

寒冷地での農業生産支援に必要な高安定・高精度な 温床線制御法の基礎検討

若沢 卓道[†]・中山 滉平^{††}・柴田 幸司^{†††}・花田 一磨^{††††}

Fundamental study on control method of hotbed line with high stability
and high accuracy for agricultural production support in cold district

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

ABSTRACT

One of my authors has built up a mechanism to connect securely over VPN by using various network functions of Linux microcomputer Raspberry Pi. Then, it was shown that this system can connect to clients from far away locations, and sensor information such as temperature and humidity and video information can be acquired. Therefore, in order to apply these systems to agricultural support in cold districts, this study constructed a mechanism of agricultural support system that can monitor and control on the network function VPN (Virtual Playbaet Network). We constructed a mechanism that warms the temperature inside the vinyl house cabinet by using a general-purpose high-temperature bed line that is 100 V AC and still simple in structure.

Key Words: Remote Control, Remote Monitoring, Raspberry Pi, Embedded Linux

キーワード: 遠隔制御, 遠隔監視, ラズベリーパイ, 組み込みLinux

1. 背景と目的

インターネットの普及は昨今飛躍的なもので、これに伴いデータ通信の品質や高速化、低遅延

化が進んでいる。また、従来ではテレメータ・システム等の実現にはISDN等の専用回線が必要だったが、近年では仮想プライベートネットワーク(VPN)を用いることにより、安価かつ安全に遠隔地にクライアントからサーバ等にアクセスすることが可能となった。その為、例えばセンサ情報の場合は温度や湿度情報に加え、映像や音声情報などもセキュアに送信することが可能となった。更に、近年の高速かつ低遅延な携帯電話回線網の普及により、マイコンを実装したセンサ機器がLANや公衆ネットワークだけでな

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

く、携帯電話回線網などを介して、直接情報のやり取りを行うことができる。この技術の応用として、家庭や工場に加えて、公共交通機関や農業など様々な場所にネットワーク端末を設置し、作業や移動に必要なコストの最適化だけでなく、リアルタイムモニタリングのデータ共有などを行うことができる¹⁾²⁾。このネットワーク技術はIoTやM2Mと呼ばれ、モバイル・インターネット技術などと組み合わせ、幅広い用途への発展が期待されている。

一方、寒冷地では秋、冬につれ寒暖差が激しくなり、これは農業にとって致命的なことで、温度対策等が必須となる。本研究では、北東北など寒冷地における農作物生産での不利益な損失防止と、農作業の簡略化を目標とし、マイコンとセンサによる温床線の温度制御及びインターネットを介した遠隔監視の基礎検討を行った。具体的にはまず、センサ類とインターネットのインターフェイスにLinuxマイコンを用いてVPNプログラムを組み込み、次にUSBやI2Cなどのセンサ機器を接続し、VPNルータやセンサ情報取得装置を不要とした超小型かつ、安価で極めて運用コストの低く汎用性の高い遠隔監視と遠隔制御が可能な農作業支援システムを構築した。そして更に、土壌内の温床線を敷設することにより、先のマイコンを用いハウスで庫内温度を一定にするフェードバック制御システムも構築し、農業支援に適用が可能か検証した。

このシステムにより、寒冷地での農作物の生産時における不利益な損失防止や農作業の簡略化、移動に関わる時間やコストなどを抑えることが出来ると考える。

2. 開発したシステムの概要

本システムでは、Linux OSが動作するマイコンであるRaspberry Pi、SSR(ソリッドステートリレー)、温度湿度取得センサであるUSBRH、監視カメラとして利用するWEBカメラ及び発熱機から構成され、農耕地に設置された温床線を制御す

る為のLinuxマイコンであるRaspberry PiをVPNに参加させる。そして、交流100Vを制御し温床線のON・OFFのためにSSR(Solid State Relay)を導入し、ARMマイコンがセンサ情報を参照してSSRがON・OFFを繰り返し一定温度を保つ様にした。これら構成したシステムの全体構成は図1に示す。このシステムでは交流100Vを電力分配用の端子に接続し、制御側と動作側に電源を供給する。制御側では、AC-DC変換によってLinuxマイコンに電源を供給し、電源制御装置であるSSRを制御する。それに加え、マイコンに接続された各種センサより温度湿度情報の取得及び監視カメラ映像の取得を行う。次に、制御側では交流100V端子からSSRに接続し、マイコンにより、温度センサによる情報との比較から電圧のON・OFFの切替えにより温床線を制御して温度センサによる温度の測定値からハウス庫内の温度を制御する仕組みとなっている。

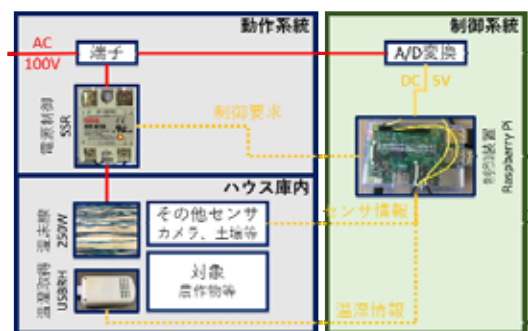


図1 構築したシステムの全体構成図

ここで、土壌内に敷設する発熱器には農業用発熱器として使われる温床線を選定した。温床線は、交流100Vを使用し尚且つ構造が単純であり、電力のON・OFFにより温度の制御が簡単に行えることからビニールハウス内の温度制御のため使用した。図2に示す様に線状になっており、今回使用するタイプは交流100V、長さ31m、消費電力250Wのものを用いた。これを実際にビニールハウス内での温度管理に応用する為、図3に示すとおり、プランタ(レリーフプランタ700, 29L 555mm×170mm)にホームセンタにて植物栽培用

の土を購入し、温床線同士が干渉しない様に割り箸を使い等間隔に配置した。更に、その上に土を被せ地中に敷設した。そして、保温対策のため図4の様に温床線を敷設したプランタの周りにビニールハウス(GRH-13S, 1540mm×718mm×600mm)を設置している。

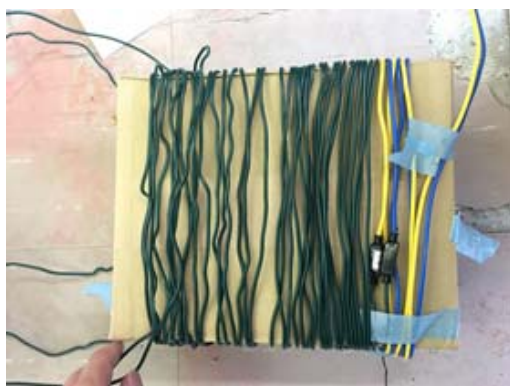


図 2 温床線



図 4 ハウスの設置



図 3 プランタへの温床線の敷設

3. システムの構築手順

そこで、図 1 の機能を有したシステムを実際に動作させる為、図 5 の制御用マイコンである Raspberry Pi に図 6 の温度湿度センサであるストロベリーリナックス社の USBRH と図 7 の SSR を接続し、温度センサによる情報を元に設置した温床線の温度制御を行った。具体的には、Raspberry Pi の GPIO 11 番ピンと GND の間で 0V 及び 3.3V の電圧を入力し、SSR に入力して温床線の ON・OFF を制御した。



図 5 Raspberry Pi



図 6 USB接続型温湿度センサ USBRH

この様な手順にて、マイコンによる温度湿度の時系列データの確認が出来た為、次に実際にハウス内の温度制御のため温床線に加える電力をON・OFFする為に交流100Vのスイッチ制御を行う。その際、用いたマイコンは交流100Vを直接制御ができない為、図7のソリッドステートリレー (SSR) にて、具体的にはRaspberry PiのGPIO端子であるGND及び11番ピン間に0V及び3.3Vの電圧出力を直接SSRに入力し出力側のSSRを制御した。その際、SSRを電氣的にON・OFFするため、Pythonを用いて状況に応じてON・OFFを繰り返すプログラムを作成した。このシステムの妥当性を確認するため、始めに安全性確認のため白熱電球を用い庫内の温度制御を確認した。その上で温床線を用いた電流のON・OFFを制御した。

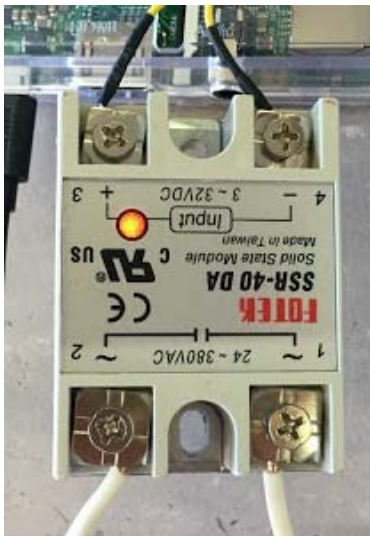


図 7 ソリッドステートリレー (SSR)

実際にマイコンと温度湿度センサ、SSR及び温床線による動作の内容を図8のフローチャートに示す。提案したシステムでは、まず温度湿度センサのUSBRHから温度情報の取得要求を行い、マイコンにてセンサ情報が正しく更新されているかを確認する。そして更新が正しく行われていない場合には、GPIOの出力を停止するように要求する。一方、更新が正しく行われている場合には、温度が例えば今回は摂氏30℃を目標とすれば、この値を超えているか超えていないかを条件処理する。摂氏30℃以上であればGPIOの出力を停止し、摂氏そして30℃を満たしていない場合には、異常動作がないかを判断した上でGPIOの出力を行いSSRの動作をONにする。また、プログラムに致命的なエラーが発生した場合(内部・外部問わず)、GPIOの出力を停止させ指定されたメールアドレスにエラー内容等の状況を送信する。これは、交流100Vを取扱う性質上の配慮から実施した。具体的なきっかけとしては、プログラムによる動作異常によって緊急停止した場合、GPIOの出力が保持され継続的にSSRを動作状態にする問題があった。この状態では、際限なくビニールハウス庫内の温度を上昇させ、危険な状態に陥る恐れがあったため、例外処理を施して異常動作時にははじめにGPIOの出力を停止させる様にした。更に、異常動作時には指定されたメールアドレスにエラー情報を送信するように設計し、使用ユーザに危機管理を促す仕組みを構築した。更に、センサ情報が更新されていない場合、実際とは異なる温度が制御プログラムに入力され際限なく温度を上昇させようとする場合や、永続的に停止した状態になる場合もあった。よって、これを解決するため、センサ情報が正しく更新されているかをハッシュ値で出力し、具体的には日付情報、時間情報、温度湿度情報をPythonのライブラリを使用して16進数32桁のハッシュ値に変換し、この値の変化を比較した。

最後に簡易操作に向けた取り組みとして、一般の農作物生産従事者にとってはOSのコマンドライン上での操作(CUI)は敷居が高い。この不便を

解消するためにWEB上でのGUIでの操作により、ブラウザ上で温度を調整できるようにした。

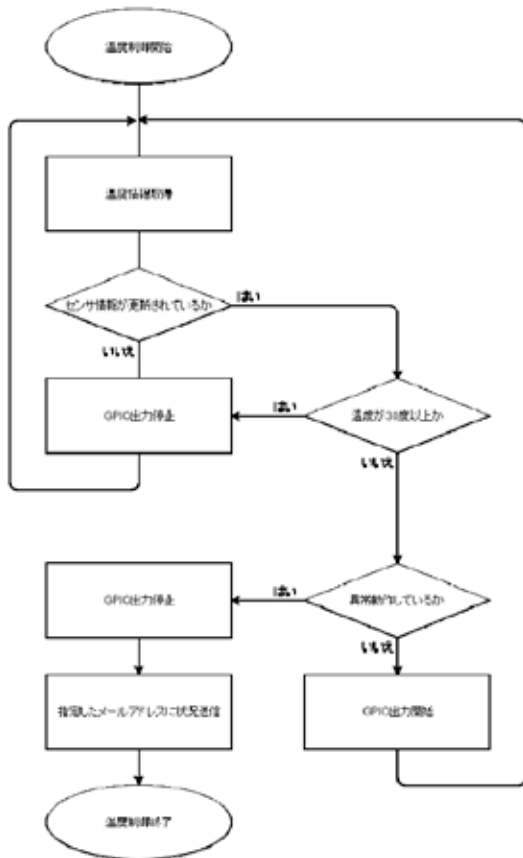


図 8 フローチャート図

図 9 が制御プログラムの一部であり、Python 言語で記述されている。なお、ここでは死活監視を適用しているものとして、GPIO は 11 番ピンでなく 12 番ピンに設定していることに留意する。

```
#取得PID明記
print("お使いのPIDは["+str(pid)+"]です。")

while True:

    try:

        #設定読み込み
        f = open('/var/www/html/pos/set.config')
        config = f.read()
        f.close()
        config_mail_rc, config_temp_rc = (config.split(';'))

        #読み込み 比較 A
        f = open('/home/pi/Data/100v.tx')
```

```
        usbrh_input1 = f.read()
        f.close()
        usbrh_insp1 = (usbrh_input1.split())

        #比較用時間
        time.sleep(2)

        #読み込み 比較 B
        f = open('/home/pi/Data/100v.tx')
        usbrh_input2 = f.read()
        f.close()
        usbrh_insp2 = (usbrh_input2.split())

        if len(usbrh_input1) == 0:
            print("INPUT#1 はキャンセルされました。")
            GPIO.output(12, False)
        elif len(usbrh_input2) == 0:
            print("INPUT#2 はキャンセルされました。")
            GPIO.output(12, False)
        else:
            usbrh_hash1 = usbrh_insp1[4]
            usbrh_hash2 = usbrh_insp2[4]
            if usbrh_hash1 != usbrh_hash2:
                usbrh_temp = float(usbrh_insp2[2])
                usbrh_humi = float(usbrh_insp2[3])
                config_temp = float(config_temp_rc)

                #温度一定値維持
                if usbrh_temp >= config_temp:
                    GPIO.output(12, False)
                else:
                    GPIO.output(12, True)

            else:
                GPIO.output(12, False)
```

except:

#例外処理（動作不具合の場合、安全に停止させるための部分）

```
barx = ("=====<<ERROR CODE>>=====")
GPIO.output(12, False)
print("異常動作によりプログラムを強制的に終了します。")
```

```
GPIO.output(12, False)
print("GPIO No.12 の動作が停止しました。")
```

```
#取得PID明記
print("お使いのPIDは["+str(pid)+"]です。")
```

```
#システムエラー報告
error_see = traceback.format_exc(sys.exc_info()[2])
error_trac = ("発生時刻:") + str(usbrh_insp2[0]) + ("
") + str(usbrh_insp2[1]) + ("
") + ("対象
PID:") + str(pid) + ("
") + str(barx) + ("
") + str(error_see) + ("
") + str(barx) + ("
") + ("usbrh_tx2 >>
") + str(usbrh_insp2) + ("
") + ("usbrh_tx1 >> ") + str(usbrh_insp1)
```

```
f = open('/home/pi/Data/error.log', 'w')
f.write(error_trac)
f.close()
#終了
```

図 9 制御プログラムの一部

更に、メールの設定プログラムと画面を図10および11に示す。

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>温度調整</title>
</head>
<body>
<h1><font color="#FA500"></font>現在の設定</h1>

<?php
$filename = 'set.config';
$fp = fopen($filename, 'r');
$txt = fgets($fp);
$drop = split(";", $txt);
echo 現在の緊急時送信先アドレス : $drop[0]. '<br>';
echo 現在の設定温度 : $drop[1]. '<br>';
fclose($fp);
?>

<h1><font color="#FA500"></font>温度調整指示メニュー</h1>
<form action="regist.php" method="post">
  緊急時送信先アドレス : <br />
  <input type="text" name="mail" size="30" value="<?php echo
$drop[0] ?>" /><br />
  選択温度 : <br />
  <select name="temp">
<option value="20.0">20℃</option>
<option value="21.0">21℃</option>
<option value="22.0">22℃</option>
<option value="23.0">23℃</option>
<option value="24.0">24℃</option>
<option value="25.0">25℃</option>
<option value="26.0">26℃</option>
<option value="27.0">27℃</option>
<option value="28.0">28℃</option>
<option value="29.0">29℃</option>
<option value="30.0">30℃</option>
<option value="31.0">31℃</option>
<option value="32.0">32℃</option>
<option value="33.0">33℃</option>
<option value="34.0">34℃</option>
<option value="35.0">35℃</option>
<option value="36.0">36℃</option>
<option value="37.0">37℃</option>
<option value="38.0">38℃</option>
<option value="39.0">39℃</option>
</select>
<br />

  <input type="submit" value="設定実行" />

</form>
</body>
</html>

```

図10 メールアドレス指定と温度選択画面

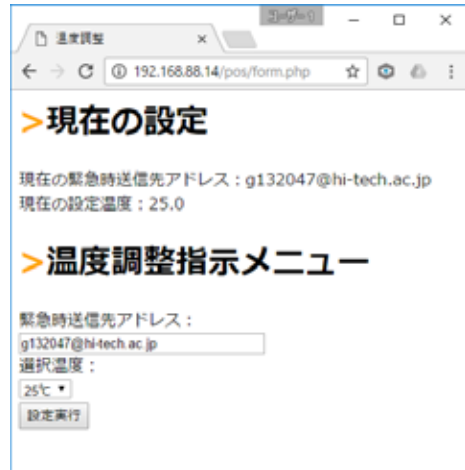


図11 メールアドレス指定と温度の設定画面

そして更に、実際に図12に示す様に、制御マイコンやSSRなどを防水ボックス内に導入し、小型化及び感電対策を行った。



図12 制御装置の防水ケースへの収納

そこで、これら一式を図5で示した防水ケースに格納し、実際にビニールハウス内の温度を一定に保てるかを確認した。ここで、提案システムでは制御マイコンを内包する制御装置を実際に遠隔地で制御可能にするため、Linux OSの動作するマイコンであるRaspberry PiをVPNクライアントであるHamachiに参加させる必要があった。このVPNサービスにRaspberry Piに参加させることで、VPNサービスに参加している同グループ内からインターネットを介し、場所を問わずどこからでも制御マイコンを監視、制御及び保守が可能と

なる。また、インターネットを介したVPN接続が切れて制御不能になることを回避する為、死活監視プログラム及びOSがハングアップした場合に再起動を行うwatchdogというアプリケーションも常駐させた。

そして更に、土壤に含まれる水分を検出するため、図13に示すDFRobot社の土壤センサSEN0114の追加を行った。しかし、Raspberry Piでは土壤センサから送られてくるアナログ数値を直接読み込むことが出来ないため、図14に示すMCP3425アナログデジタル変換(ADC)を仲介しつつRaspberry Piに数値を受け渡すシステムを構築し、土壤に含まれる水分量を取得した。

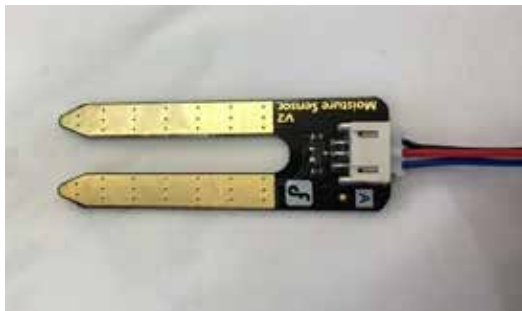


図 13 土壤センサ SEN0114

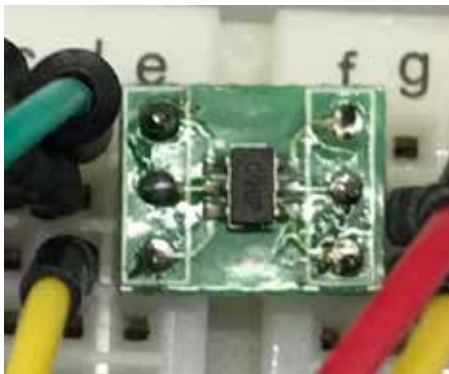


図 14 ADC 半導体集積回路 MCP3425

4. システムの動作検証

そこで構成したシステムにて、実際に温度が正しく制御できているかを確認した。実際の稼働中の様子を図15に示す。そして、WEB GUI画

面より摂氏25℃に指定し、ハウス庫内の温度を制御した結果を図16に示し、比較のためのハウス外温度を図17に示す。これにより、庫内温度はハウス外温度と比べ指定した任意の温度である摂氏25℃付近にまで制御されていることが分かった。なお、制御期間は平成29年1月24日 午前9:00より開始し、現在まで約2週間程度継続的に稼働することを確認できた。1日あたりの平均消費電力は2kWhであった。また、緊急時のメール通知が正しく行われているかを確認するため、実験的にエラーシグナルを送信した。図18では、実際にメールが受信した様子であり、異常動作の反応を確認することが出来た。更に、図19に示す通り、土壤センサから送られた水分検出率を時系列データで、濡らしたティッシュを土壤センサの上に被せティッシュが乾燥していく様子も観測できた。今後は実際の土壤における含水率と抵抗率との関係を明らかにする必要がある。



図 15 制御ハウス庫内の様子

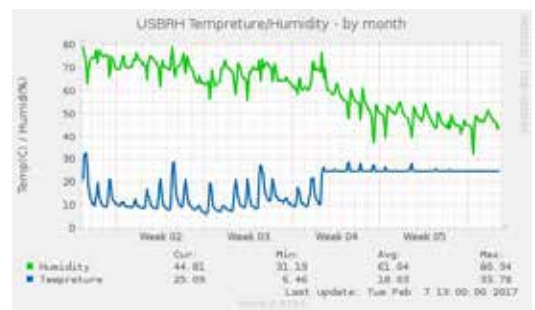


図 16 ハウス内の温度グラフ

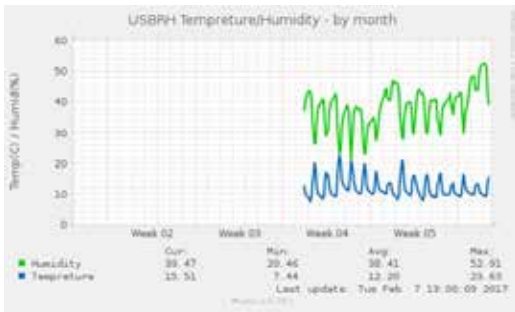


図 17 ハウス外の温度グラフ



図 18 制御対象のハウスの様子の確認

ルゴウの種を植えて、実際に植物が育つか検証した。その結果、種を蒔いてから図 21 に示す 4 日後に発芽を確認した。その後はハウス内の温度制御をしつつ、ソルゴーは順調に生育している。今後の課題として、土壌の水分を自動的に補給させる機能の実装が必要と考える。

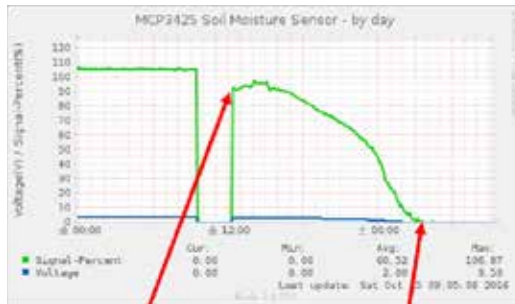


図 20 ハウス内での植物の育成状態

6. まとめ

本研究では、北東北の様な寒冷地における農業に対しての不利益な損失防止と、農作業の簡略化を目標とし、マイコンとセンサにより温床線の温度制御の為の基礎検討を行った。具体的にはセンサ類とインターネットのインターフェイスを有するLinuxマイコンを用いてVPNプログラムを組み込みUSBやI2Cなどのセンサ機器を接続し、VPNルータやセンサ情報取得装置を不要とした小型かつ安価で極めて運用コストが低く、汎用性の高い遠隔監視と遠隔制御が可能なシステムを構築し、更にビニールハウス庫内温度を一定にするフェードバック制御システムも構築して、農業支援に適用が可能なかを検証した。その結果、ハウス外温度に比べハウス内の温度は摂氏25℃付近に収まった。

今後の予定として、実際に図20及び図21に示すとおり実際に植物の育成として実用性に足りうるかを確認したい。更には、携帯回線網との組合せにより場所を問わず管理できるシステムを



水分を含むティッシュ



乾燥したティッシュ

図 19 土壌センサの時系列データ

5. 実際の栽培による検証

そこで、実際に図 20 に示すプランタ内に、ソ

構築していきたい。

参考文献

- 1) 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.
- 2) 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.
- 3) 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) K. Shibata and K. Hanada “Development of an Ultra-small Sensor Information Remote Monitoring System with an Embedded VPN and Linux Microcomputer Operation”, Proc. of International Conference on Engineering and Applied Science, ICEAS2015, Sapporo, Japan, 2015-7.
- 7) 成田博貴, 菊地桐吾, 柴田幸司 “Linux マイコンによる安価な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応用,” 2015 年度電気関係学会東北支部連合大会, 1D01, 2015-8.

要 旨

筆者の一人は以前、Linux マイコンである Raspberry Pi の多様なネットワーク機能を利用して、VPN 上に接続させセキュアな通信が可能となる仕組みを構築している。そして、このシステムが遠く離れた場所からもクライアントに接続することができ、温度湿度などのセンサ情報や映像情報の取得が可能となることを示した。そこで、これらのシステムを寒冷地での農業支援に応用するため、本研究ではネットワーク機能である VPN(仮想プライベートネットワーク)上で監視・制御できる農業支援システムの仕組みを構築した。そして、交流 100V で尚且つ構造が単純である汎用性の高い温床線を使用してビニールハウス庫内の温度を温める仕組みを構築した。

キーワード: 遠隔制御, 遠隔監視, ラズベリーパイ, 組み込みLinux