

LoRa を用いた圃場管理向け通信プロトコルの提案

寺田 恵太郎^{†1} 豊田 真治¹ 平田 忠明¹ 高田 裕也¹ 松本 恵子¹
袖 美樹子²

概要: 我々は農業従事者の作業負軽減や知識伝承を助けるため、フィールドサーバを用いた圃場管理システムの開発を行っている。本稿では、フィールドサーバの導入障壁である運用費用を削減する目的で、回線使用料が不要な無線ネットワークの構築を検討した結果を報告する。昨年 12 月に稲作圃場で基本通信特性調査を行った結果、通信距離および実圃場での活用容易性の観点から通信規格 LoRa が最適であると判断した。そこで LoRa を用いたネットワーク構築手法について検討を行った。フィールドサーバは圃場で動作させるため、低消費電力で稼働することが望まれる。これを実現する間欠動作通信プロトコルについて詳細に検討し、農業従事者の要求である田植えから稲刈りまでの 6 ヶ月間モバイルバッテリーで稼働可能な低消費電力プロトコルを開発した。提案したネットワーク構築手法を用いてネットワークを構築し、検証を行った。結果、親機とフィールドサーバ間でデータのやり取りが想定通りに行われていること、特にフィールドサーバからデータを受信できなかった場合においても、再送処理を行うことでデータを親機が受信することを確認できた。フィールドサーバの消費電力量を算出した結果、1 日あたり 75.36[mW]であり、理論上 995 日の稼働できることを確認した。農作業の負軽減のためには、フィールドサーバにかかるメンテナンスの頻度は低いことが望ましく、995 日の連続稼働という性能は稲作作業のモニタリングには十分な稼働時間である。

Proposal of communication protocol for field management using LoRa

KEITARO TERADA^{†1} MASAHARU TOYOTA¹ TADAAKI HIRATA¹
YUYA TAKADA¹ KEIKO MATSUMOTO¹ SODE TANAKA MIKIKO²

1. はじめに

近年、農業従事者の高齢化が進み、早急な知識や技術の伝承が必要とされている[1]. 我々はそれらを手助けするための稲作用フィールドサーバと、それに付随するアプリケーションの開発を行ってきた[2]. フィールドサーバは電源の無い圃場に設置されるため、田植えから稲刈りまでの 6 か月間をモバイルバッテリーで稼働することが望まれている。それを実現するためには、フィールドサーバの低消費電力化が必須である。

現在市販されている製品は、フィールドサーバから取得したセンサデータ等をクラウドに保存する際に、3G 回線が利用されている[3][4]. しかし、フィールドサーバの台数分、3G 回線の月額使用料が必要であり、運用コストを増加させるため、フィールドサーバの導入障壁となりうる。そこで我々は、運用コストの削減を行うために、回線使用料の必要がない、Wi-SUN, LoRa 等の基本通信特性を実圃場で調査した。結果、稲作圃場では LoRa が適していることが判明した[5][6][7].

また、電源の無い環境で 6 か月間稼働させるためには、センサデータ取得および通信を行っていない間、電源を OFF にする必要がある。そのため、間欠動作通信プロトコ

ルおよび時刻同期技術が必要である。時刻同期技術については、広く研究が行われている[8][9][10][11].

電波時計を用いた時刻同期手法が提案されている[8]. この手法は、時刻合わせを行うために 3 分間程度の時間を要する。そのため、稼働時間が長くなり、低消費電力の実現は困難である。また、受信した信号(タイムコード)をフィールドサーバ内で扱える形式への変換を必要とするが、メモリの少ない PIC マイコンにこの処理を行うプログラムは書き込むことができない。そのため、稲作用フィールドサーバに用いることは困難である。

GPS を用いて時刻同期を行う手法が提案されている[9]. この手法を用いるには、各フィールドサーバすべてに GPS 受信モジュールを搭載する必要があり、導入時における初期費用の増加につながる。このため、導入障壁を下げることを望まれる稲作用フィールドサーバに用いることは困難である。

RBS(Reference Broadcast Synchronization)は、無線ネットワークの時刻同期手法として提案されている[10]. ある任意のブロードキャストパケットを受け取ったノード同士で時刻情報を交換することにより時刻同期を行う。しかし RBS は受信ノード数の増加に比例して交換する情報量も増えるため、消費電力を増加させる問題がある。そのため、稲作用フィールドサーバに適用することは困難である。

RBS の改善方法として TPSN(Timing-sync Protocol for

¹ 金沢工業大学 ² 金沢工業高等専門学校
[†] b1415461@planet.kanazawa-it.ac.jp

Sensor Networks)が提案されている[11]。TPSNは、親機をトップとした木構造を形成し、親機(ルートノード)は子ノードと時刻同期を行う。その子ノードは、さらに下位の子ノードと時刻同期を行う。このように全体を通して時刻を同期する仕組みである。この手法では、時刻同期に長い時間を必要とするため、低消費電力の稼働を望まれる稲作用フィールドサーバには適さない方法である。

そのため我々は稲作用フィールドサーバ向けの LoRa を用いたネットワーク構築方法および間欠動作通信プロトコルの検討を行った[2]。間欠動作を行うためには複数のフィールドサーバの同時起動および、これにともなう同時送信を避けるための協調動作が必要である。時刻同期を行わず、フィールドサーバ内部のマイコンにより時間管理を行う場合、1 か月で最大 4,688 秒のずれが生じる。また我々のフィールドサーバに搭載できるリアルタイムクロック IC を複数検討した結果、IC によって異なるが、ずれが最小となる IC でも、1 か月に約±13 秒のずれを生じることが判明した[12]。6 か月間稼働させると、1 分以上のずれが生じることとなる。フィールドサーバは複数台存在するため、そのずれを考慮して通信プロトコルを作成した場合、6 か月間の稼働は困難である。

そこで我々は、これらの問題を解決する時刻同期手法の検討を行ったため報告する。提案手法はフィールドサーバが親機へセンサデータを送信したのち、これを受信した親機がフィールドサーバに時刻補正信号を送信する。これにより、時刻同期を行うものである。本論文の2章ではシステム構成、3章には通信プロトコルと時刻同期、4章には稼働試験および結果、最後に5章でまとめとする。

2. システム構成

2.1 システム全体概要

我々は農作業の負荷軽減に貢献する目的で、圃場の環境情報を各種センサによって取得および活用する圃場管理システムの開発を行っている[2][13][14]。圃場管理システムの全体構成図を図 2.1.1 に示す。本システムはフィールドサーバ、親機、クラウドサービスから構成されている。

フィールドサーバは圃場に設置し、センサデータを取得したのち、無線ネットワークを通じて親機へそのデータを送信する。親機は各圃場に設置されているフィールドサーバからのセンサデータを集約し、3G 回線や Wi-Fi を介してクラウドへ転送する。

クラウドサービスはスマートフォンやタブレット等のアプリケーションおよび Web ページで提供される。農業従事者に対して水位等の警告、作業計画の提案、作業記録の保存などの機能を提供する。

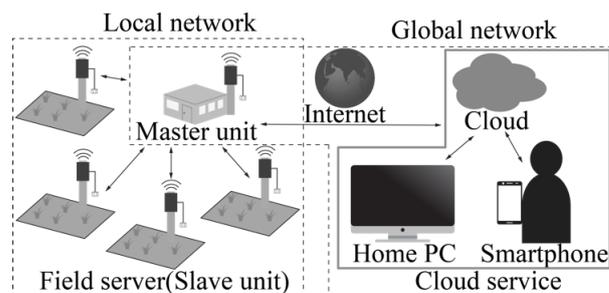


図 2.1.1 圃場管理システム全体イメージ図

フィールドサーバと親機間の通信に関しては、長距離通信が可能な LoRa を用いる。LoRa は我々が行った基本通信特性調査により、実用通信距離が 3,000~4,000m であることが判っている[7]。今回想定している石川県の圃場では、親機とフィールドサーバ間の直線距離は約 3,000m 以内である。そのため我々はフィールドサーバと親機間で直接通信が可能な LoRa を採用した。

想定した石川県内の農業法人の各圃場と事務所の位置関係を図 2.1.2 に示す。A, B, C, D, E, F, G が各圃場の位置を表し、この地点にフィールドサーバを設置する。P は事務所の所在地を表し、事務所に親機を設置する。フィールドサーバ A~G と親機 P 間の直線距離は、AP 間 397m, BP 間 923m, CP 間 943m, DP 間 684m, EP 間 1,150m, FP 間 1,440m, GP 間 1,910m である。



図 2.1.2 各圃場と事務所の位置関係

2.2 フィールドサーバおよび親機概要

フィールドサーバと親機により構成される無線ネットワークの詳細を説明する。

フィールドサーバの構成を図 2.2.1 に示す。フィールドサーバは、モバイルバッテリー、電源 ON/OFF 回路、AVR マイコン、LoRa 無線モジュール、各種センサ、SD カードモジュールで構成されている。フィールドサーバは、圃場に設置することから電源供給が困難であるため、モバイルバッテリー(リチウムイオン二次電池)によって電源供給が行われる。低消費電力を実現するため、電源 ON/OFF 回路によって1時間のうち、数十秒間のみ稼働する。無線通信やセンサの制御は、AVR マイコンによって行われる。センサは5種類搭載し、温度、湿度、水位、土壌温度、土壌水分度を計測している。

SD カードにはセンサデータをタイムスタンプとともに保存する。これは親機にセンサデータを送信できない場合や親機からの時刻補正信号を受信できない場合においても、センサデータを確実に保存するための機能である。親機との時刻同期に失敗した場合は、SD カードに記録されている前回記録時のタイムスタンプに 3,600 を加算し、今回記録用のタイムスタンプを求める。

電源 ON/OFF 回路は PIC マイコンと FET により構成されており、電源 ON/OFF 回路に電力線で繋がるモジュールへの電源供給を制御する。PIC マイコンは GPIO ピンで HIGH/LOW 出力することによって FET を制御する。PIC マイコンは電源制御に必要な時間を内部クロックのタイマ割り込みを用いて計測し、制御を行っている。

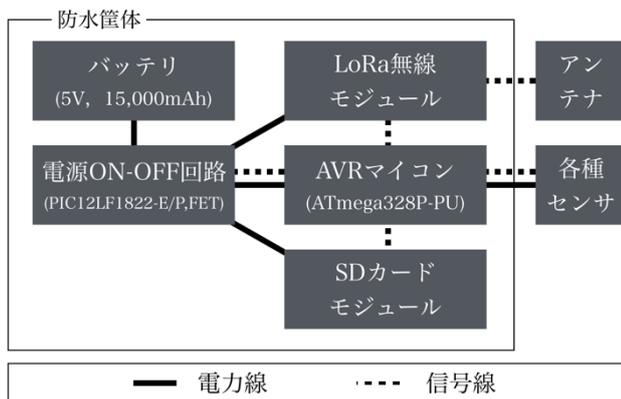


図 2.2.1 フィールドサーバの構成

親機の構成を図 2.2.2 に示す。親機は、Raspberry Pi、送信用 LoRa 無線モジュール、受信用 LoRa 無線モジュール、3G ドングルで構成されている。

フィールドサーバは初回電源投入時、時刻を保持していないため待機することなく起動する。そのため親機は常に受信状態を維持し、農業従事者がいつフィールドサーバを設置してもフィールドサーバからの通信に対応できること

が望ましい。また、LoRa 無線モジュールには受信モードと送信モードが存在し、モードの切り替えには時間を要する。そのため、親機では受信専用と送信専用の LoRa 無線モジュールを2個搭載することにより、送受信の待機時間削減と常時受信状態の維持を実現した。

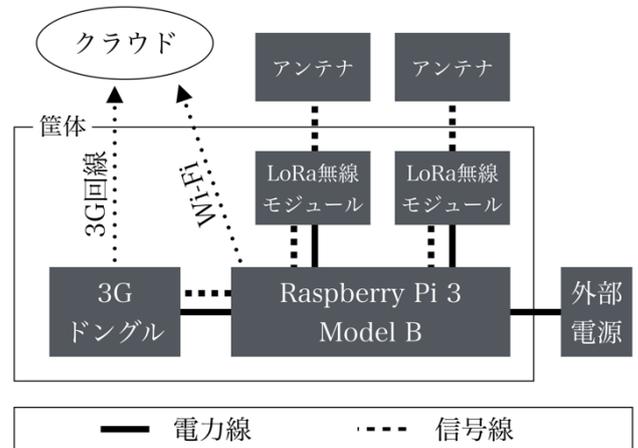


図 2.2.2 親機の構成

3. 通信プロトコルと時刻同期

3.1 通信プロトコル概要

フィールドサーバは PIC 内時刻が 3,600 (0) 秒の時に起動し、センサデータの取得を行う。そのセンサデータを親機に送信し、親機がセンサデータの受信に成功した場合、親機はフィールドサーバに補正時刻を送信する。フィールドサーバは補正時刻を受け取り、AVR マイコンが PIC マイコンに補正時刻を転送する。PIC マイコンはこの補正時刻をもとに PIC 内時刻を補正する。転送が終了した時点、もしくは後述の1時間あたりの稼働時間が経過した時点で AVR など電源 ON/OFF 回路以外は電源を OFF にし、次回起動時まで待機する。

通信に用いるフレームフォーマットを表 3.1.1、表 3.1.2、表 3.1.3 に示す。表 3.1.1 は共通フレームフォーマットである。送信先、送信元およびペイロードで構成される。表 3.1.2 はセンサデータ送信のフォーマットである。これはフィールドサーバから親機へ送信される信号である。現在搭載しているセンサは5種類あるため、センサデータ1~5までにそれぞれ2Byteでデータを格納できるフォーマットを定義している。またセンサの種類が増えた場合には、センサデータ送信のフォーマットに、2Byte分ずつ追加することで対応できる。表 3.1.3 は時刻補正信号のフォーマットである。これは親機からフィールドサーバへ送信される信号である。親機が取得したタイムスタンプとそれをもとに計算した各フィールドサーバへの補正時刻が格納される。

表 3.1.1 共通フレームフォーマット

送信先	送信元	ペイロード
1Byte	1Byte	可変

表 3.1.2 センサデータ送信フォーマット(ペイロード)

センサ データ 1	センサ データ 2	センサ データ 3	センサ データ 4	センサ データ 5
2Byte	2Byte	2Byte	2Byte	2Byte

表 3.1.3 時刻補正信号フォーマット(ペイロード)

タイムスタンプ (UNIX 時間)	補正時刻
4Byte	2Byte

通信プロトコルのシーケンス図を図 3.1.1 に示す。図 3.1.1 は、フィールドサーバ 3 台であり、すべての通信において再送が発生しない場合の例である。最初にフィールドサーバ A(以下、FS-A)が起動する。起動後はセンサが安定するまで固定時間待機する。その後 FS-A はセンサデータを取得し、表 3.1.2 のセンサデータ送信フォーマットに従って送信パケットを生成したのち、センサデータを親機へ送信する。フィールドサーバは LoRa 無線モジュールを 1 個のみの搭載であるため、送信モードから受信モードへ切り替えを行う。この切り替えには数秒を要する。受信モードに切り替えた後は、設定されたタイムアウト時間まで受信待機し、親機から送られる時刻補正信号の受信を行う。

親機は FS-A からのセンサデータを受け取ると時刻補正信号を FS-A に対して送信する。FS-A はその時刻補正信号をもとに PIC 内時刻を補正する。補正後は再送可能時間の途中であっても、スリープ状態へ移行する。フィールドサーバが時刻補正信号の受信もしくは親機が送信に失敗した場合は再送を行う必要があるため、再送可能時間を設けている。この一連の動作をフィールドサーバ B(FS-B)、フィールドサーバ C(FS-C)も同様に行う。

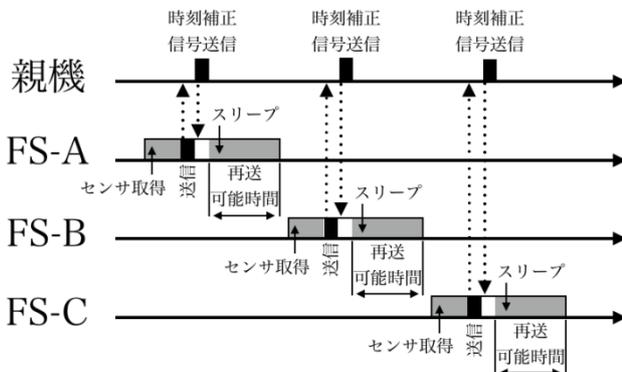


図 3.1.1 通信プロトコルシーケンス図

1 時間あたりの稼働時間は、式(3.1.1)から求められる。この稼働時間は事前に AVR マイコンに書き込まれ、AVR マイコンに電源が投入される毎に PIC マイコンへ送られる。各フィールドサーバの稼働時間は、

$$\begin{aligned} \text{稼働時間} = & \text{センサ安定待機および取得時間} + \\ & \text{センサデータ送信時間} + \\ & \text{モード切替時間} + \\ & \text{受信待機時間} + \\ & \text{受信時間} + \\ & \text{再送可能時間} \end{aligned} \quad (3.1.1)$$

となる。センサ安定待機および取得時間は、電源投入後の電圧が安定し、センサ取得が可能になるまでの固定待機時間およびセンサデータを測定する時間である。センサデータ送信時間は親機にセンサデータを送信する時間である。モード切替時間は、LoRa 通信モジュールを送信モードから受信モードへ切り替える固定時間である。受信時間は親機から時刻補正信号を受信する時間である。再送可能時間はセンサデータを親機に送信できない場合、再送処理を行える時間である。

3.2 再送発生時の動作概要

フィールドサーバは時刻補正信号を受信できず、かつ受信待機時間が経過した場合、タイムアウト動作を行う。タイムアウト後は、フィールドサーバは 0.1 から 5.0 秒までランダム秒待機したのち、再送を行う。この場合のシーケンス動作を図 3.2.1 に示す。図 3.2.1 において、フィールドサーバ C(FS-C)がタイムアウトとなり、再送動作を行っている。タイムアウトの発生要因は、

- 1) 電波減衰や復調できない程のノイズによって、親機がフィールドサーバからの通信を受信できない場合
- 2) 複数のフィールドサーバの送信時刻の重なりにより、送信データに衝突が生じた場合

などが考えられる。その際、親機はフィールドサーバに対して時刻補正信号を送らないため、フィールドサーバの受信待機はタイムアウトとなる。タイムアウトとなったフィールドサーバは再送を行うが、初回電源投入のフィールドサーバが行う通信との再衝突を防ぐため、ランダム秒の待機時間を設けている。

フィールドサーバは再送を行なったのち、送信モードから受信モードへ切り替え、受信待機を行う。この動作をフィールドサーバは、親機からの時刻補正信号を受け取れるまで続ける。ただし、他のフィールドサーバと通信の衝突を避けるため、他のフィールドサーバの稼働開始時刻になった場合、強制的に電源を OFF にする。他のフィールドサーバの稼働開始時間までに安定して再送可能な時間を考慮し、1 回の再送を行う。

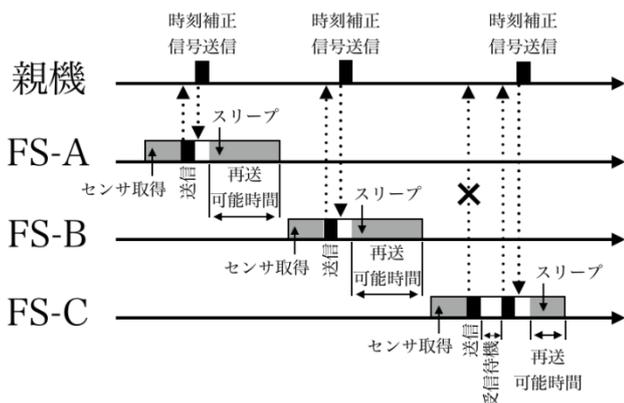


図 3.2.1 センサデータ衝突時のシーケンス図

3.3 時刻補正信号

親機は常時受信待機を行なっているため、フィールドサーバは任意のタイミングでセンサデータを送信できる。親機は事前に NTP サーバと時刻同期を行い、正確な時刻を取得している。親機はこの時刻によって時刻補正信号を生成し、フィールドサーバからセンサデータを受信したのち、時刻補正信号を送信する。表 3.1.3 の時刻補正信号のフォーマットに示すとおり、タイムスタンプは UNIX 時間が 4Byte に入る。これは AVR マイコンの EEPROM や SD カードへセンサデータの書き込みに使用する。補正時刻は、PIC 内時刻を補正する際に利用するため、現在時刻から各フィールドサーバ用にずらした時刻を 0~3599 秒として 2Byte で入れる。通信エラー検出および訂正に用いられる CRC 等は、LoRa 通信モジュールが追加するため、フォーマットでは定義していない。

図 3.3.1 にフィールドサーバと親機間の時刻同期の仕組みを示す。親機は NTP サーバによって得られた正確な時刻を補正時刻形式でフィールドサーバに送信する。フィールドサーバは補正時刻を受信すると、AVR マイコンを経由し、電源 ON/OFF 回路内の PIC マイコンに転送する。補正時刻は他のフィールドサーバと同時に起動することを防ぐため、現在時刻を適切にずらしたものである。この時刻をずらす計算式を式(3.3.1)に示す。式(3.3.1)に関して、各フィールドサーバは一意に割り当てられたフィールドサーバ識別子(以下FSID)を持っており、現在時刻 t_1 から $FSID \cdot 30$ を引くことにより、フィールドサーバへ送信する補正時刻 t_2 が求められている。またフィールドサーバの電源投入時間によっては、時刻 t_2 が負の値となるが、この場合は t_2 に 3,600 を加算する。

$$t_2 = t_1 - 30 \cdot FSID \quad (3.3.1)$$

FSIDは 0x00 から 0x77 の範囲で使用でき、FSID の昇順に起動を行う。FSIDを用いることにより各フィールドサーバの同時起動を防ぎ、フィールドサーバの送信信号を衝突さ

せないようにしている。

電源 ON/OFF 回路を制御する PIC では常に内部のクロックで 0 から 3,599 秒をカウントしている。PIC 内時刻が 3,600(0)秒になると、AVR マイコンへ電源を供給する。フィールドサーバは親機から時刻補正信号を受け取ると、PIC 内時刻を親機から送信された補正時刻に補正し、カウントを続ける。これを行うことで、フィールドサーバと親機で時刻同期を行っている。2 回目以降の起動においても PIC 内時刻が 3,600(0)秒に達することで AVR マイコンへ電源供給を行う。

親機は通信エラーなどによりフィールドサーバからのデータを正常に受信できなかった場合、時刻補正信号を送信せずにセンサデータの受信待機状態を維持する。

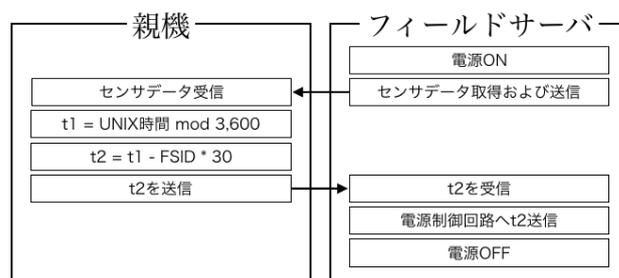


図 3.3.1 親機とフィールドサーバ間の時刻同期の仕組み

4. 稼働試験

4.1 通信プロトコルの検証方法

LoRa および考案した通信プロトコルを利用し、親機とフィールドサーバ間で想定通りの動作を行うか確認する目的で、7 日間の稼働試験を行った。

検証内容は、

- 1) 親機がフィールドサーバへ時刻補正信号をフィールドサーバの受信待機時間内に返送可能かの確認
- 2) 時刻補正信号を受け取ったフィールドサーバは、即時にスリープ状態へ移行するかの確認
- 3) 時刻補正信号によって各フィールドサーバが適切にタイミングを変更し、指定した時刻に起動するかの確認

である。

通信プロトコルの検証では、フィールドサーバの台数を我々が対象としている圃場例と同数の 7 台で行った。フィールドサーバと親機間の距離は、親機を中心とし、全フィールドサーバを半径 1m 以内に設置した。フィールドサーバから親機へ送られたセンサデータは、検証用クラウドへ保存した。

また 7 日間の稼働試験に加え、フィールドサーバと親機間の通信に失敗した際、想定した再送動作を、各フィールドサーバに割り当てられた稼働時間内に行えるかを検証す

る。検証方法は、フィールドサーバが1回目のセンサデータを送信する際に、親機は受信待機をせず、フィールドサーバの受信待機をタイムアウトさせる。その後、フィールドサーバが2回目のセンサデータを送信する際に、親機は受信待機および時刻補正信号をフィールドサーバへ送る。フィールドサーバがこれを受信し、スリープ状態へ移行するか確認する。

本プロトコルは圃場管理を対象としており、各圃場から収集するセンサデータの時刻に数秒程度のずれは許容する。しかし、各フィールドサーバの起動時刻に大きなずれが生じた場合は、送信するセンサデータの衝突や親機からの時刻補正信号を受け取れないなどの問題を生じる。そのため、各フィールドサーバに指定した一定の時刻に起動するかを検証した。検証方法は各フィールドサーバのFSIDが0x02のフィールドサーバから0x08の順序で通信を行う。フィールドサーバはそれぞれ30秒間隔で起動させる。親機がセンサデータを受信した時刻をもとに72時間分の平均誤差の算出を行い、それぞれ平均最大誤差[s]と平均最小誤差[s]を求めた。

4.2 通信プロトコルの検証結果

表4.2.1に通信プロトコルの検証結果を示す。フィールドサーバの稼働日数は7日間であり、センサデータ取得および送信は1時間間隔である。フィールドサーバが親機へデータを送信した回数は、フィールドサーバ7台、1時間間隔で7日間のため、1,176回である。これに加え、フィールドサーバが親機から時刻補正信号を受信できず、再送を行った回数は9回である。そのため総センサデータ送信回数は1,185回となる。再送を行った9回のうち、9回すべてが再送1回目で時刻同期を完了した。

また再送動作の検証では、フィールドサーバが再送処理でのセンサデータ再送を行ったのち、親機からの時刻補正信号を受信し、スリープ状態へと移行した。この動作がシーケンス通りに行われることを確認した。よって問題なく動作していると判断できる。

表 4.2.1 通信プロトコル検証結果 (7日間稼働)

種別	回数
センサデータ送信	1,185
再送回数	9
時刻同期完了(再送1回目)	9

各フィールドサーバに指定した一定の時刻に起動するかを検証した結果を表4.2.2に示す。1時間間隔で行われるすべてのフィールドサーバの通信において、各フィールドサーバに与えられた想定起動時刻と実際に起動した時刻の差を絶対値で調べ、その平均最大・最小誤差を求めた。

結果は平均最大誤差が5.277秒、平均最小誤差が4.041秒であった。このことより、各フィールドサーバの起動タイミングはおおよそ5~6秒のずれが生じると判断できる。時刻同期を用いることで誤差を小さく抑えられているが、少なからず誤差は存在している。

これらの誤差が生じる要因として、電源ON/OFF回路に用いているPICマイコンにおけるクロックの誤差、LoRa通信モジュールの個体差による送信時間のぶれ、CRCによる誤り訂正時間、LoRa変調および復調にかかる誤差などが挙げられる。

表 4.2.2 各フィールドサーバの平均誤差算出結果

	平均誤差 [s]	平均最小 誤差[s]	平均最大 誤差[s]
1日目の送信	4.497	4.039	6.236
2日目の送信	4.400	3.929	4.717
3日目の送信	4.433	4.157	4.877
平均値	4.443	4.041	5.277

PICマイコンの誤差に関しては、測定することが可能であるため検証を行った。PICマイコンを1時間計測し、実際の時刻とのずれを誤差とした。この計測をPICマイコン7台(A~G)で12時間分計測し、PICマイコンごとに平均誤差を算出した。その結果を次の表4.2.3に示す。表より、最大で誤差が6.512秒である。このことから各フィールドサーバの誤差を招く要因としては、PICマイコンの誤差が大半であることがわかる。

しかし、今回提案する圃場管理用通信プロトコルにおいて、この誤差は他のフィールドサーバとの衝突が起きない範囲内である。そのため、この通信プロトコルにおいて時刻同期は効果的であり、使用時間に比例する時刻の誤差増加を減少できていることが確認できた。

表 4.2.3 1時間計測した際のPICマイコンの平均誤差

PICマイコン	PIC内時刻[s]	平均絶対誤差[s]
A	3,605.613	5.613
B	3,603.962	3.962
C	3,598.473	1.526
D	3,594.136	5.863
E	3,598.782	1.217
F	3,597.512	2.487
G	3,593.487	6.512

4.3 消費電力の評価

通信プロトコルの検証とは別に消費電力を検証するための稼働試験を行った。検証の目的は、本システムが要求されている期間、運用可能であることを確認するためである。

検証内容は、各プロセスでの消費電流をテスターによって測定し、理論的に消費電力を算出することにより要求期間の連続稼働が可能かを確認することである。測定結果を表 4.3.1 に示す。

表 4.3.1 消費電力検証結果

	時間 [s]	電圧 [V]	電流 [mA]	消費電力 [mW/h]
センサ安定 待機・取得	7.0	5.0	43.90	0.43
センサデータ 送信	1.0	5.0	157.30	0.22
モード切替	3.0	5.0	37.40	0.16
受信待機	3.5	5.0	48.80	0.24
受信	1.5	5.0	48.60	0.10
スリープ時間	3,582.0	5.0	0.40	1.99
			合計	3.14

式(4.3.1)にシステム稼働時における電力量 W_N [mWh]を求める式を示す。ここで、 V_1 はフィールドサーバの定格電圧[V]、 I_a および t_1 はセンサ安定待機時とセンサ取得時の電流[mA]および時間[s]、 I_b および t_2 はセンサデータ送信時の電流[mA]および時間[s]、 I_c および t_3 はモード切り替え時の電流[mA]および時間[s]、 I_d および t_4 は受信待機時の電流[mA]および時間[s]、 I_e および t_5 は受信時の電流[mA]および時間[s]を示している。

$$W_N = V_1 \{ (I_a \cdot t_1) + (I_b \cdot t_2) + (I_c \cdot t_3) + (I_d \cdot t_4) + (I_e \cdot t_5) \} \quad (4.3.1)$$

式(4.3.2)にシステムスリープ時における電力量 W_S [mWh]を求める式を示す。ここで、 I_g はスリープ時の電流[mA]を示している。

$$W_S = V_1 \cdot I_g \{ 3600 - (t_1 + t_2 + t_3 + t_4 + t_5) \} \quad (4.3.2)$$

式(4.3.3)に稼働時間 T_h [h]を求める式を示す。ここで、 V_2 はモバイルバッテリーの電圧[V]、 Q はモバイルバッテリーの電気量[mAh]を示している。分子の $V_2 \cdot Q$ によってモバイルバッテリーの電力量[mWh]を求め、分母の W_N および W_S でフィールドサーバが消費する電力量[mWh]を求めている。

$$T_h = 3600 \cdot \frac{V_2 \cdot Q}{W_N + W_S} \quad (4.3.3)$$

稼働日数の理論値を式(4.3.1)～(4.3.3)と表 4.3.1 より求めた。1時間あたりの消費電力量は 3.14[mWh]であり、1

日あたり 75.36[mWh]消費する。今回想定したモバイルバッテリーの容量は 75,000[mWh]であることから、995 日程度稼働できることが分かった。この日数は理論値であるが、モバイルバッテリーの自然放電等を考慮しても農業法人の要求である 6 か月間の稼働は可能であると考えられる。

4.4 通信プロトコルと消費電力の考察

今回考案した通信プロトコル、特に時刻同期の効果について考察する。従来我々が開発したフィールドサーバでは、同様のハードウェア構成で時刻同期を行わない通信手法を用いている[2]。この手法は親機からすべてのフィールドサーバに対して、データ受信確認およびスリープ命令を兼ねたブロードキャスト信号を一定間隔で送る。ブロードキャスト信号を送る時間の間隔は、フィールドサーバにおける時刻の誤差を考慮して設定しなければならない。そのため、間隔を広く設定する必要がある。また、親機が管理するフィールドサーバの台数増加に比例して時間の間隔を広く設定する必要がある。これは、台数増加による時刻の揺れをすべて許容しなくてはならないためである。我々の想定している圃場例と同数の 7 台の場合、フィールドサーバ 1 台あたりの稼働時間は最小 48 秒で、消費電力は 6.10[mWh]であった。時刻同期を行った本提案の通信プロトコルでは、稼働時間は約 16 秒で消費電力は 3.14[mWh]であった。

従来の通信手法に対して、今回考案した通信プロトコルでは時刻同期を行うことにより、フィールドサーバ 1 台あたりの稼働時間を最大 30 秒に短縮でき、約 2 倍の効率で低消費電力の実現に成功した。

5. まとめ

本稿では、田植えから稲刈りまでの 6 ヶ月間稼働するフィールドサーバ開発と LoRa を用いた圃場管理向け通信プロトコルについて述べた。特に時刻同期を行う通信プロトコルの安定性について述べた。提案した通信プロトコルを用いてネットワークを構築し、検証を行った。その結果、センサデータ送信は 1,185 回行われ、そのうち 9 回が再送 1 回目でも成功した。そのため 1 回目の通信に失敗した場合でも、すべてのフィールドサーバからセンサデータの受信に成功することが確認できた。

フィールドサーバの消費電力については、1日あたり 75.36[mWh]であった。よって、理論上 995 日の連続稼働が可能であることを確認した。

各フィールドサーバの起動タイミングとしては、おおよそ 5～6 秒のずれが生じる。しかし、このずれによって他のフィールドサーバに干渉することはない。そのため、この通信プロトコルにおいて時刻同期は効果的であり、稼働時間に比例する誤差増加を減少することができた。

これらのことから、本フィールドサーバは安定して長期

間の稼働が可能であるため、圃場管理システムに有効である。そのため、農作業の負荷低減に本システムは非常に期待できる。

しかし考案した通信プロトコルでは、同一の周波数で管理できるフィールドサーバの台数に制約があり、120 台を超える接続ができない。そのため、考案した通信プロトコルを拡張し、マルチホップに対応することで接続台数の上限を改善する必要がある。

参考文献

- [1] 農林水産省: 農業労働力に関する統計, 農林水産省(オンライン), 入手先<<http://www.maff.go.jp/j/tokei/sihyo/data/08.html>> (参照 2017-04-18).
- [2] 豊田 真治, 寺田 恵太郎, 高田 裕也, 平田 忠明, 袖 美樹子: LoRa を用いた圃場管理用ネットワーク構築手法の検討, 電子情報通信学会技術研究報告, NS2017-11, pp. 61-66.
- [3] 富士通: FUJITSU Intelligent Society Solution 食・農クラウド Akisai(秋彩), 富士通(オンライン), 入手先 <<http://jp.fujitsu.com/solutions/cloud/agri/>> (参照 2017-05-03).
- [4] ベジタリア株式会社: スマート農業における水稻の水位・水温を計測する水田センサ PaddyWatch, ベジタリア株式会社(オンライン), 入手先 <<http://field-server.jp/paddywatch/>> (参照 2017-05-03).
- [5] Wi-SUN Alliance: Home - Wi SUN Alliance, Wi-SUN Alliance(online), from<<https://www.wi-sun.org/index.php/en/>>(accessed 2017-05-03).
- [6] LoRa Alliance: Home, LoRa Alliance(online), from<<https://www.lora-alliance.org/>>(accessed 2017-05-03).
- [7] 川上 悠太, 豊田 真治, 寺田 恵太郎, 袖 美樹子: Sub-GHz 無線を使用した圃場間通信の検討, 電子情報通信学会技術研究報告, NS2016-138, pp.107-112.
- [8] 渡辺 貴文, 森戸 貴, 南 正輝, 森川 博之: 電波時計を用いた同期型バッテリーレス無線センサネットワークの設計と実装, 情報処理学会研究報告モバイルコンピューティングとユビキタス通信, vol.2007, No.44, pp.113-118(2007).
- [9] ポジション株式会社: 時刻基準としての GPS 受信について, ポジション株式会社(オンライン), 入手先<<http://www.posit.co.jp/mission/pdf/1998.6.pdf>> (参照 2017-05-03).
- [10] J. Elson, L. Girod and D. Estrin: Fine-Grained Network Time Synchronization using Reference Broadcasts, Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI' 02), Boston, Massachusetts (2002).
- [11] S. Ganeriwal, R. Kumar and M. B. Srivastava: Timing-sync Protocol for Sensor Networks, Proceedings of the 1st ACM Conference on Embedded Network Sensor Systems (SenSys'03), Los Angeles, California (2003).
- [12] セイコーエプソン株式会社: RX-8025SA | RTC (I²C) リアルタイムクロック | 製品情報 | エプソンデバイス, セイコーエプソン株式会社(オンライン), 入手先<<http://www5.epsondevice.com/ja/products/i2c/rx8025sa.html>> (参照 2017-05-03).
- [13] Takashi Furuta, Hideto Nakagawa, Taichi Kitamura, Yuta Kawakami, Kiyokazu Kurosawa, Kazuma Kogami, Mikiko Sode Tanaka: Agricultural support system equipped with short-range wireless communication system, 2016 IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development, pp.58-64 (online), DOI:10.1109/TIA R.2016.7801214(2016).
- [14] Yuta Kawakami, Takashi Furuta, Hideto Nakagawa, Taichi Kitamura, Kiyokazu Kurosawa, Kazuma Kogami, Naoto Tajino, Mikiko Sode Tanaka: Rice Cultivation Support System Equipped with Water-level Sensor System, The 5th IFAC Conference

on Sensing, Control and Automation Technologies for Agriculture (2016).