

# データ通信機能に電源供給・回線復旧・基準信号供給機能を 加えたI2Cベースの近距離有線ネットワークの提案

大野 浩之<sup>1</sup>

**概要:** 著者の研究グループでは、IoT デバイスが多数接続されたネットワークのための情報セキュリティ確保に関する検討を重ね、その過程で Raspberry Gate や Raspberry Guardian を提案し、設計実装や評価をさまざまな視点から進めているが、IoT デバイスを前提としたネットワークの物理層の構成については、無線接続を中心にすでにさまざまな方法が提案され実現していることもあり、これまで特段の議論をしてこなかった。このような経緯はあるものの、著者は以前より実験室や小規模オフィス程度の室内環境であれば、配線の手間はあっても、安全性、確実性、可用性の点で有線接続に利がある場面が多いと考えており、I2C を好んで採用してきた。しかし、I2C は本質的にロックアップしやすいという問題を抱えている。そのため、ロックアップから復旧する統一的なしくみの導入を行い、さらに、有線接続の利を生かした電源供給等も実現しつつ、絶縁技術を用いて安全性と可用性も向上させたいと考えた。本報告では、これらの要求を満たした上で、さらに高精度の時間信号の提供機能や構成管理機能等も付加した、「既存の I2C バスに 6 つの機能を付加した近距離有線接続手法」を提案する。

## I2C Based Wired In-house Network with Power Supply, Lockup Recovery, Precise Base Clock Supply Functions

HIROYUKI OHNO<sup>1</sup>

### 1. はじめに

近年、小型で単機能なセンサやアクチュエータ等の機材に通信機能を持たせ、相互に通信させたり既存の情報通信機器と通信したりすることが安価に簡単に実現できるようになった。このような情報通信の世界を指して IoT (Internet of Things, もののインターネット) という用語を使うことが増えている。

著者の研究グループでは、IoT 環境で使われる機器 (以下 IoT デバイス) が今後広く普すと考え、このような時代に資する、「多数の IoT デバイスが接続されたネットワーク環境」の情報セキュリティ確保を検討している。その過程で、すでに Raspberry Gate や Raspberry Guardian を提案し、設計実装および評価を続けている [1]。本報告ではこの提案に基づく Raspberry Guardian Model モデルを

RG モデルと略記する \*1。

RG モデルでは、セキュリティゲートウェイである Raspberry Gate の内側に、Raspberry Garden があり、さらにその内部には、数個から数十個の IoT デバイスからなる小規模な IoT ネットワーク (Raspberry Patch) がいくつか存在する。Raspberry Patch は OSI7 階層モデルの第 3 層としては単一のネットワークであり、その主体である IoT デバイスは IPv4 または IPv6 で相互にあるいは上流に位置する Raspberry Gate 等の装置と通信できる。著者らは Raspberry Patch の構成方法についてはこれまで詳しい議論をしてこなかった。しかし 2016 年初頭からは、Raspberry Patch で用いる IoT デバイスはその内部で用いる I2C ベースの有線接続方式について検討を開始し、本報

<sup>1</sup> 金沢大学 総合メディア基盤センター  
Kanazawa University, Kanazawa, Ishikawa 920-1192, Japan

\*1 文献 [1] の 2 節において “Raspberry Field” とした範疇は、その後 “Raspberry Patch” に名称変更しており、本報告でも “Raspberry Patch” を用いている

で述べる 6 つの追加機能の提案に至った。

## 2. 背景

RG モデルでは、IoT 環境における諸機能を図 1 のような「木いちごを栽培する農園」のアナロジーで表現している。

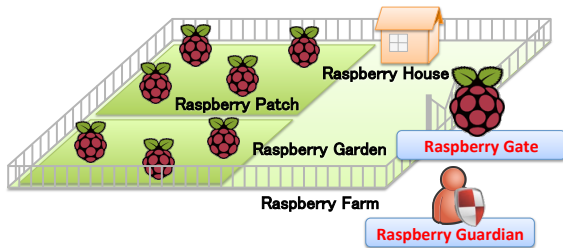


図 1 RG モデルにおけるアナロジー

このアナロジーのうち本報告に関連する概念には以下がある。これらの項目は RG モデルにおける概念を示し、括弧内は何のアナロジーかを示す。

### Raspberry Farm (農園)

ある者が所有し、特定の目的のために構築して管理運用する IoT ネットワークの全体。Raspberry Garden, Raspberry House, Raspberry Gate から構成される。

### Raspberry Patch (農園内の畑の一区画 (畝))

IoT デバイス中心の単一のネットワーク。ネットワーク内の通信は IP 通信であるとは限らないが、当該ネットワーク外との通信は IPv4 または IPv6 で行う。

### Raspberry Garden (農園内の畑 (畝の集まり))

IoT デバイスで構成されるネットワーク (Raspberry Patch) の集まり。

### Raspberry House (農園内の家)

IoT デバイスのためのネットワーク群 (Raspberry Garden) を管理するための管理用ネットワークまたは管理用の機器。管理機能を Raspberry Gate に実装した場合には、単独では存在しないこともある。

### Raspberry Gate (農園の門)

Raspberry Garden や Raspberry House と、インターネットとの間に位置するゲートウェイ (ルータ、プロキシ、ファイアウォール等)

### Raspberry Guardian (農園の門番)

Raspberry Farm の管理者。通常、Raspberry House か Raspberry Gate に遠隔ログインして管理作業を行う。

Raspberry Garden を構成する個々の Raspberry Patch は、それぞれが独立した IP ネットワークを構成する。このネットワークには、IoT デバイスを接続する。なお、本報告では IoT デバイスを「IPv4 または IPv6 で通信する機能を持つ、マイクロコントローラ (以下マイコン) と関連周辺機器 (主にセンサやアクチュエータを想定) から構成される装置」と定義している。この定義により、IoT デバイス

内のマイコンは、一つ以上のセンサや複数のアクチュエータを制御して計測や制御を行う機能と、他の機器と IPv4 または IPv6 で通信を行う機能が必要である。なお、近年ではセンサやアクチュエータが単体で IPv4 や IPv6 で通信する場合がある。この場合、その内部では「センサやアクチュエータ+マイコン」という上記と同一の構成をとっているため特別視する必要はないが、一体化しているため、内部のマイコンと関連周辺機器との接続に改善を施すようなことはできないか困難である。

IPv4 または IPv6 で通信する能力を持った IoT デバイスが直接あるいは間接的にインターネットへの到達性を持つ場合、当該デバイスや当該デバイスが属するネットワークに及び得る多種多様な脅威に耐え「安全で安心で快適」な動作をすることが求められる。すなわち、情報セキュリティで言う CIA (Confidentiality-機密性, Integrity-健全性, Availability-可用性) を確保する姿勢が IoT デバイスにも必要であり、このことについては、Raspberry Gate / Raspberry Guardian の研究開発でも指摘してきた [2][3][1][4]。Raspberry Patch 内にある「IoT デバイス」の場合は、その通信を OSI 参照モデルの視点から見ると、OSI 2 層以下にあつては無線 (たとえば WiFi や Bluetooth) や有線 (たとえばイーサネットや非同期シリアル通信) を用い、OSI 3 層以上にあつては IPv4 や IPv6 を用い、同じ Raspberry Patch 内の IoT デバイスとは直接に、同一 Raspberry Farm 内の他のデバイスとは直接あるいは Raspberry Gate 等を介して、Raspberry Farm 外とは Raspberry Gate を介して通信する。

このように、Raspberry Patch 内の IoT デバイスが他のデバイスと通信する部分についてはこれまで検討を行ってきたが、IoT デバイス内部で「マイコン」と「センサやアクチュエータ」がどのように接続され、どのように通信するのがよいかについては、これまで特段の議論をすることなしに、単にその時々目的や必要な通信距離や通信速度、さらに予算や開発時間をもとに使えそうな手法を使ってきた。たとえば、無線なら WiFi, ZigBee, BLE (Bluetooth Low Energy), 有線なら I2C, SPI, UART (RS-232C や RS-485 など) などである。これらを選んだ結果として、この部分の通信は WiFi を用いた事例を除くといずれも IP 通信ではなく、簡素なハードウェアと単純な通信方式となった。今回、IoT デバイスの通信部分を見直したところ、マイコンが複数のセンサやアクチュエータを持ち、相互の距離が数 m 以上数 10m 以下の範囲の場合、新たな機能をいくつか追加したりこれまでの使い方を見直したりするだけで健全性や可用性が増すとの結論に達した。そこで、「Raspberry Patch 内の IoT デバイス」のための「マイコンとセンサやアクチュエータを高い健全性と可用性を保って接続可能」な「有線接続方式」について検討し、設計と試験的実装を行い、継続的に使用して評価すること

にした。本報はその初報に相当する。

### 3. IoT 環境に適した有線接続方式の提案

#### 3.1 本報告で提案する有線接続方式

##### 3.1.1 I2C の採用

IoT デバイスの構成要素のなすマイコンと、当該マイコンが計測・制御する対象のセンサ・アクチュエータ類との接続方法(すなわち IoT デバイス内部での接続方法)は上述のようにいくつも候補があるが、今回はこれらのうち I2C[5] を選んだ。

I2C は、クロック線とデータ線(それぞれ SCL, SDA と表記することが多い)を用い、クロックに同期してデータをやり取りするシンプルな同期式 2 線式双方向シリアルバスである。I2C バスに接続される機器(以下 I2C 機器)には、クロック線の操作が許され通信全体を支配できる「マスタ」(master)と、マスタが送るクロックに同期してデータの送受を行い、8 ビット分の送受が済むごとに 1 ビットの肯定応答か否定応答をする「スレーブ」(slave)の二種類がある。I2C 環境においては、I2C 通信機能を搭載したマイコンがマスタになり、I2C 通信に対応したセンサやアクチュエータがスレーブとなるのが一般的である。マイコンが I2C スレーブになることは可能だが、センサやアクチュエータが I2C マスタになることはない。なお、I2C 機器のうちスレーブとなる機器は、I2C スレーブ、I2C デバイス、I2C モジュールなどと呼ばれることが多いが、後二者は I2C マスタを指すことも可能であり、I2C スレーブのみを指しているかは文脈依存となる。よって I2C スレーブが最適の呼称となるが、本報告では「I2C 素子」と表記することもある。同様に、SPI インタフェースを持つ機器についても SPI 素子と表記することがある。

I2C においては、すべてのスレーブにはアドレスが付与されている(一方、I2C マスタにはアドレスはない)。I2C バスのアドレス空間は通常 7 ビット \*2 で、いくつかの予約領域を除くと有効なアドレスは最大 112 個である。一つの I2C バスに接続されるすべてスレーブは原則として異なるアドレスを持つ必要がある\*3。また、I2C スレーブとして動く素子の多くはアドレスが固定され変更の余地がないか、当該素子の仕様で決まっているいくつかの候補から選択することしかできない。仮に、ある I2C 素子のアドレスが固定だった場合、同じアドレスを持つ I2C 素子を取付けてしまうとアドレスが衝突し、当該アドレスを読み出すと二つの素子が同時に応答するため通信が成立しない。この問題を解消するため、任意の I2C 素子を任意のタイミングで I2C バスに接続したり切り離したりする機能を持つ、I2C バスに特化したセレクトアやマルチプレクサ IC が製品

化されており、これを使うことでこの問題を対処できる。この場合、当該 I2C 素子にアクセスするためには、当該 I2C 素子の I2C アドレスに加え、マルチプレクサの I2C アドレスとマルチプレクサ上のチャンネル番号の 3 つを指定する必要がある。

I2C バスの 1 クロックは、電圧が低い状態 (LOW) から高い状態 (HIGH) を経て再び低い状態に戻る単純な方形波であり、その 1 周期が 1 クロックに相当する。仕様上の速度は 10kHz (= 10kbps, 以下同じ)、100kHz、400kHz、3.4MHz などがあるが、著者の周囲では 100kHz で動かししている事例が大半である。これは使用するマイコンや I2C 素子の組み合わせから必然的に決まる傾向であるため、マイコンを活用する多くの同様の研究開発環境でも同様の傾向が見られる。同期通信方式における 100kbps というクロック速度は、マイコンが UART を用いた非同期シリアル通信を使う際の実用速度よりは速いが、WiFi やイーサネットに比べたら 2 桁ないし 3 桁以上低速である。しかし、センサやアクチュエータのような複雑な通信機構の搭載が難しい素子であっても多対多通信が実現できる簡便性があるため、それほど速い通信を要求しない用途には I2C は適している。

IoT デバイスを構成するマイコンとセンサやアクチュエータが通信する方式の次の候補は SPI で、数 Mbps の通信が可能のため通信速度の点では I2C に勝る。しかし、SPI 素子はアドレスを持たず、その代わりに素子を選択する線 (chip select) を素子の数だけ用意し、マイコンから各素子まで配線する必要がある。本報告では最大で 25m 先にある複数のセンサやアクチュエータを制御することを想定しており、マイコンとの間に素子選択線を直接接続する必要がある SPI は候補から外れた。なお、I2C では対応できないが SPI なら対応可能な実装には有線 LAN (イーサネット) インタフェースがある。有線 LAN の速度は遅くとも 10Mbps であり、クロックが 100kHz や 400kHz の I2C では実用的な通信は難しいが、SPI なら速度的に完全ではないが、I2C よりは一桁速い通信速度を確保できる。

##### 3.1.2 従来の I2C バスに追加する機能

前項での考察を踏まえ、既存の I2C が提供する通信機能に以下の 6 機能を追加することにし、設計実装を行った。これらのうち、機能 1,2,3 は必須機能として先行して実装し、機能 4 は推奨機能とした。機能 5 は実験的に搭載して評価しているが機能 4 の実装がないとあまり意味を持たない。機能 6 はオプション (今後試行) とした。

##### 機能 1: 通信距離の延長と絶縁

通信距離を延長した I2C バスは独立して用意し、マイコンや I2C と直接接続はせずデジタルアイソレータによって絶縁接続する。当面の通信距離は 10m とし、今後の目標距離は 25m とする。なお SCL は 100kHz とする。

\*2 10 ビットのアドレス空間も定義されているが、普及状況を鑑み、本研究では 10 ビットアドレスは考慮していない。

\*3 まれに特定の受信専用の I2C 素子に対して意図的に同じアドレスを割り振ることもある

## 機能 2：電源供給

通信距離を延長した I2C バスは DC12V で駆動する。このバスに DC12V を供給する電源装置には給電能力に余裕を持たせ、各 I2C 素子への電源供給を可能にする。供給電圧 DC12V から必要な電源電圧 (5V, 3.3V, 1.8V のいずれか) への変換は受電側である I2C 素子の周辺で行う。また、機能 1 との兼ね合いから単なる電圧変換ではなく絶縁電源とする。

## 機能 3：ロックアップ回避・回復

I2C バスが陥り易いロックアップ状態を回避または同状態から回復する機能を搭載する。

## 機能 4：基準時間信号の供給

I2C ホストとなるマイコンのシステムクロックを高精度の外付発振器から供給することで、マイコンは I2C バスを高い時間精度で駆動できる。

## 機能 5：基準時間信号の検出

機能 4 による、高い時間精度で駆動される I2C バスは、正確な時間精度を持つ SCL で I2C バスを駆動する。その結果、この I2C バスに接続する I2C 素子は、I2C バスから高精度のタイミング信号を再生し利用可能になる。

## 機能 6：1-Wire での構成管理 (オプション)

I2C バスバッファのリセット、I2C マスタ、スレーブの配置順の確認等の I2C 単体では実現できない構成管理を 1-Wire を利用して支援する。

ところで、I2C 通信に必要な配線は電源を含めて 4 本である。後述する機能 1 の追加によって最大 25m 程度の通信距離を目指す、その際の配線の引き回し(ケーブリング)には、有線 LAN や電話の配線に用いられているツイストペアケーブルを使えるように配慮した。ツイストペアケーブルであれば、品質のよい製品が安価で入手し易く、極細を唱う製品も多い。また、環境によっては余剰品を再利用できる場合もあり、コストを低減できる場合がある。

I2C の通信速度には前述のように何種類かあるが、広く普及している I2C 素子が通信時に受け付けるクロック (SCL) は 100kHz か 400kHz であり、イーサネットなどと比べると圧倒的に低速である。よって、配線の引き回しに際して通信品質を劣化させるクロストークなどの悪影響は相対的に生じにくい受けにくい。そのため、最大通信速度 (周波数) によって各種存在するツイストペアケーブルのカテゴリの中でも、古くて低速なカテゴリのケーブルで十分対応できる。機能 1 の設計にあたっては、本来電話用で有線 LAN 用としては適さない、カテゴリ 3 ケーブルでも通信できるよう配慮した。

ツイストペアケーブルには電話用や有線 LAN に用など利用目的に応じて芯線の数に種類があり、少なくとも 4 芯 (2 芯 2 組) の製品と 8 芯 (2 芯 4 組) の製品がある、4 芯 (2 芯 2 組) なら I2C 機能は 1 系統 (SCL, SDA, Vcc, GND の

4 本) だけ提供可能だが、8 芯 (2 芯 4 組) なら、有線 LAN (2 芯 2 組) と共存させたり、8 芯すべてを本目的に使い、電源 2 芯を除いた残り 6 芯を用いて「I2C × 3 系統」あるいは「I2C × 2 系統 + 管理機能 (機能 6) × 2 系統」として使うことも可能になる。

なお、今回の報告では上記の各機能ごとの設計と実装状況を述べるが、各機能の評価結果に基づいて全体を再調整する方針である。このため、最終的な形態は上記から変化する可能性がある。最終的な形態については数ヶ月後 (2017 年度夏頃) を目途に取りまとめ、改めて報告したい。

## 3.2 関連事項

今回追加する 6 機能の関連事項を以下にまとめる。

機能 1 に含まれる「I2C バスの延長」については、I2C を利用した回路の設計者からは需要があるようで、半導体デバイス製造会社からはさまざまな情報が提供されている [6]。また、I2C バスの絶縁については、その必要性は I2C バス規格が公開された当初から認識されていたようだが、I2C バスが双方向であるのに対して信号絶縁用素子としてこれまで最も普及していた「フォトカプラ」の信号伝達が単方向であること、一部の例外的な製品を除くとフォトカプラの信号伝達速度が 100kHz に届かないことなどから、絶縁はあまり進んでいなかった。しかし、最近では「デジタルアイソレータ」の登場で、双方向かつ比較的高速な絶縁通信が可能になり、I2C バス専用のデジタルアイソレータも少量であっても入手が可能になった。

機能 2 に関連する「小規模・小範囲のネットワークのための電源確保」についても、すでにさまざまな試みがなされている [7]。たとえば、無線接続のネットワークでは電力の無線伝達が検討されており、有線接続のネットワークでもイーサネットでは PoE (Power over Ethernet) で電源を提供する技術がすでに普及している。この他、I2C の利用が想定される環境で、2 線のみで電源供給と双方向通信を実現することを意図した特許出願も見られる [8]。

機能 3 については、I2C バスのロックアップ問題が広く認識されているため、さまざまな試みや報告があり、たとえば半導体デバイス製造会社からも情報提供がある [9]。

機能 4 については、外部からクロックを供給してマイコンを動かすことは単体では特に目新しい機能ではなく、CPU のマニュアルなどに記載されている情報をもとに誰でも実装できる。マイコンやパソコンのシステムクロックを原子時計にした事例も存在するが、相対的に大掛かりな実装が多く、本報告で後述するような「超小型のルビジウム原子時計 (MAC) を Arduino にシールドとして実装し、ルビジウム原子時計をシステムクロックとするマイコンをモバイルバッテリーでも利用可能とした」事例は、本報告執筆時点では見いだせていない。

機能 5 では、「I2C バスの SCL は、間欠的にまとまって

(バースト的に)送られてくるタイミング信号」と捉えている。ここから継続的なタイミング信号を取り出す試は、CDR (クロックデータリカバリ)の一種である。CDRについては資料や文献が多数ある。

機能6のひとつとして今回実装したロックアップ回避方式は、1-WireによるIoT機器の構成管理支援の利用例の一つである。行なっていることは、I2C素子一つあたり数ビット程度の低速デジタル入出力を1-Wireで提供しているだけである。

## 4. 各機能の検討と実装

前節までに述べた、Raspberry Patchで利用する目的で既存のI2Cバスに追加する機能について述べる。

### 4.1 機能1：通信距離の延長と絶縁

もともとは、同一基板内あるいは同一筐体内の基板間を接続する程度を想定していたI2Cバスは、普及に伴い筐体間を接続する用途にも利用されるようになった。I2Cバスは、クロックはマスタからスレーブへの一方方向であるが、データは読み出しと書き込みで情報が流れる方向が逆になるので、少なくともデータ線であるSDAを担当するバスバッファは双方向性でなければならない。さらに、マスタ機能を担うデバイスが同一I2Cバスに複数存在するマルチマスタを許容しているので、この場合はクロック信号の流れる方向も単方向ではない場合がある。しかし、I2Cにはバスバッファの信号伝達方向を制御する信号は存在しないし生成できない。詳細は割愛するが、このためI2Cバスバッファは微妙な制御を強いられ、ここにノイズが乗ることでバスバッファがロックアップすることがある。I2Cバスのロックアップは、ハードウェア的にバスがラッチされた状態なので、I2Cマスタ側ではソフトウェア的にはなす術がない場合が多く、バスをハードウェア的にリセットする必要がある。ロックアップした状態からの復旧については機能3で対応する。

I2Cバスは、I2C用バスバッファが正しく機能している範囲においては、I2Cの「バスの静電容量は400pF以内」という仕様を満たしているはずである。したがって、この範囲でバスを延長できることになる。なお、一部のI2Cバスバッファはこれより大きな静電容量でも対応できるように作られており、その結果それらのバスバッファは30m以上の通信距離を確保している。

I2Cバスが延長できると、延長した分だけ配線がノイズを拾いやすくなり、ロックアップは防げても、I2C素子本体にダメージを与える事故が発生する確率が上昇する。そこで、I2Cバスを延長する部分と各々のI2C素子はデジタルアイソレータで電氣的に分離した。

### 4.2 機能2：電源供給

機能1で、I2Cバスの延長のための電源にDC12Vを使うので、これを降圧型レギュレータで安定化して、25m(または30m)の距離で5Vまたは3.3Vで1.5Aの給電は保証したい。電源供給系は、本来は適切な電圧と電力(電流)を供給できる安定化電源装置であれば十分だが、今回は機能3とも連携するために次項で述べる「非安定化機能」を搭載した。

### 4.3 機能3：ロックアップ回避

上述のように、I2Cは電源線2本とクロック線(SCL)、データ線(SDA)の4本で構成される。そして、クロック線を支配するマスタ側が常に主導権を握り、SCL線に同期信号を送出している。この同期信号のどのタイミングで、マスタとスレーブのどちらがSDA線を反転させるかで、「START」「データ送信または受信」「STOP」の状態が表現され、この順に遷移し繰り返す。STARTは、これからデータの送信または受信が始まることを示し、「STOP」はデータのやりとりが完了しバスが開放されることを示す。これだけであれば、マスタが必ず主導権を握って通信を支配できるが、実際には「クロックストレッチ」という、マスタの要求に応答が間に合わない判断した場合にスレーブがSCLを強制的にLOWにする第4の状態がある。この時、マスタは次のクロックの送出手を見合わせ、当該スレーブがSCLをHIGHに戻すのを待つ。この機能によって処理が遅いデバイスもI2C接続が可能になる反面、何らかの理由でSCLがLOWになったままになるとマスタはクロックストレッチが続いていると判断し通信を続けることができなくなりI2C通信は継続不可となる。これがロックアップを引き起こす。SCLがLOWになったままになる状態は、通信距離が長くなり双方向バスバッファを挿入している場合にノイズ混入に伴って発生し易い。I2C用の双方向バスバッファは、逆向きに配置された二つのバッファが相互に接続され一方の入力ともう一方の出力が接続されているので本質的にラッチし易い構造であるからである。

このような状態から回復するには、バスの状態をリセットする機能を持ったバスバッファの採用は効果的である。当該バスバッファがマスタの近くにある場合には、バスバッファまでリセット線を延ばすことができるが、今回の有線接続方式ではマスタとスレーブの距離が離れている上、スレーブの台数がI2Cバスの規格上の上限まで対応可能にしたいので、マスタと各スレーブを一对一で接続するのは無理である。この問題に対処する方法の一つは、配線を1本追加して1-Wire方式のデジタル出力素子を各バスバッファの近傍に配置し、バスバッファごとにリセットをかける方法である。実際、本有線接続の機能6はこの方法を可能にしている。しかし、そこまでせずリセットを実現するのが機能3のロックアップ回避である。

機能3のロックアップ回避では、機能2が各ノードに供給する電源電圧が12Vであり、各ノードが最終的にノード内に供給する電圧である3.3Vや5Vに対して十分余裕がある点に着目した。すなわち、供給電圧を意図的に一定の範囲で変化させても各ノードのレギュレータはそれを吸収し安定化後の電源電圧には影響が及ばない。そこで、電源電圧の変動が予め定めた変動と合致した場合、これをきっかけバスバッファをリセットする回路を挿入した。これは1BPS程度の独自方式の直流PLC(電力線通信)と解釈でき、リセットのための配線を別途用意する必要がなくなる。

実験では、通常は12Vを出力する定電圧電源の出力電圧を外部からの指示で約3V低下させる機能(非安定化機能)を付加し、約1bpsの速度でリセット指示を送出できる電源系を構成し動作を確認した(図2)。著者らはこの電源系を「非安定化電源」と呼ぶ。

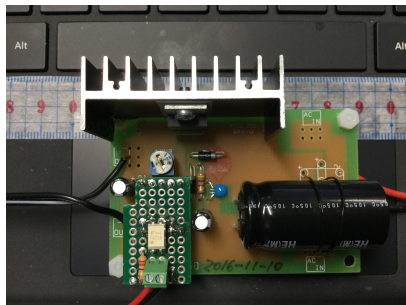


図2 機能3確認のための電源電圧を意図的に変動できる電源回路(非安定化電源)

#### 4.4 機能4：基準時間信号の供給

機能1から機能3までで、室内程度の距離(25mを想定)利用でき、電源供給機能とロックアップ回避機能を持つI2Cベースの有線接続が実現可能になった。それに対して機能4は必須ではないが、機能5を導入するなら必須となる機能である。

I2C素子を用いて計測や制御を行う場合、時間の経過を正確に把握できると計測や制御の精度が向上するが多い。必要な精度が秒単位であれば、高精度の水晶発振子を搭載したRTC(リアルタイムクロック)に1秒毎に割り込みを発生させれば目的は達せられるが、より高速に動作させたい場合にはマスタのシステムクロック自体を高精度の水晶発振子で生成したクロックに置き換える方法が有効になる。

実時間性という意味ではRaspberry PiよりArduinoが勝るので、今回はArduinoを「正確な周波数源から16MHzの供給を受けてI2Cマスタとして動く装置」にし、基礎的な実験をした。

まず、Arduino UNO互換機を用意し、16MHzの外付水晶発振子をシステムクロックにするよう、CPU(ATmega328P)のフューズビットを書き換えた。この水晶発振子を用いた外部水晶発振器は74HCU04を使ったオーソドックス

な回路であるが、VCO(電圧制御発振器)として用いるため、発振周波数を100ppm程度変化できるように配慮した。1000ppm程度にまで広げることができるとは今回はそこまではしなかった。[10]上記の水晶発振子とは別により高精度の発振器を用意し、両者とも100kHzに分周した上で、位相比較器して水晶発振器の発振周波数を修正するPLL回路を構成した(図3)。

実験の結果、まずVCOは、制御電圧を変化させると16MHzの水晶発振器の発振周波数を数10ppm変化できることを確認できた。その上で、高精度の基準信号源とVCOからそれぞれ100kHzを作り位相比較するPLL回路を作り、同期時に16MHzが得られるようにしたところ、Arduinoはこの16MHzをシステムクロックにして正常に動作した。

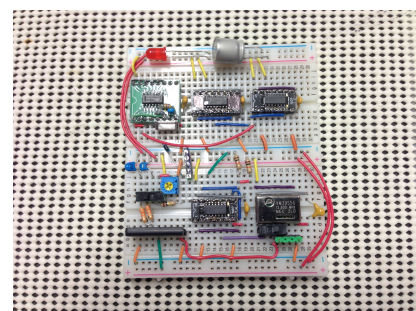


図3 機能4確認のためのPLL回路

#### 4.5 機能5：基準時間信号の検出

機能4によってI2Cマスタのタイミングクロック(SCL)の精度が向上するのであれば、I2CスレーブではSCLを参照してSCLと同じ精度の周波数のクロックを生成できるはずである。

I2Cバスの特性上、クロックは必ずマスター側が刻んでいる。仕様上は複数のマスターが存在可能だが、ここでは機能4を持つマスターが1台であると仮定する。また、I2CのSCL速度は、従来型で100kHz、400kHz、高速型では3.4MHzなどがあるが、ここではマイコンで一般的な100kHzであるとする。また、デバイスアドレスは一般的な7ビットアドレスのみで構成されているとし、10ビットアドレス形のことは考慮しない。

I2Cマスタが1台である場合、I2Cバスの送受信のメカニズムからSCLまわりについては以下が言えると判断した。

- I2CマスタはSCLとSDAを駆使して、START状態、データ通信状態、STOP状態等を作り出す。
- START状態からSTOP状態までにやり取りされるデータは全部でNバイトである。
- 1バイトの転送に際してやり取りされるデータは9ビットで、SCLに同期して行われる。このうち先頭の8ビットがデータであり、最後の1ビットはスレーブ側が肯定応答するか否定応答(実際には無応答)の為に

使われる。

- N バイトのデータがやりとりされる時、各バイトの 2 ビット目から 8 ビット目までは、マスターが自分のペースで送出できる。9 ビット目はスレーブがストレッチ状態を作る可能性があるのでタイミングとしては正確ではない。START 状態になってから 1 ビット目終了までの所用時間もマスターが支配できるはずであるが、実装の影響が揺らぎが見られた。
- 以上より、各バイトの 2 ビット目から 8 ビットまでが送受されている時の SCL に対して PLL をかければスレーブは正確な SCL のタイミングを抽出でき、これをもとに 1MHz, 10MHz, 16MHz などの SCL の周波数 (現状は 100kHz) の整数倍の基準時間信号を再生できる。

そこで、図 4 のような実験系を作って評価を行ったところ、SCL からシステムクロックの再現ができた。

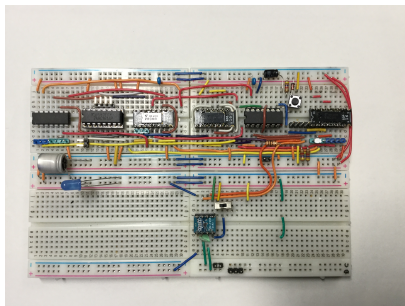


図 4 機能 5 の確認のための I2C クロックから基準信号を検出する試作回路 (撮影のため外部との接続線は外した)

#### 4.6 機能 6(構成管理機能)：余った配線を使い構成管理をする(オプション)

室内での使用を前提とし全長を最長で 25m ないし 30m に設定しているため、見渡せる範囲にすべての I2C スレーブが配置されている状況が想定できるが、壁の中、壁の向こう、天井の上、床下など見えない場所に設置されていることも想定でき、I2C スレーブの配置状況を物理的にも論理的にも最小限の手間で把握する仕組みがあると利便性が高まる。

物理的な配置状況は「見ればわかる」場合もあるが、同一機能同一形状のセンサーが複数接続されている場合は、目の前のセンサーが同一 I2C バス上に存在する複数の候補のどれに相当するかは判断が難しい。I2C においてはバス上の位置は意味を持たないので、論理構成の把握とは「接続中の I2C スレーブの I2C アドレス」と「担当する機能」のペアを集めることを意味する。実際の運用においては、特定の I2C スレーブをバスから切り離して当該 I2C スレーブと同じ I2C アドレスを持つ別の I2C スレーブを接続して評価するといった場面が想定できる。本研究で提案している有線通信方式では、手動で特定のノードを I2C バスから切り離す機構は持っていて、I2C ネットワークか

ら分岐する地点でスイッチを手動で ON/OFF することで着脱を実現している。この部分をホスト側から遠隔操作できれば、特定の I2C スレーブを切り離せる。現状では、I2C-WIRED の配線には 2 芯 4 ペアからなるツイストペア線を使うことを想定しているため、構成によっては制御用の線を 1 ペア確保できる。このペアを使い、1-wire プロトコルを使って、I2C スレーブを切り離す動作を可能にする。

## 5. 考察

本報告では、I2C バスに 6 つの機能を追加した有線接続を提案し、期待した動作を確認したが、評価は完了していない。特に長期間の利用に伴う経年変化についてはまだ予想しかできない。

### 5.1 現在の到達点での評価

ここでは現状と当面の課題を機能ごとに示す。

#### 機能 1：通信距離の延長と絶縁

距離の延長と絶縁は、想定どおりの性能が得られている。本報告では言及できなかったが、インターネットへの接続性を持ち、I2C マスターとなる IoT デバイスを Raspberry pi 単独で実現すると、機能 4(高精度の時間基準の提供)が難しくなる。Raspberry Pi と Arduino の組み合わせで実装した場合、インターネット側との通信を担当する Raspberry Pi と I2C を担当する Arduino をどう連携させるかが課題となる。UART 接続は低速な上に、Raspberry Pi と Arduino の組み合わせだと 57.6kbps が上限となる。この速度だと、I2C のクロック速度を下回ってしまい効率のよい通信が実現しない。両者は SPI 接続することになる。

#### 機能 2：電源供給

現時点では電源供給機能は、機能 3 の一部でもある非安定化機能も含め順調に機能している。

#### 機能 3：ロックアップ回避

非安定化電源によるリセットは想定通りの動作をしている。現在は電源の非安定化に伴い、すべての I2C バスバッファにリセットがかかるが、機能 6 が実装されれば電源を非安定にせずに個別リセットが可能になる。ただし、ロックアップ解除のための個別リセットを実施するには、「リセットが必要な I2C バスバッファ」を担当している「1-Wire 接続のデジタル出力素子」のアドレスを正しく把握していなければならないので、機能 6 の実装で個別にロックアップ回避ができるようになっても、非安定化電源によってすべての I2C バスバッファに一斉にリセットを指示できる機能 3 の有効性は減じない。

#### 機能 4：基準時間信号の供給

現在の実装は、高精度の基準周波数で Arduino を駆動しているだけであるが、PLL 周りのローパスフィルタ

の時定数や VCO の性能にはチューニングの余地がある。また、時刻源として採用した超小型ルビジウム原子時計の評価は、別の共同研究で連携がある NICT の日本標準時関連グループと検討する予定である。

#### 機能 5：基準時間信号の検出

間欠的に I2C 通信をおこなっている I2C バスの SCL から 100kHz のクロックを抽出し、これを PLL の基準信号にして 160 通倍した信号が、16MHz で駆動する Arduino のシステムクロックと同期できることを確認できた。ただ、機能 4 以上に PLL まわりのパラメータのチューニングが重要だということも明らかになった。引き続き調整を続ける。

#### 機能 6：1-Wire での構成管理

1-Wire による I2C バスのリセット実験は未着手である。一連の評価実験にはカテゴリ 5A ケーブルを使っており芯線に余裕があるので急いで着手したい。

### 5.2 今後の展開

機能 1～5 を加えたことで、データ通信機能に電源供給、回線復旧、基準信号供給機能を加えた I2C ベースの近距離有線接続が構築できることを確認できた。前項でも述べた個別の課題に順次対応し、機能 6 を加えた全体テストを 2017 年度初頭には開始したい。

なお、今回はコストについての議論はしていないが、試作で用いた部品は、ロジック回路、PLL 回路、絶縁回路、電源回路のいずれにおいても、同じ分野の部品よりは相対的に特殊で高価な部品が多かった。また、複数の IC を組み合わせることで配線数も専有面積も機能を追加した分だけ大きくなった。コスト削減と専有面積の縮小は今後の課題とするが、このうち後者については、一辺の長さが 5mm 以下の超小型 FPGA が小規模試作用途で利用可能になってきたので、デジタル回路の FPGA 化は本報告執筆時点での最優先課題となっている。たとえば機能 5 は、本通信環境下のノードごとに設置する必要があるためできるだけシンプルで小さな回路にしたい。そのため Intel/Altera MAX10 10M02 のような超小型の FPGA に必要機能を盛り込むことを検討している。

また、機能 6 において、1-wire を使って構成管理を行っても I2C が 1 系統であれば必要な芯数は全部で 5 本であり、8 芯ケーブルを用いるのであれば 3 本の余裕がある。この余った 3 本を 2 系統目の I2C に割り当ててではなく、新たに I2S に割りあてて、高品質のオーディオ信号を通すことも試したい。

## 6. まとめ

Raspberry Patch での使用を前提に、既存の I2C バスに 6 つの機能を追加した有線接続方式を提案し、設計・実装して評価を開始した。6 つの機能には、必須機能と任意選択

機能があるが、これらの機能を活用することで、全長 25m 程度の屋内環境で実験や試験的運用を繰り返す用途には、良好な保全性と可用性を提供できることが確認できた。引き続き日常的に実際の機材の利用しつつ評価を継続する。さらに、この有線接続方式を RG モデル配下で用いることで、完全性、気密性、可用性をバランスよく実現できることを立証したい。

なお、本報告に関連した資料、情報は以下の URL から参照できる。<https://goo.gl/rHqnnK>

**謝辞** 著者は、電子工作愛好者が実際に集って技術的な情報交換する「木いちごの会」というハンズオンを定期的に石川県野々市市や同県金沢市で開催している。本報告で述べた有線通信方式の試みは、同会における Raspberry Pi を用いた IoT デバイスの安全安心な運用についての議論がきっかけとなって始まった。それぞれ異なる技術的背景と好奇心を持って同会に集う、実行力あふれる電子工作愛好者各位に感謝する。また、本研究の成果を最初に提供することになる、Raspberry Gate や Raspberry Guardian の研究開発は、北口善明東京工業大学准教授、鈴木裕信専修大学非常勤講師と連携して実施している。両教員にも感謝したい。なお、本研究は科研費（課題番号：15K00119）の助成を受けたものである。

#### 参考文献

- [1] 大野浩之，鈴木裕信．電子工作愛好者向けセキュリティゲートウェイの構築（第四報：Raspberry Guardian の実証実験に向けて）．マルチメディア，分散，協調とモバイル（DICOM02014）シンポジウム講演論文集，Jul 2014.
- [2] 大野浩之，鈴木裕信．6D-4 電子工作愛好者向けセキュリティゲートウェイの構築（第一報：設計と実装）．情報処理学会全国大会講演論文集，Vol. 2014，No. 3，pp. 47--48，mar 2014.
- [3] 大野浩之，鈴木裕信．6D-5 電子工作愛好者向けセキュリティゲートウェイの構築（第二報：運用と管理）．情報処理学会全国大会講演論文集，Vol. 2014，No. 3，pp. 49--50，mar 2014.
- [4] 大野浩之，鈴木裕信．電子工作愛好者向けセキュリティゲートウェイの構築（第三報：構成要素の検討と性能評価）．情報処理学会第 160 回マルチメディア通信と分散処理研究会予稿集，Jul 2014.
- [5] *UM10204 I2C-bus specification and user manual*. NXP Semiconductors, 2014.
- [6] *AN10658 Sending I2C-bus signals via long communications cables*, 2008.
- [7] ネオテス株式会社．ワイヤレス電力伝送/ワイヤレス給電の分類（2016 年）．<http://www.tlm.co.jp/wpt/kindofWPT.pdf> 2017 年 2 月 1 日閲覧．
- [8] 日本国特許庁（JP）公表特許公報（A）．特表 2014-534686 高速データおよび配電のための 2 線式通信システム．[https://www23.j-platpat.inpit.go.jp/pkd/all/pkd1/JPA\\_2014534686.pdf](https://www23.j-platpat.inpit.go.jp/pkd/all/pkd1/JPA_2014534686.pdf) 2017 年 2 月 1 日閲覧．
- [9] Jim Greene. *AN686 Implementing an I2C Reset*. Analog Devices, 2009.
- [10] 西村芳一．無線によるデータ復調技術．CQ 出版社，2003.