

medkit - Manually Embedded Device Konfiguration and Integration Technology

Project Proposal.

Name is negotiable

Motivation

Medical devices are extremely expensive, with a huge cost to the populace [cdc-medical-debt-source], including a very large population of Americans who do not seek, or adhere to, medical treatment as a consequence of prohibitive cost and the spectre of crippling medical debt [crna-paper]. In hopes of addressing this fact, many efforts have arisen to design low-cost medical devices [insulin pump, etc], and a whole host of low-cost sensors [wearables paper, etc]. These devices are difficult to put into production due to the high input cost often required to attain medical certification, and are often in an environment full of incompatible proprietary medical devices and sensors that may not have the possibility of interoperation, if they even have any hardware to do so.

To solve this problem, this document sets out two designs:

- **I) The Protocol** - a specification for communication over a network of connected medical devices.
- **II) The Bridge Device** - a device that, once manually installed on an arbitrary, well-specified host device and allows it to interact with the network by implementing The Protocol (I).

(I) The Protocol

The protocol specifies the interactions between several wearable medical devices, specifically three main categories:

- **Sensor Devices:** devices that measure biometrics and transmit them to the network.
- **Delivery Devices:** devices that can be commanded to perform a set of medical tasks in accordance with a delivery policy
- **Control Devices:** devices communicating delivery policy details and managing information accumulation and exchange across the network as a (locally) central point.

The primary communication over the network falls under these general categories (with exceptions):

- devices implementing **Sensor Device** behavior communicating with a **Control Devices** to send biomedical information
- devices implementing **Control Device** behavior communicating with a **Delivery Device** to change a delivery policy.

It is also recognized that it is unreasonable for a design to be adopted broadly by manufacturers, or previous designs lacking in the hardware to communicate with the design without major change in operation, which is costly, and unlikely. Instead, the protocol is designed to be easy to implement the hardware interfaces as software modules, which will allow an easier process of building an open source database of reusable device configurations. This concept will be leveraged in section (II), where a device design is proposed that will allow a large class of button-operated devices to be integrated almost natively.

Protocol Structure

To explore the organization of the network, we will chose the example of a wearable network designed for a person with diabetes. A common configuration would be the following:

```

        glucose data
<Sensor: CGM> -----> <Control: Delivery Policy> -----> <Delivery: Insulin Pump>
                                delivery guidance

```

A continuous glucose monitor connected to a user reports glucose levels to a controller. The controller monitors and adjusts its delivery response and communicates its guidance to the insulin pump. The insulin pump verifies that the insulin directive makes sense and adjusts its rate of delivery accordingly.

In certain phases of this network, however, it is clear that a delivery policy might want to query actual real-time delivery in its own modelling, in which case we end up with the following relationship:

```

        insulin delivery data
<Sensor: Insulin Pump> -----> <Control: Delivery Policy>

```

With this demonstration, it is clear that a very flexible protocol is needed, with devices not fitting into neat categories.

In recognition of this, The Protocol is designed as a mechanism for communication where devices can negotiate configuration options directly with a control mechanism, from broad biometric data to sampling rates and the abstraction of wearable medical hardware.

In an ideally interoperable configuration such as this, the drug delivery policy controller, insulin pump, and glucose monitor could all be developed separately and interchanged, allowing a modular and easily upgradable medical device system

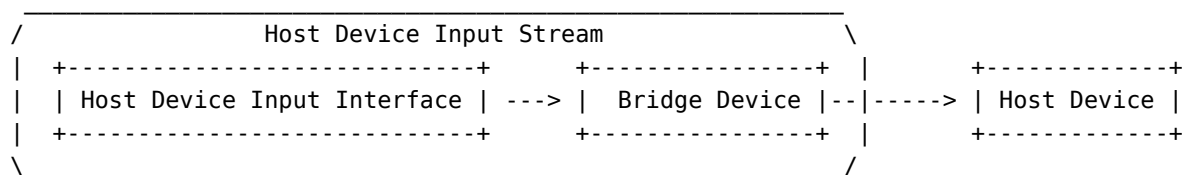
(II) The Bridge Device

The bridge device will be the focus of our demo, and represents a tangible implementation of the policy towards a useful problem, namely that of integrating a subset of non-protocol-supporting devices so that they can interact with devices that do implement the protocol. This is a very useful function of the device, as it is unattractive to design with a given protocol in mind if the vast majority of (if not all) devices that you require for the function of your device (i.e. who wants to write an glucose **Control Device** if there is no **delivery device** for insulin that you can interface with, and/or no **Sensor** to give you glucose readings).

To solve this, a design is proposed for a device that uses the GPIO capabilities of a common microcontroller to inject itself into the traditional input stream of the host device. Specific drivers could be uploaded to the device, or packaged with it through non-networked based flashing mechanisms that would “bridge” (hence the name) the device, implementing a subset of the full protocol features.

Crucially, full implementation of the protocol’s feature set is not required for a device to communicate over the network, and different devices must specify a minimum functional set of features for its proper operation.

Bridge Device Block Diagram



Considerations

Before an actual release could be made with this software, a much more robust cryptographic approach would obviously have to happen. The current risk of an attack where an attacker is able to get within close range bluetooth and drastically threatening a user's health and safety, or upload malicious drivers is far too great a risk to take without extensive analysis that extends beyond the current scope of the design project.

A few major precautions to take now:

- Specifying non-volatile constant limits in rate based on a given amount of time (i.e. you can't spike

the insulin if you can only increase between safe levels without manual adjustment in edge cases).