

HoUZK/JP - 手作り USB HID を用いたシステム管理支援, 特殊入力装置作製, およびプレゼンテーション手法について

大 野 浩 之^{†1}

非同期シリアルインタフェースを介して USB キーボードや USB マウス等の USB HID を操作する装置を設計実装した。HoUZK と命名したこの装置を使えば、これまでにない多様な USB 入力装置を容易に実現できる。その際、USB ホスト側からは標準的な USB HID に見えるのが特徴である。HoUZK は、UZK という先行事例を取り込んで実現したが、今後は、AVR ベースのマイクロコントローラボードに LUFA (Lightweight USB Framework for AVR) プロジェクトの成果を取り込むといった方法も検討し、利便性の改善を図る。

**HoUZK/JP - A System Management Tool, Special Human
Interface Devices and A New Presentation Tool in One
Serial to USB HID Converter**

HIROYUKI OHNO^{†1}

An asynchronous communication interface to USB HID (such as USB Keyboard and USB Mouse) converter has been designed and implemented by the author. The converter named HoUZK works as a standard USB HID for USB host. However users can control the USB HID perfectly through HoUZK's asynchronous communication line. With HoUZK, we can create new input devices with USB interface easily. In the near future, we will introduce the outcome of LUFA (Lightweight USB Framework for AVR) project onto AVR based micro controllers. This will make HoUZK more comfortable and easier to use.

^{†1} 金沢大学
Kanazawa University

1. はじめに

著者は、多数の KVM スイッチ^{*1} が導入された環境での統一的なシステム運用に 10 年以上前から興味を持ち、いくつかの提案を行ってきた¹⁾。大規模 KVM スイッチ群を効率よく的確に操作するには、「非同期シリアルインタフェースを介してキーボードやマウスを遠隔操作する装置」があるとよい。しかし、必要な機能を持ち、安価で小型で、回路図とファームウェア公開されていることという条件を全て満たす製品は存在せず、しかたなく既存製品を組み合わせて妥協したり、「USB デバイスとして機能する Linux ベースの電子手帳」のカーネルに手を加え、非常時における情報通信環境の一助にすべく電子手帳上に開発した「-ARK デバイス²⁾」を USB マウスや USB キーボードに見せかけてシステム管理業務に投入を試みる³⁾ など、さまざまな工夫をしてきた。

その結果、それなりの成果は得られたが、目的達成のために利用してきた重要な機材が相次いで製造中止になり、これまでのアプローチをそのままの形で継続することは困難になった。加えて、最近の技術動向には注目し値する事項が見られることから、今般、新たに設計と実装を見直した。この見直し作業の過程で生まれたのが HoUZK/JP (ほおずき プロジェクト) であり、HoUZK という装置である。本報告では、HoUZK/JP のこれまでの経緯と現状、そして今後の展開について述べる。

2. 本研究の目的

本研究では、非同期シリアルインタフェースを介して USB キーボードと USB マウスを自由に操作する 図 1 のような装置を、現時点で容易に入手可能な技術や製品を元にして新たに作製する。この装置を用いれば、この装置を接続した機器 (パソコンが該当する) のキーボード操作とマウス操作がこの装置の非同期シリアルインタフェースを経由して可能になる。これにより、大規模 KVM スイッチ群の操作だけでなく、個人常用環境、研究室環境、業務遂行環境等で利用可能な多彩な入力装置を新たに構築できる。

図 1 のような、非同期シリアルインタフェースと USB インタフェースを一つずつ持つ「非同期シリアルインタフェースを介して USB キーボードや USB マウスのような USB HID を自由に操作する装置」を本研究では「UART/HID 変換器」と呼ぶ。なお、USB HID (Human Interface Device) クラスには、キーボードとマウス以外の装置もあるが、現時点

^{*1} Keyboard/Video/Mouse Switch の意味。PC 切替器ということもある。本報告では KVM を Kernel-based Virtual Machine の意味で使うことはない。

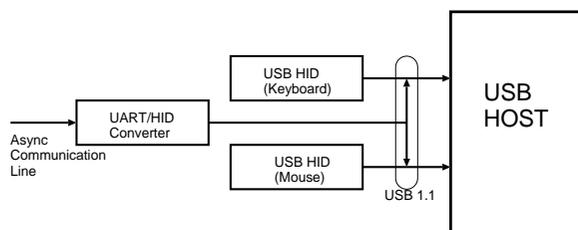


図 1 非同期シリアルインタフェースを介して USB キーボードと USB マウスを操作する装置
Fig.1 A Device Which Controls USB Keyboard and USB Mouse Over Asynchronous Serial Communication Line

では USB キーボードと USB マウスのみを念頭においている。

「UART/HID 変換器」は、これを自らの USB ポートに接続した USB ホストから見れば、標準的な USB HID に見える。このため、新たにカーネルモジュールやデバイスドライバ等を新規に開発して追加しなくても、パソコン等の USB ホストに接続するだけで直ちに利用可能になる。これは「UART/HID 変換器」を用いる方法の大きな特長の一つである。

これまででも、特定の用途に特化した専用の入力装置をハードウェアから新規に作り、専用ハードウェアのための専用デバイスドライバを作り、その上で専用アプリケーションを作るというアプローチは、数多く行われてきた。このアプローチは「かゆいところに手が届く」サービスを提供可能にするが、ソフトウェアが提供されない環境では何もできない。たとえば、デバイスドライバやアプリケーションが Windows にしか対応していなければ、Mac OS X や Linux を OS と機器の利用者は利用できない。ドライバやアプリケーションソフトウェアがオープンソースでなければ、提供された専用ドライバやアプリケーションを修正したくても自分で直すことはできないし、オープンソースであっても異なる OS への移植は容易でない場合が多い。

これに対して、「UART/HID 変換器」を用いる方法には、上記のアプローチにない特徴がある。「UART/HID 変換器」にできるのは、USB キーボードや USB マウスのエミュレーションにすぎないので、専用ハードウェアと専用ソフトウェアでは実現可能でもキーボードやマウスにできない操作は実現できない。たとえば、タッチパッド上の複数のポイントを同時に押えて操作する「マルチタッチ」のように、キーボードとマウスだけでは実現困難な操作は、「UART/HID 変換器」でも実現できない。しかし、キーボードとマウスでできる操作であれば、USB ホスト側に新たなデバイスドライバやカーネルモジュールを組み込むことなく「UART/HID 変換器」でも実現できる。さらに、非同期シリアル通信インタフェース

(UART) 上の通信プロトコルを公開すれば、「ホスト側から見れば標準的なキーボードやマウスと認識されるが、実態はそれとは全く異なる新たな入力装置」の第三者による開発も容易になる。

3. HoUZK の開発

3.1 「UART/HID 変換器」の実現方法

「UART/HID 変換器」(非同期シリアル通信で制御可能な USB キーボード/マウス) を実現する方法には、大別して3つの方法がある。

- (1) 既存の USB キーボードや USB マウスのハードウェアを改造し、キーボードやマウス内にリレースイッチ、フォトカプラ、アナログスイッチ等のスイッチ素子とこれらによる操作する回路を組み込み、人間の手による本来のキーボード打鍵やマウス操作によって生じる電気信号と同等の電気信号を、非同期シリアル通信で送られて来た情報をもとにスイッチ素子を利用して発生させる方法。
- (2) 非同期シリアル通信で送られて来た情報をもとに PS/2 デバイス (PS/2 キーボードと PS/2 マウス) をエミュレートするハードウェアを作製し、さらに市販の PS/2 - USB HID コンバータを介して、これらの PS/2 デバイスを USB キーボードや USB マウスが実在するように見せかける方法。
- (3) 非同期シリアル通信で送られて来た情報をもとに USB HID (USB キーボードや USB マウス) をエミュレートするハードウェアを作製し、USB ホストからは USB キーボードや USB マウスが実在するように見せかける方法。

方法 (1) は、実在するキーボードやマウスを改造して用いるため USB ホスト側から見ると完璧な USB デバイスに見える。実在のキーボードやマウスを使うため小型化はしにくいですが、USB ホスト側から見た場合一番確実な方法である。ただし、改造箇所と改造方法が改造するキーボードやマウスごとに異なり、あまり現実的な方法ではない。

方法 (2) は、PS/2 デバイスの諸情報が公開されており、そのハードウェアが単純であることに着目したアプローチである。PS/2 キーボードと PS/2 マウスをエミュレートする部分までは新規に作製し、PS/2 デバイスから、複雑な動作をする USB HID への変換は、安価に市販されている「PS/2 キーボード/マウスを USB キーボード/マウスに変換するアダプタ」に任せる。この方法は、PS/2 デバイスの仕様中の信号のタイミングに関する部分に柔軟性がありすぎる点が問題になる。このため、「PS/2 デバイスをエミュレートした装置」と「PS/2 デバイスを USB HID デバイスに変換するアダプタ」との間にいわゆる相性問

題が生じてしまい、なかなか思うように動かないことが有り得る。また、「PS/2 キーボードを USB HID キーボードに変換するアダプタ」には、その挙動が必ずしも適切でない場合があるとの報告もある⁴⁾。このため、方法 (2) は、少ない工数で小型軽量に実現できそうであるが、満足の行く結果にならない場合がある。

方法 (3) は、非同期シリアル通信で制御可能な USB キーボード/マウスそのものを作るアプローチである。今回は、消去法でこの方法を採用することになったが、この方法の場合、新たに作る USB デバイスのベンダ ID やプロダクト ID をどうするかという問題が残る。実験室内での動作確認の範囲であれば、USB デバイス開発事業から撤退したベンダの ID を借用するといった強引な方法もあるが、本格的に利用するのであればベンダ ID を正規に取得する必要がある。

検討の結果、今回は方法 (3) を採用した。

3.2 UZK と HoUZK

前項の方法 (3) を実現するためには、USB デバイスとして振る舞える機能を持ったハードウェアが最低限必要になるが、(USB ホストではなく) USB デバイスになる機能を持ち、ユーザが USB デバイスをプログラミングできるパソコンや電子手帳は、現状ではほとんど存在しない。SHARP 社の Zaurus SL-C3100/C3200 シリーズは、Linux を OS とする電子手帳で、カーネルモジュールの追加によって USB デバイス (HID) として振る舞わせることができたが、同機はすでに製造中止になっており流通在庫も払底している。したがって、マイクロコントローラを搭載した CPU ボードと USB デバイス用の専用 IC を組み合わせるか、CPLD や FPGA で同等のことは実現するなどして新規に製作する必要がある。

3.2.1 UZK

UZK⁵⁾ は、uratan 氏が開発しインターネット上で公開しているキーボード・マウス共有・延長器で、オプティマイズ社製の MINI EZ-USB⁶⁾ を使用している。詳細は同氏の UZK ホームページに譲るが、当該ホームページにある UZK に関する説明を要約の上引用すると、「親 PC、子 PC を共に USB で結び、親 PC の本物のキーボードの入力をフロントエンドアプリケーションにより OS から取得し、子 PC に渡すことで USB キーボード・マウスをエミュレートし共有する機器」となる。UZK は、回路図とファームウェアが公開されているので、これらを解析することでそのしくみを詳しく把握できる。その結果、UZK は、筆者が求めている「UART/HID 変換器」をフォトカプラを挟んで対向させた構成になっていることがわかった。

3.2.2 HoUZK

前項の検討をもとに、今回は「UZK の半分」を使って「UART/HID 変換器」を構成することにした。このアダプタは、「UZK の半分」に相当することから、「Half of UZK」の頭文字から HoUZK と名付けた。HoUZK は、「ほおずき」と読む。

UZK にとっては、UZK 内部で非同期シリアル通信を行う部分は外部からは見えない部分であり、そのハードウェア仕様や通信プロトコルを公開しなくても UZK 利用者には特に支障はないが、実際には詳細情報が詳しく提供されている。

一方、HoUZK にとっては、非同期シリアル通信部分は他の機器との通信を行うためのインタフェースなので、第三者が HoUZK に接続できるシリアルデバイスを作成するにはハードウェアや通信プロトコルの詳細情報が必要となる。

HoUZK の非同期シリアル通信部分のハードウェア仕様や通信プロトコルを独自に定めることは可能であるが今回は UZK のシリアルインタフェースに準拠ことにした。これにより、以下の長所と短所が生じる。

長所：

- UZK と同一のシリアルインタフェースを持つことでその詳細仕様を新たに検討せずに実績のある UZK に依存できる。
- UZK は、ハードウェアもファームウェアも全て公開されているため、利便性が高い。(ソフトウェアは X11 ライセンス)
- HoUZK の情報はインターネット上にはまだほとんどないが、UZK の公開情報は充実している。HoUZK に接続するデバイスを第三者が製作する場合に UZK の情報を利用できる。
- UZK の後継の ArmUZK は完成品が入手可能で、これを改造すれば 1 台の ArmUZK から 2 台の HoUZK を比較的容易に製作できる。

短所：

- HoUZK を自力で作るには、現状では CPLD (MINI EZ-USB) を使った回路を実装したりファームを書き込んだりする技術力が必要。
- UZK の非同期シリアル通信プロトコルはあくまで UZK 内部での利用を前提としているため、一対一通信である。これをそのまま HoUZK に転用した場合、非同期シリアルインタフェースのハードウェアあるいは通信プロトコルの拡張なしに複数の HoUZK を同時に制御することは難しい (後述)。
- HoUZK は UZK を参照しているが、UZK 側は HoUZK を認知しているわけではない

ので、将来 UZK のプロトコルやインターフェースが変更になった場合上記の長所は失われる。

3.3 HoUZK のハードウェア

HoUZK のハードウェアは、UZK のほぼ半分に相当する、USB 1.1 インターフェースと非同期シリアルインターフェースを持つ MINI EZ-USB で実装した変換アダプタである。

HoUZK の USB 1.1 インターフェース部分と MINI EZ-USB 部分は、UZK の当該部分と同一であるが、非同期シリアルインターフェース部分は、UZK と構成が異なる。

UZK では、MINI EZ-USB からの信号は TTL レベルのデジタル入出力を行うフォトカプラを介して UZK のもう半分に送られ、もう半分の UZK からの信号も同じ種類のフォトカプラを介してもう一方の MINI EZ-USB に送られる。フォトカプラを介するのは、UZK の一方の半分に接続した機器と UZK のもう一方の半分に接続した機器が電氣的に干渉しないようにするためである。

UZK では非同期シリアルインターフェースは、内部バスであり、UZK の設計・実装者以外はその存在を意識する必要はないが、HoUZK では、この部分は HoUZK 利用者が非同期シリアルインターフェースを持つ機器を接続する部分であるので、明確な仕様をもとに実装され、拡張性も確保しておく必要がある。

HoUZK には、現在以下の4つのモジュールが用意されている。HoUZK の利用者は、そのうちの一つを選んで自らが用意した機器と HoUZK を接続する非同期シリアル通信インターフェースを構成する。

- (1) 非同期シリアルインターフェースと HoUZK 内部の MINI EZ-USB との間にフォトカプラを配置して電氣的に絶縁した非同期シリアル通信モジュール (信号は TTL レベル)
- (2) (MINI EZ-USB 側から見て) 上記 (1) の先にレベルコンバータを付加した、RS-232C に準拠した非同期シリアル通信モジュール
- (3) フォトカプラを使わずに MINI EZ-USB に Bluetooth 通信モジュールを接続したワイヤレスシリアル通信モジュール
- (4) Bluetooth の代わりに ZigBee 通信モジュールを接続したワイヤレスシリアル通信モジュール

複数台の HoUZK を 1 台の制御装置から一括して制御する場合には、一対多通信が可能な上記 (3), (4) を用いるか、(2) に準ずる形で (1) に RS-485, I2C, SPI 等のコンバータを付加する必要がある。

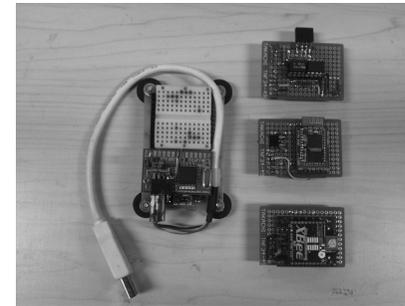


図 2 HoUZK の構成 (左: 本体, 右上: 有線通信モジュール (フォトカプラ対応), 右中: Bluetooth モジュール, 右下: XBee (ZigBee) モジュール)

Fig. 2 Structure of HoUZK

3.4 HoUZK の通信プロトコル

HoUZK の非同期シリアルインターフェースが採用している通信プロトコルは UZK が内部で用いている通信プロトコルと同一である。このプロトコルを UZK プロトコルと呼ぶ。

UZK のソースコードによると、UZK プロトコルは、USB HID における USB キーボードや USB マウスが USB ホストとやりとりしているデータパケットにほぼ準じたデータ形式と手順になっている。なお、HoUZK の非同期シリアルインターフェースの通信速度は 9600bps である。これは、HoUZK のもとになった UZK のファームウェアの制限をそのまま引き継いだためであるが、UZK や HoUZK では、マウスやキーボードを制御するだけなので、9600bps という通信速度が問題になることはない。

4. HoUZK の実装と評価

4.1 実装

HoUZK の中核部分は、UZK と同じくオプティマイズ製の MINI EZ-USB を使用し、非同期シリアルインターフェースを 4 種類用意した。これらの交換を容易にし、これらとは別のインターフェースを仮組みしていつでも評価できるよう、HoUZK は MINI EZ-USB とブレッドボードを並べた図 2 のような構成になった。HoUZK はバスパワーで稼働させている。

4.2 活用事例 (1)

HoUZK は、当初の想定どおり大規模な KVM スイッチ群の操作に利用できる。KVM スイッチ群を操作する場合、接続先のコンピュータは通常はサーバやルータであり、キーボードからの打鍵だけで操作が完了することが多いが、HoUZK を用いればキーボードだけでなくマウスの操作も可能である。

この事例では、KVM スイッチ群が現在選択している機器のコンソール画面の扱いが問題になる。KVM スイッチのコンソール画面が手近にあり、操作の大半は操作者が本来のキーボードとマウスを利用して直接行い、一部の複雑なキーシーケンスのみを HoUZK に委ねるといった使い方であれば、現状の HoUZK で何ら問題はない。しかし、離れた場所にあるデータセンターに設置した KVM スイッチを操作したいといった場合には、コンソール画面を転送するしくみか別途必要になる。コンソール画面をキャプチャしてネットワークごとに配信するための機材としては、epiphany 社の VGA2Ethernet のような機材があるのでこれらを併用することになる。

今回は、HoUZK の動作確認を兼ねて、Mac OS X 上の X11 上に Xlib を使ったテストプログラムを作製した。このテストプログラムは、キーボードやマウス操作に伴って発生する全ての X イベントを検出し、これらを UZK プロトコルに変換し、USB シリアルポート経由で HoUZK に送る機能を持つ。このプログラムは、単純な構成ながら特に HoUZK 開発の初期段階の基本動作確認時には大いに役だった。

4.3 活用事例 (2)

HoUZK と非同期シリアルインタフェースを介して接続する機器は、UZK プロトコルで通信することで USB キーボードと USB マウスをエミュレートできるので、新しい形態のキーボードやマウスをこれまでより容易に構成できる。新しいキーボードやマウスの作製者は、UZK プロトコルで非同期シリアル通信する部分まで作るだけで、USB ホスト側の OS に関わりなく USB ホスト側で利用可能になる。この特徴をデモンストレーションするため、新しい入力装置（キーボードとマウス）の試作を行っている。ここではその一例を以下に示す。

前項で、Mac OS X 上のテストプログラムで HoUZK の基本動作が確認できたので、非同期シリアル通信で UZK プロトコルにしたがって通信すれば、USB ホスト側から見た場合 USB キーボードや USB マウスと認識されることが確認できた。そこで、AVR マイコンコントローラを搭載し、使いやすいことで定評のある Arduino でも動作確認を行った。Arduino と HoUZK をシリアル接続してテストプログラムを実行したところ、Arduino Duemillanove (後には Arduino UNO) 上で実行したスケッチ (プログラム) の指示にしたがって USB ホスト側のキーボードやマウスが動いた。

この時点で、Arduino が収集した情報をもとに UZK プロトコルで HoUZK を通信すれば USB ホストのキーボードやマウスとして動くことが明らかになったので、スケッチを変更した。Arduino に二つのキースイッチを取り付けて、エレキー (Electronic Keyer) を構

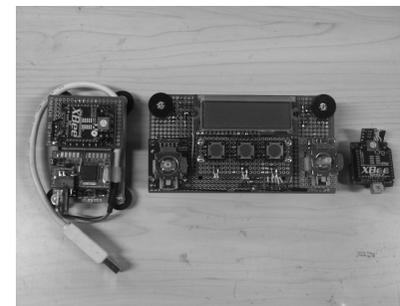


図 3 HoUZK を使ったプレゼンテーション支援装置 (左: HoUZK, 中: 制御装置, 右: 加速度センサ)
Fig. 3 A new presentation support tool by HoUZK

成してモルス信号の送付を可能にし、モルス信号を打鍵した際に、これをただちに解釈してキーボードやマウスの動きに変換することを試みた。キーボードとしての動作はすぐに確認できたので、モルスコードを試験的に拡張し、マウス操作もモルス符号で表現する試みを続けている。

4.4 活用事例 (3)

HoUZK を利用することで、USB ホスト側から見ると標準的な USB キーボードや USB マウスに見え、ユーザ側からは非同期シリアル通信で制御できる装置を容易に作れるようになったので、プレゼンテーションに特化した図 3 のような装置を試作した。

この装置は、以下のように動作する。

- (1) 操作者は、アナログ加速度センサと XBee (ZigBee 対応通信モジュール) からなるセンサユニットを両手に取り付けて、両手を大きく動かしてジェスチャーを行う。
- (2) アナログ加速度センサの出力をその場で A/D 変換し、ZigBee プロトコルで制御装置に送る。
- (3) 制御装置は、受信した加速度情報をもとに、その変化からジェスチャーの意味を解釈し、解釈に対応したマウスの動きやキーボード操作を決定する。
- (4) 最後に、決定したキーボードやマウス操作に対応した情報を ZigBee で HoUZK に転送する。

操作者の見ぶりや手ぶりをコンピュータに反映させる技術自体は珍しくはないが、HoUZK では、コンピュータ側からみたら単なるマウスに見えるため、特に何かをインストールしたりせずに済む点が特徴である。その分、加速度情報からジェスチャーの意味を解釈してキーボードやマウスの操作に変換するといった作業を制御装置がしなければならないが、USB

ホスト上では特に何もしなくてもジェスチャーが有効になるという効果の方が、制御装置にジェスチャー解釈のプログラムを新たに組み込む手間よりも大きいとみなせる。

5. 考察と今後の展開

5.1 HoUZK の優位性

HoUZK の中核をなす部分は UZK であるので、UZK の持つ、ハードウェアが小型軽量で回路図が公開されており、ファームウェアはオープンソース (MIT X11 ライセンス) であるといった特徴は全て引き継いでいる。

また、UZK では内部バスである非同期シリアルインタフェースを外部とのインタフェースに転用し、有線接続 2 種類 (TTL レベル, RS-232C 準拠。いずれもフォトカプラで絶縁)、無線接続 2 種類 (Bluetooth, ZigBee) を用意したことで、多様な接続を可能にした。さらに、これらのインタフェース上を流れる通信プロトコルも明確なので、第三者が新たな USB キーボードや USB マウスを容易に構築できるといった長所がある。

5.2 HoUZK の拡張

現在の HoUZK は、非同期シリアルインタフェースを介し、HoUZK を制御する側と一対一で通信する。UZK においては、UZK プロトコルが一対一通信以外を想定する必要がないことの当然の帰結であるが、HoUZK では複数台の HoUZK を一括して制御したいという要求がある。この要求に答えるには、複数台の HoUZK とこれらを制御する装置を、一対多通信 (制御側も複数台の場合は多対多) が可能な物理層に配置する必要がある。制御装置と HoUZK を、Bluetooth や ZigBee のような無線で接続した場合には、各々の HoUZK に取り付けられた Bluetooth アダプタや ZigBee アダプタに適切なアドレスを付与することで一対多通信が (ZigBee の場合には容易に多対多通信も) 可能になる。有線の場合には TTL - RS-232C コンバータの代わりに TTL 信号を RS-485, I2C, SPI などに変換するコンバータを用いることで、無線の場合と同様に一対多通信が可能になる。

5.3 HoUZK の今後

HoUZK は、UZK の成果を取り込んで作製した。HoUZK の非同期シリアルインタフェースには多様なアダプタを取り付けられるので、さまざまな用途に適用できることを示せた。著者は、現在の HoUZK に一応満足しているが、第三者がこれから HoUZK を作ろうとする場合には、MINI EZ-USB を購入して UZK のファームウェアを書き込み、非同期シリアルインタフェースに必要な通信アダプタを取り付けるといった一連の作業が必要になる。現在の HoUZK と同等な機能をもう少し簡単にできないかを検討した結果、Arduino UNO +

LUFA という組み合わせが候補になった。LUFA (Lightweight USB Framework for AVR) は、その名前が示すとおり、AVR マイクロコントローラに軽量の USB フレームワークを搭載する試みである。2010 年末に発表になった Arduino の最新機種である Arduino UNO には LUFA の成果を取り込むことができる。そのため Arduino UNO で USB HID を構成する試みがすでに始まっている。今後は Arduino + LUFA という構成の HoUZK も検討したい。

6. おわりに

本報告では、「非同期シリアルインタフェースを介してキーボードやマウスを遠隔操作する装置」である HoUZK を実装し、HoUZK を使えば、USB ホスト側から見ればただの USB キーボードや USB マウスにしか見えないさまざまな入力装置を実装できることを示した。HoUZK は、UZK という先行事例を取り込んで実装したが、今後は、Arduino UNO に LUFA の成果を取り込むといった方法も検討してゆきたい。

謝辞 HoUZK は、UZK という先行事例がなければ存在できなかった。UZK を開発し公開された uratan 氏に深謝する。本研究のうち、-ARK デバイスに関わる部分は、総務省戦略的情報通信研究開発推進制度 (SCOPE) 地域 ICT 振興型研究開発案件として平成 21 年度に新規採択されたプログラムに基づいて実施したものである。総務省および同省北陸総合通信局の関係各位に深謝する。

参考文献

- 1) 大野浩之, 松本文子, 山崎靖博: 高度情報通信危機管理研究施設の構築: (1)KVM スイッチを核としたオペレーション室の設計と実装, 第 29 回分散システム/インターネット運用技術 (DSM) 研究会, Vol.2003, pp.45-50 (2003-04-25).
- 2) 猪俣敦夫, 大野浩之: 乾電池でも運用可能な「非常時対応電子アーミーナイフ」(-ARK) を用いた非常時情報通信システムの実装, *Internet Conference 2008*, pp.15-24 (2008).
- 3) 大野浩之, 猪俣敦夫: 非常時を前提に開発した -ARK のシステム管理業務への応用, 第 48 回 分散システム/インターネット運用技術 (DSM) 研究会, Vol.2008, pp.49-54 (2008-03-06).
- 4) 藤枝和宏: ちょっと偏った PS/2 USB 変換アダプタ選び. <http://www.jaist.ac.jp/fujieda/ps2usb/>.
- 5) uratan: UZK ホームページ. http://optimize.ath.cx/mini_ezusb/.
- 6) オプティマイズ: MINI EZ-USB ホームページ. <http://www002.upp.so-net.ne.jp/uratan/UZK/>.