成を行い、アプリ 詳細に名前と概要、URLを書き込む。
- D) Keys and access tokens をクリックし Application Settings の Consumer Key (API Key)と Consumer Secret (API Secret)の番号情報をコピーする。 Your Access Token の Creat my access token クリ ックして access token を発行し、Access Token と Access Token Secret の認証番号情報をコピー する。

この4つの手順は、後のユーザ認証を行う時に使用する。



Attend Canad	Real and with Daniel and an inclusion	
Nacional Lengt		
	The second second	
Application	Actons	
Banana P	assumer Key and Secret Chango Age Permissions	
and the second of the		
- and an and a		

図 3 アプリケーションのセッティング

- ② Raspberry Piのセットアップ
- A) セットアップツール等のインストール そして次に、RaspbianOSのコマンドラインか ら以下の通り、Raspberry PiにTwitter用のライ ブラリであるTwythonのインストールを行い、 さらに USB カメラと温湿度センサ (USBRH)のドライバをインストールする。

#setuptoolsのインストール \$ sudo wget http:// peak.telecommunity.com/dist/ez_setup.py \$ sudo python ez_setup.py

#pipのインストール \$ sudo wget https:// raw.github.com/pypa/pip/master/contrib/ get-pip.py \$ sudo python get-pip.py

#twythonのインストール

\$ sudo pip install twython

カメラ画像を取得するためのWebカメラのセット アップ

\$ sudo apt-get install fswebcam

ここまでのセットアップツール等がインストー ルし終わったら再起動を行う。

B) そして更に、温湿度センサ(USBRH)を Raspberry Piに接続後、同様にRaspbianOSから 以下の通り、温湿度センサ(USBRH)のドライ バをインストールする。

#USBRHのインストール \$ sudo apt-get install gcc libusb-dev \$ cd/tmp \$ sudo wget http://www.dd.iij4u.or.jp/~briareos/soft/ usbrh-0.05.tar.gz \$ tar xvfz usbrh-0.05.tar.gz \$ cp -rf usbrh-0.05/usr/local/src/usbrh-0.05

温湿度センサ(USBRH)のインストールが終わった ら再起動を行う。そして、その後ドライバのイ ンストールとLinuxマイコンの再起動が終わった ら、usbrh_main.cの追加編集を行う。その際、 USBRHはRaspberry Piではそのままだと使用でき ないバグがあり、下記の部分を追加編集する。

iff(rc = usb_set_configuration(dh, dev->config->bConfigurationValue))<0){ if(rc = usb_detach_kemel_driver_np(dh, dev->config->interface->altsetting-> bInterfaceNumber)<0){/追加編集 puts("usb_set_configuration error"); usb_close(dh); exit(3); }/追加編集 }

編集後、再起動する。

\$ cd/tmp

e seizet with adjust

C) 次に、usbrhのコマンドで温湿度の値を確認 できるように設定する。この操作でusbrhと コマンドを打ち込むだけで温湿度の値を確 認できるようになる。

\$ cd usbrh-0.05 \$ make \$ sudo chown root:root usbrh \$ sudo chmod u+s usbrh \$ sudo mv -i usbrh /usr/local/bin/ \$ usbrh 21.41 39.68

そして、コマンドで温度と湿度が確認できたら 再起動を行う。これでLinuxマイコンへのインス トールとセットアップの作業が終了する。

③ Twitter投稿用のPythonプログラムの記述

A) 次に、新規でPythonプログラムを作成する。 今回、Twitter上に表示させるのは時間、その 時間の画像情報、気温・湿度情報とする。 これらの機能をスクリプトとして表現した Ptythonプログラムを以下に示す。

twibot.py # coding: utf-8

import commands import sys import time import os from twython import Twython

CONSUMER_KEY = 'xxxxxxxxx' CONSUMER_SECRET = ' xxxxxxxxx ' ACCESS_KEY = xxxxxxxxxx ' ACCESS_SECRET = xxxxxxxxxxx '

api=Twython(CONSUMER_KEY, CONSUMER_SECRET, ACCESS_KEY,ACCESS_SECRET)

#USBRHからの気温および湿度を外部コマンドで実行 usbrh_pack=commands.getoutput("usbrh") usbrh_xx, usbrh_yy=(usbrh_pack_split())

#現在時刻を取得

timestamp='date +%H時%M分%S秒' current_time=os.popen(timestamp).readline().strip() #現在の画像を撮影取得 os.system ("fswebcam -d/dev/video0 -r1280x960 --no-banner roompic.jpg")

#結果出力部分 print current_time print usbh_pack photo = open('roompic.jpg','rb') #twitterでの表示 image_ids = api.upload_media(media=photo) api.update_status(status= current_time+ '現在、\n部屋の気温は '+usbh_xx+'度 \n 湿度は '+usbh_yy+'%です。\n', media ids=image_ids['media id'])

B) crontabコマンドの設定

crontabコマンドとはunix系で使用できる定時実 行のスケジュール管理に使われるコマンドで ある。これを使用すると投稿する日時の指定 や、分単位で投稿時間を指定できる。そこで 今回は1分間隔で投稿するようにスケジュール を設定する。

設定が終わったら再起動を行う。再起動すると 自動的にTwitterへの投稿が開始される。

\$ crontab -e #設定ファイルを開いたら一番下に記入

#(/home/pi/)下は自分がbotを作成したディレクトリを指定

* * * * * /home/pi/twibot.py

4. Twitter Bot装置の動作の確認

Raspberry Piに以上の設定を施した後、装置を連 続稼働させることにより、Twitterアカウント (ttps://twitter.com/kanshi0)として運用状況の確認を 行った。まず、表示および実行結果については、 図4および図5より、月、日、時間、気温と湿度、 その場の画像を撮影して投稿が確認できた。お よそ12時間の間に1分間隔で、連続投稿している ことを確認できている。さらに図6と図7の通り、 屋内で常時運用するために防塵用小型ケースに 収納し、コンセントを使用する屋内での常時監 視用の遠隔監視システムを作成することで、小 規模なスペースにも設置することが可能となっ た。本システムは、八戸工業大学の柴田研究室 にて2016年10月8日から12月1日現在、1ヶ月半以 上連続で稼働している。実際の活用例として、 研究室の鍵が保管されている場所を常時監視し、 研究室内の人の有無を確認している。



図 4 プログラムの実行状況の確認

							· fanderigen
-	OF DR. DR. DR	the last we					
			-		1 14 21 21	14.11	
7 T T T			• • • • • •	4 (1 (H (1))	1 14 21 21	89	
	- 1 - 4 - 1 星日-時間	精の線	e er or er o		1 14 21 21	89	
+*3I	813-MAR	新の線	4 (4) (7) (4) (4)	4 (1 (H) I)	1 16 19 77	89	Ø Lenter og
	¥13-M-B	新の時	A	a -1 36 31	T 14.21.71	89	© Landigs open
	 213-M-8	5. 537-07 880	н н н н н М ()	a -1 36 31	T 14.31.71	89	0 kanshat ayar
4-173	213-1412 213-1412	57-0-88 第-0-88	A 11 11 11 1	a (1) (a) (1)	T 14 JT 71	8.9	C kandharayaan
4-173	213-1412	NT-O-EN	A		T 14 JT 77	8.9	C Landial aguar
H-173	2()-M(2	新の線	A	a 11 16 17 1	T 14 JF 77		0 kandharingnam
	20-442	* * * 新の略	A	a 11 16 17	T 14 31 71	8.9	0 taxthurapan

図5プログラムの実行状況の確認



図6防水ケース内への機器の設置



図7 学内への装置の設置

5. ストリーミングサーバによる動画の公開

本システムに用いるラズベリーパイに実装し ている OS である Raspbian は Linux をベースとし ており、マルチタスク OS であることから、複数 のプログラムを同時に並列動作させることが出 来る。さらに、Raspberry Pi2 は4 コアの CPU を有 しており、複数のタスクを動作させられる。そ こで本研究ではシステムに、動画をストリーミ ングサーバである Ustream ヘとアップロードする サービスを実装し、動画の連続配信が可能か検 討した。その手順は以下の通りである。

- ① まず、Ustream のアカウントを作成し、ログ イン
- ② RTMP URL とストリームキーを取得
- A) Ustream のホームにて、ユーザーアイコンを クリックし、ダッシュボードを開く。
- B) ダッシュボードのチャンネル設定からチャンネルを新規作成し、ライブ配信設定をクリック
- C) エンコーダ設定の右側にある「設定」タブ をクリック
- D) RTMPアドレスとストリームキーをコピー
- ③ Video4linux2のインストール Video4linux2は、初期インストールされてい る場合が多い。インストール済みであるか 確認するため、以下のコマンドを入力

\$ sudo apt-get install v4l-utils

④ ffmpeg配信サービスのインストール
以下のコマンドを入力し、ダウンロード、
コンパイル、ffmpegをインストール

\$ cd/usr/src

- \$ sudo mkdir ffmpeg
- \$ sudo chown pi:users ffimpeg
- \$ git clone \$ git://source.ffimpeg.org/ffimpeg.git ffimpeg \$ cd ffimpeg

以下のコマンドを入力する。このコマンドは実 行に多少時間がかかる。

\$./configure

コマンドが実行し終わったら、次のコマンドを 入力する。

\$ make

最後に以下のコマンドを入力する。

\$ sudo make install

- ⑤ ライブ配信の開始
- A) ライブ配信を開始するために、シェルスク リプトを作成

\$ nano ~/ustream

#!/bin/bash

RTMP_URL=<rtmpurl> STREAM_KEY=<streamkey> padsp ffinpeg -f video4linux2 -s 176x144 -r 30 -i /dev/video0 -f flv \$RTMP_URL/\$STREAM_KEY

RTMP_URL=<rtmpurl>の<rtmpurl>部分に取得した RTMP の URL を入力、

STREAM_KEY=<streamkey> の<streamkey>部分にストリームキーを入力する。

B) シェルスクリプトを実行可能にするために権
限を付与

chmod +x ~/ustream

C) コマンドを入力すると配信が開始

~/ustream

6. ストリーミングサーバの動作の確認

Raspberry Pi2 に上記設定を行った後、コマンド を入力すると、配信を開始する。今回、解像度 は 176x144、フレームレートは 30fps で動作する ように設定を行った。この時、図8および図9の ように USB カメラで撮影した動画データを Ustream 上で確認することが出来た。図10 に示す ように配信時のフレームレートは 20 から 30fps、 画像データは 200kbps で Ustream 上に転送されて いた。

しかし、画像データが時々更新されなくなり、 画面が暗転する事が確認された。これは Raspberry Pi の GPU 性能の不足や、ffmpeg と video4linux2 といったソフトウェアが、動画処理 に使用されている CPU のコア数が、1 コアのみ で処理しているからだと考えられる。

よって今後は、画像の圧縮を行ってから ustream ヘアップロードしたり、Raspberry Pi2の4 コアをすべて使用して、並列動作による動画の 処理を行わせたいと考える。



図8ストリーミング・プログラムの実行状況



図9ストリーミング・プログラムの実行状況

				×	
7月1日 福美田 話を回 えいつ-んの う	NOFTING ANTHE				
trames15451 tosi 27 cs3.0 sizer	23123ki 1 res00:25:45.20 bitrates	302	Zabite	64	ï
rane=15673 fee: 27 e:2,7 eize:	23185kB time:00:25:46.90 biteate:	20.2	24111	24	
trape= 508 fps= 27 of? 7 sizes	282216R time=00:25:48.50 bitrate=	202	2. hits	<i>1</i> 2	
raget [520] Just 27 or2.5 sizet	3025348 Liser00:25:58.20 bitrater	20.2	2 doits	Э£.	
frame=15520 fec= 27 e-2.5 gize=	\$8307k5 time=00:25:51.90 bitrate=	202.	2-bits	24	
rame*15537 fpg= 27 g12.5 sizes.	28344k8 time:00:25:53.60 hitester	202	2-hete	24	
tane=15554 fest 27 er2.7 sizet	28390kg time=00:25:55.30 bitrater	202	2-6116	A	
rase=19271 fec= 27 e=2.5 cize=	38428kB time=00:25:57.00 bitrate=	202	2.4611	Je	
trase 1587 fps= 27 or2.6 size*	28463kB time=00:25:58,60 bitrate=	202	2-bits	14	
frame=1904 fost 27 or2.5 wiget	39510k8 Fige=30;29:00.30 bitrates	202	2-drifts	74	
rase=15621 fpc= 27 o=2.5 cize=	38546k8 time=00:25:02.00 bitrate=	202.	2-bits	10	
frase*1563E fps* 27 o*2.4 size*	28592kB time=00:26:03.70 bitrate=	292.	3-bits	28	
tramer15655 fpst 27 or2.4 sizer	3983148 1 ice=00:28:05.40 bitrate=	202	2-bits	18	
frame=19072 fps= 27 a=2.5 size=	\$8989kB time=00:28:07,10 bitrate=	202.	lidits	76	
traset 15685 fpct 27 of2.4 sizet	38715kB time=00:28:08.80 bitrate=	202.	2-bits	15	
traset15708 fpst 27 or2.5 sizet	\$8752kB time=00:25:10.50 bitrate=	202.	lidate	14	
rame=15722 fps= 27 c=2.5 size=	\$8799kB time=00:28:12,10 bitrate=	202.	2dits	14	
trane*15789 fpc* 27 e/2.3 size*	33835kB time:00:28:13.80 bitrate:	202.	lide its	/s	
frame=15758 fps= 27 or2.4 size=	\$8873k8 time=00:25:15.50 bitrate=	202.	didit:	1	
frame+15773 fps= 27 <=2.3 size+	S8919kB Line=00:25:17.20 bitrate=	202.	Rabits	16	
frame=15790 fpc= 27 o/2.5 size-	33956k8 time:00:26:18.90 bitrate/	202.	libits	/s	
rame#15007 fps# 27 of2,4 size#	33004kB time=00:25:30.60 bitrate=	202.	3-bits	/4	
rame#15623 fps# 27 c#2,9 size#	5904168 time=00:25:22,20 bitrate=	202.	Isbits	14	
frame 1584) fpc= 27 o/2.6 slize-	39079k8 time:00:28:23.90 bitrate/	202.	liabite.	25	
frame=15857 fest 27 et2.3 sizet	29127k8 1 inc:00:25:25.60 bitrate:	202.	3-bits	15	
rase=15874 fps= 27 <=2,5 size=	2916468 time=00:25:27.20 bitrate=		Bubits	$\sqrt{\kappa}$	
rase=15091 fpc= 27 c=2.4 size=	39212kB time-00:26:29,00 bitrate-	202.	2 dbits	2/c	
rase*15607 fpg= 27 o*2.5 size*	39247kB time=00128130.60 bitrate=	202.	l-bits	13	
rase=15024 fps= 27 c+2.5 size4	59285k5 time=00:25:32,30 bitrate=	202	Dabits	$\sqrt{\pi}$	
rase=19841 fpc= 27 ==2.3 size=	\$9932kB Line=00:26:34.00 bitrates	202.	lideita	\sqrt{c}	
rase115658 fpst 27 ot2.7 sizet	29370kB time=00125135.70 bitrate=	202.	Nabits	18	
rase=19075 fps= 27 o/2,7 size=	20418kB time=30:25:37,40 bitrate=	302.	Babits	44	
frame=15992 fpc= 27 <=2.7 size+	39456kB time=00:20189.10 bitrate=	202.	lideits	20	
frame=16009 fpg= 27 o=2.4 size=	39505k8 time=00:25:40.80 bitrate=	212	2-bits	10	
rase*1825 fps* 27 o*2.8 size*	2024268 time=00:25:42,40 bitrate=	A12.	Schits	28	
Trate 10:42 fps= 27 m2.6 size	3958.kE 1 ime-00:20:44.10 bitrate-	202.	fidets.	10	
frame=18055 feg= 27 e-2.5 size=	39525k5 time=00125145,80 bitrate-	20.5	2-0111		
trames ittirs toss 27 ca2,4 sizes	2000048 fine#20128:47.40 bitrate#	945	Deficts	2.1	
seed 27 Bla					-

図 10 ストリーミング・プログラムの実行状況

7. まとめ

今回、LinuxマイコンであるRaspberry PiにWebカ メラと温湿度センサを接続し、Pythonプログラム でカメラ画像と温湿度の情報を処理、アカウン ト等の認証処理を行い、SNSであるTwitterに気 温・湿度、カメラ画像と時間情報を自動投稿す るBot装置を構築し、インターネット上にある Twitterサーバへの連続的な接続を可能とするシス テムを構築した。その結果、Twitterの@kanshi0に て、画像および温湿度の1分毎の投稿を実現し、1 ヶ月以上の連続動作を確認した。1ヶ月の投稿は 24時間*60分*30日=43200回となり、1分間に1回の 投稿なら、Twitterからの書き込み規制も無いこと が分かった。

動画配信では、短時間だがUstreamへライブ配 信することが出来た。その際、動画処理には1コ アのみしか使用しない事など複数の問題点を知 ることが出来た。しかしUstream上にデータをア ップロードできる事が分かったので、画像の圧 縮や4コアでの分散処理での動作による、ストリ ーミング配信を行うことで、他IoT機器への応用 にも利用できると考える。

今後の展望として、この装置を八戸市内の観 光地といった、公共施設などに設置することに より、地域おこしへの応用が可能であると考え る。さらに、エアコンなどのIoT機器と組み合わ せて、遠隔地で温湿度を監視しながら、動画で 部屋の状況を確認、エアコン操作を行えるシス テムの開発などを行いたい。

参考文献

- Zhen Zhu and Ruchun Cui, "Remote Intelligent Monitoring System Based on Embedded Internet Technology," Proceedings of the 2007 IEEE International Conference on Automation and Logistics, pp. 2665-2669, 2007-8.
- Y. Ha, "Dynamic Integration of Zigbee Home Networks into Home Gateways Using OSGi Service Registry,"IEEE Transactions on

Consumer Electronics, vol. 55, no.2, 2009.

- 柴田幸司,花田一磨,落合翼 "Linux マイコンを用いた組込み VPN による超小型センサ情報遠隔監視システムの開発" 八戸工業大学紀要 33, pp115-120, 2014-3.
- 4) 柴田幸司,花田一磨,飯野真弘,武美里,赤塚一磨 "Linuxマ イコンを用いた組込み VPN による超小型センサ情報遠隔 監視システムの開発と教育への応用" 信学技報 教育工学 研究会, Vol.114, No.441, ET2014-83, 2015-1.
- 5)柴田幸司,飯野真弘,武美里,赤塚優磨,花田一磨 "震災対応 のための Linux マイコンを用いた超小型センサ情報遠隔監 視システムの開発とネットワーク教育への適用," 電子情 報通信学会総合大会、D-15-5,2015-3
- 6) Raspberry Pi (ラズベリーパイ): Ustream で配信ビデオ (https://goo.gl/TW5yul)

要 旨

本研究では、筆者らが以前開発した Linux マイコンを使用した超小型遠隔監視システムを 利用し、気温、湿度センサにて計測した情報とカメラ画像情報を Linux マイコンで処理し、 SNS である twitter につぶやきとして投稿するシステムを構築した。これにより Twitter サー バへのアクセスが 1ヶ月以上、確認できた。具体的な構築手順としては、Linux マイコンで ある Raspberry Pi に Web カメラと温湿度センサ(USBRH)を接続し、必要なパッケージの インストールや、プログラムファイルの編集、設定などを行い、Python プログラムにて、 カメラ画像と温湿度情報の処理とアカウント情報の認証処理を行った。指定したツイッタ ーアカウントに温湿度情報、カメラ画像、時間情報が書き込まれる。本システムを構築す ることによって、コンピュータ・ネットワーク教育や IoT 技術や SNS がどのように利用さ れているかの理解の助けになると考えている。

キーワード: SNS, 遠隔監視, Linuxマイコン, RaspberryPi, IoT技術