The Medical Device Dongle: An Open-Source Standards-Based Platform for Interoperable Medical Device Connectivity

Philip Asare[1], Danyang Cong[2], Santosh G Vattam[2], BaekGyu Kim[2], Shan Lin[3], Oleg Sokolsky[2], Margaret Mullen-Fortino[4], Insup Lee[2]

[1]Electrical and System Engineering Department, University of Pennsylvania, Philadelphia, PA, USA
[2]Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA, USA
[3]Computer Science Department, Temple University, Philadelphia, PA, USA, [4]University of Pennsylvania Health System, Philadelphia, PA, USA

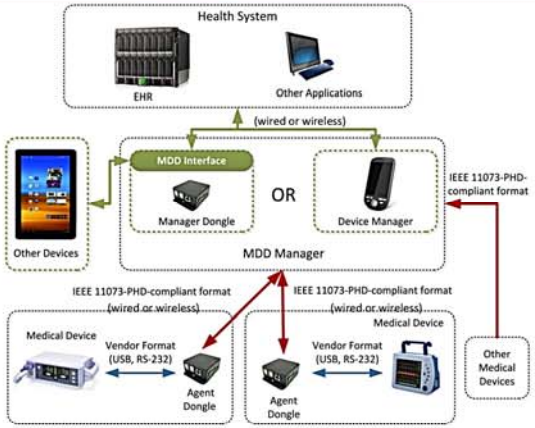
Introduction

- Emerging medical applications require device coordination.
- Most existing devices do not support wireless communication or network connectivity.
- Hospitals interested in new coordination applications must replace existing devices.
- Existing devices can be adapted for network connectivity.
- The Medical Device Dongle (MDD) is able to connect to any medical device that provides a data output port (USB or RS-232).
- We show how multiple medical devices can be connected and controlled using an open-source platform, standards-based connectivity and model driven software.

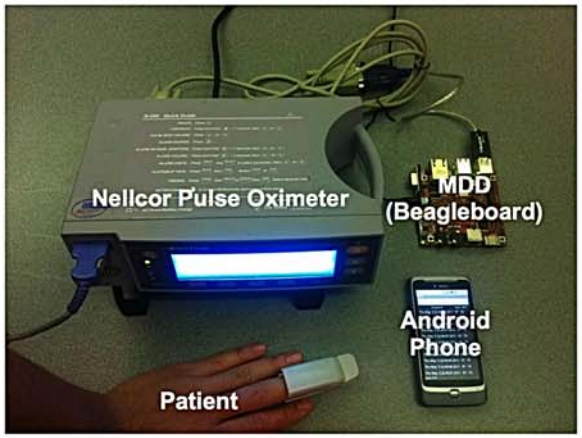
MDD Overview

- Applications of interest require that devices connect to a central point (supervisor).
- This necessitates two versions of the MDD: one for devices connected to the patient (agents) and one for the supervisor (manager).
- The MDD can be implemented in two ways:
(1) physically as a peripheral
(2) logically as software running on an existing device.
- The MDD is based on *a subset* of the IEEE 11073 protocol.
- The general architecture is based on 11073 Point of Care (PoC) and the connectivity and communication is based on 11073 Personal Health Devices (PHD).
- It supports other medical device and interoperability protocols, especially those geared towards patient safety.
- The MDD implements the following parts of 11073:
(1) Manager and agent finite state machines for maintaining connectivity
(2) GET and SET services for message exchange and command execution
(3) Medical device system (MDS) for device descriptions
(4) Medical device encoding rules (MDER) for message format.

General Architecture



Current Implementation

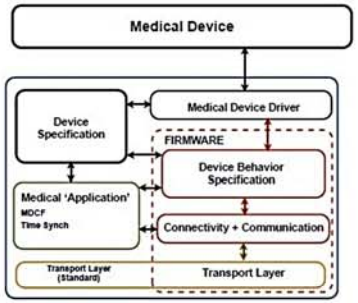


Current Work

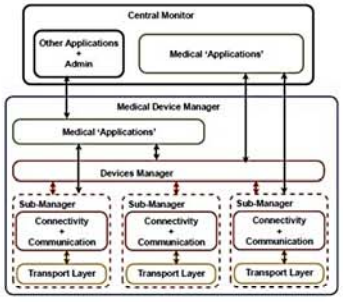
- *Validation of 11073 compliance.* Testing with known compliance testing tools and available certified 11073 devices.
- *Supporting more devices*. Developing device drivers for more devices. Currently working on ECG monitor and ventilator.
- *Providing guidelines for medical device driver development.* Allowing others to leverage our platform as well as contribute to our efforts in making devices with coordination capabilities accessible.
- *Interfacing with other protocols.* Allowing other coordination protocols to leverage MDD capabilities
- *Supporting development of medical applications.* Providing APIs for interfacing with MDD interface.
- *Enabling medical device interoperability research.* Allowing access to platform and code for researchers to investigate issues in interoperability.

Software Architecture

Agent



Manager



Acknowledgements

This research is supported in part by:

- NSF CNS-0834524
- NSF CNS-0930647
- NSF CNS-1035715
- NIH 1U01EB012470-01

Point of Contact:
Insup Lee (lee@cis.upenn.edu)

Information on MDD can be found at http://rtg.cis.upenn.edu/mddongle/

Future Work

- *Providing real-time guarantees over hospital network infrastructure.* Many applications of interest require real-time guarantees to ensure proper operation and patient safety.
- *Interoperability issues (safety and security)* Device connectivity turns devices in a patient room into a single system. Patient safety must be considered in this context with the potential for devices interfering with each other. Reliance on network (and in some cases over wireless channels) requires a look at security and privacy issues.