2016

Implementation of Local Transport Protocol Library (LTPlib) into Real-time Operating System (RTOS)

Sanam Mehta
University of Windsor

Implementation of Local Transport Protocol Library (LTPlib) into Real-time Operating System (RTOS)

By

SANAM MEHTA

A Thesis
Submitted to the Faculty of Graduate Studies
Through the Department of **Electrical and Computer Engineering**
In Partial Fulfillment of the Requirements for
The Degree of **Master of Applied Science**
At the University of Windsor

Windsor, Ontario, Canada

2016

Implementation of Local Transport Protocol Library (LTPlib) into Real-time Operating System (RTOS)

By

SANAM MEHTA

APPROVED BY:

____

Dr. A. Azab
Department of Mechanical, Automotive and Materials Engineering

____

Dr. E. Abdel-Raheem
Department of Electrical & Computer Engineering

____

Dr. K. Tepe, (Supervisor)
Department of Electrical & Computer Engineering

April 12, 2016

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication, and to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Healthcare is getting more expensive overtime. Personal telehealth systems, including remote patient monitoring and management, can facilitate caregiver to effectively deliver high-quality healthcare service at lower cost. The recent developments in information and communication technologies have increased the degree of connectivity between people using smart devices. To further enhance these developments, implementation of the Local Transport Protocol library is ported to a micro real-time operating system to achieve a low cost yet highly efficient embedded system.The selected hardware and software provide easy interface for data transfer from a monitoring and measuring device to remote locations. Targeting the Continua Health Alliance compliancy as the future task of this research and development work can be a significant contribution to the future of healthcare monitoring system.

DEDICATION

To my Parents,

Dr. Harshad Mehta and Mrs. Pallavi Mehta

For raising me to believe that everything was possible

&

To my Sister, Ashka Mehta

For encouraging me to make everything possible

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to my supervisor, Dr. Kemal Tepe, for his consistent support and guidance throughout the research. I have truly enjoyed working with him and I consider myself very fortunate to have got this opportunity. I would like to thank my committee members, Dr. A. Azab and Dr. Esam Abdel-Raheem, for their valuable input and guidance throughout the research process. I would like to extend my gratitude to my fellow laboratory members Dr. Brajendra Kumar Singh, Ms. Saneeha Ahmed, Mr. George Pantelimon, Mr. Michael Mati, Ms. Brinda Tank and Ms. Putul Saha for their constant involvement and valuable feedback. Finally, I wish to extend a special thank you to my loving family, especially my parents for always being there and believing in me. At last but not the least I would like to thank my friends for always supporting me in all my decisions and tolerating me during this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACL	Asynchronous Connection-Less
APDU	Application Packet Data Unit
API	Application Programming Interface
BBB	Beaglebone Black
BD	Bluetooth Device
BPM	Blood Pressure Monitor
BT	Bluetooth
BT SIG	Bluetooth Special Interest Group
BT-HDP	Bluetooth Health Device Profile
BT-LE	Bluetooth Low Energy
CCS	Code Composer Studio
CDG	Continua Design Guidelines
CESL	Continua Enabling Software Library

CM	Continua Manager
CRC	Cyclic Redundancy Check
DR-ZHG	Dual Radio ZigBee Home Care Gateway
DSP	Digital Signal Processor
E2E	End to End
ECG	Electro Cardiogram
FPGA	Field Programmable Gate Array
GAP	Generic Access Profile
GUI	Graphical User Interface
HDP	Health Device Profile
HRN	Health Record Network
HRN-IF	Health Record Network Interface
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
ISR	Interrupt Service Routine
L2CAP	Logical Link Control and Adaptation Protocol

LAN	Local Area Network
LAN-IF	Local Area Network Interface
LTP	Local Transport Protocol
MCAP	Multi-Channel Adaptation Protocol
MCL	MCAP Communication Link
MD	Medical Device
MDC	Medical Device Controller
MDEP-ID	Medical Device End Point ID
MDH	Medical Device Host
MDL	Mediated Data Link
MED WG	Medical Device Working Group
NFC	Near Field Communication
OEP	Optimized Exchange Protocol
OS	Operating System
PAN	Personal Area Network
PAN-IF	Personal Area Network Interface

PHD	Personal Health Device
POSIX	The Portable Operating System Interface
RTOS	Real-Time Operating System
SAR	Segmentation and Re-Assembly
SDP	Service Discovery Protocol
SPP	Serial Port Profile
SW	Software
TAN	Touch Area Network
TAN-IF	Touch Area Network Interface
TI	Texas Instruments
TPDU	Transport Protocol Data Unit
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WAN	Wide Area Network
WAN-IF	Wide Area Network Interface
WBAN	Wireless Body Area Network

WiCIP	Wireless Communication and Information Processing
WLAN	Wireless Local Area Network
WMG	Wireless Medical Gateway

1. INTRODUCTION

Technology has been an integral part of healthcare services. Recent advancements in computing, communication and networking technology are enabling healthcare practitioners to deliver timely and cost effective healthcare services using telehealth or e-health systems. Through the telehealth systems, remote monitoring of patient, consultation and training have been possible. This is advantageous to both patient and healthcare provider because it eliminates requirements of movement for both parties. There by, it saves time, effort and money. Because of these advantages, telehealth systems have been becoming popular and manufactures have started producing devices that are used in such telehealth systems. But the interoperability of such medical devices from different manufacturers has been a challenge as each manufacturer's implementation is unique because, till recently, there have been no standards defined for implementation of telehealth medical devices. Now, there exists an international non-profit group called Continua Health Alliance[1] and is setting design guidelines to standardize development of end-to-end, plug-and-play connectivity for personal, connected healthcare systems. The members of the Continua get access to its standards and specifications and certification of compliance for medical devices that are developed as per Continua guidelines. Based on Continua standards and specifications, many protocols are developed for the communication between different nodes of personal, connected healthcare systems [2]. One such protocol is, Local Transport Protocol (LTP) which defines interface between Medical Device Host (MDH) and LTP capable

Bluetooth (BT) module. In this thesis, the LTP protocol is designed and implemented on a POSIX Real-Time Operating System (RTOS) called UNISON running on a low cost embedded system acting as MDH.

1.1 Problem Statement

Currently, implementation of LTP exists only for Microsoft Windows applications [3]. Having implementations of LTP on embedded systems is desired because of the speed, cost effectiveness and other advantages that embedded systems offer.

Microsoft Windows is a generic operating system and does not offer the deterministic execution behavior which is needed in medical data collection. Thus having LTP running on a deterministic RTOS is equally desired.

The interface between the front end GUI of Microsoft applications that use LTP and the underlying LTP library is defined with shim interface standards [4]. The shim is an industry level interface standard defined by IBM for interfacing between different software modules. Using these interface standards, adds lot of memory and execution overhead and thus makes the protocol bulky. Typically, embedded systems have limited resources and it is necessary to avoid memory and execution overheads. To avoid such overhead, any proposed implementation needs to eliminate shim interface.

1.2 Thesis Contribution

In this thesis, an e-health communication protocol similar to LTP is designed and implemented on a POSIX RTOS running on a low cost embedded system. Unison is the chosen RTOS for this project. UNISON is designed and maintained by Rowebots Inc., which is the industry partner of Wireless Communication and Information Processing Lab (WiCIP Lab).

There are many low cost, yet powerful computing solutions available in the market. In this thesis, BeagleBone Black (BBB) [5] is selected as the computing platform. BBB is a small computer that has 1 GHz ARM Cortex A8 processors, runs UNIX OSes, has USB host and device controllers to connect to any plug-and-play hardware, can connect to internet and yet very cost effective. The implemented LTP protocol is tested on a medical gateway designed using BBB, a Stollmann Health Device Profile (HDP) BT adapter [6] and BT enabled Blood Pressure Monitoring (BPM) device.

In this thesis, the LTP is implemented without any shim interface to eliminate all the memory and execution overhead that is associated with shim interface standards.

1.3 Design and Implementation Requirements

The LTP protocol shall be implemented on an embedded system running an RTOS to benefit on the advantages of using an embedded system. The implementation shall be light weight by eliminating the overheads of using shim interface.

1.4 Thesis Organization

This thesis is organized into 4 chapters. Chapter 1 (this chapter), explains the problem statement and contribution of this thesis towards the solution. The related work in the field of health device communication protocols is discussed in Chapter 2. The standards applicable in the development of LTP, the design and implementation methodology and major challenges faced during implementation of LTP library, are discussed in Chapter 3.

How the protocol designed and implemented in this is used in development of a wireless medical gateway is explained in Chapter 4. Chapter 5 provides conclusion and scope of future work.

2. RELATED WORK

2.1 Introduction

Telehealth technology has evolved through years and since its inception. However, recent developments in networking and communication technology have given opportunities for engineering community to develop ever effective telehealth systems. The telehealth systems proposed in [7], [8], [9], [10], [11], [12], [13], [14] support different mechanisms of transfer of patient related information to a remotely located care giver. With no industry wide standards and guidelines available, most of these designs are very specific and are implemented with vendor specific hardware, software and interface protocols. For example, [8] and [13] use custom data transfer protocol over Ethernet and ZigBee based multi-hop communication respectively, to transfer data. A three layer application based medical information network model with customized network interface, application and operating system is proposed in [7].Another telehealth system, the Continua Manager [3] supported by the Continua Health Alliance, has industry standards and guideline based LTP. However, this application runs only on Microsoft Windows environment. This chapter explains LTP, CDG and other relevant standards that must be followed while implementing LTP library. This chapter also explains continua manager, its components and limitations.

2.2 Continua Health Alliance

The Continua health alliance is an international not-for-profit industry group and the leading organization convening global technology industry standards to develop end-to-end, plug-and-play connectivity for personal connected health. CDG based on global industry standards and test tools enable more efficient, standardized development and create new market opportunities for interoperable personal health devices and services used to collect and relay vital health information and education. Continua is a pioneer in establishing standards-based guidelines and security for connected health technologies such as smart phones, sensors, remote monitoring devices, tablets and gateways, as well as networked and cloud solutions [1]. Continua's Recommendation ITU-T H.810 [15] defines the CDG which contains specifications to ensure the interoperability of devices used for applications monitoring personal health. This guideline is based on IEEE 11073-Personal Health Device (PHD) communication standards and other communication standards like BT HDP [16] [17].The CDG contains guidelines for interoperability for different interfaces like Interface between Touch Area Network (TAN-IF) health devices and application hosting devices, Interface between Personal Area Network (PAN-IF) health devices and application hosting devices, Interface between Local Area Network health devices and application hosting devices (LAN-IF), Interface between application hosting devices and Wide Area Network (WAN-IF) health devices, Interface between WAN health devices and HRN health devices. Thus the guideline is divided into multiple clauses for implementing a medical device which can be certified by the Continua [18]. The applicable clauses for a specific device implementation have to be chosen by the developer.

The clause 1 to 5 of Continua guideline [18] mainly provides background information such as definition, supported interfaces and terminology necessary to understand the specifications. The clause 6 defines the End-to-End (E2E) system architecture of the Continua ecosystem. It describes, different components used in the system, topologies and compatibility requirements that have to be considered while designing an interface. This clause also defines the security requirements applicable to different interfaces. For the PAN-IF with BT HDP transport protocol, BT security standards are recommended and the implemented security shall cover the requirements of confidentiality, integrity and authentication.

The clause 7 of the CDG lists the design guidelines common to TAN, PAN and LAN interfaces with respect to the different connection technologies such as BT HDP, USB, Zigbee, NFC and BT LE. This clause states how different standards have to be chosen based on the interface type as shown in Figure 1.

In this thesis, guidelines applicable to the PAN-IF with BT HDP communication are followed. These standards are IEEE 11073-10407, -20601: Optimized Exchange Protocol (OEP) [18] and BT Health Device Profile [16]. Clause 8 of the guideline is about the TAN-IF, which is not in the scope of this research work.

Clause 9 of the CDG lists design guidelines specific for interoperability across certified

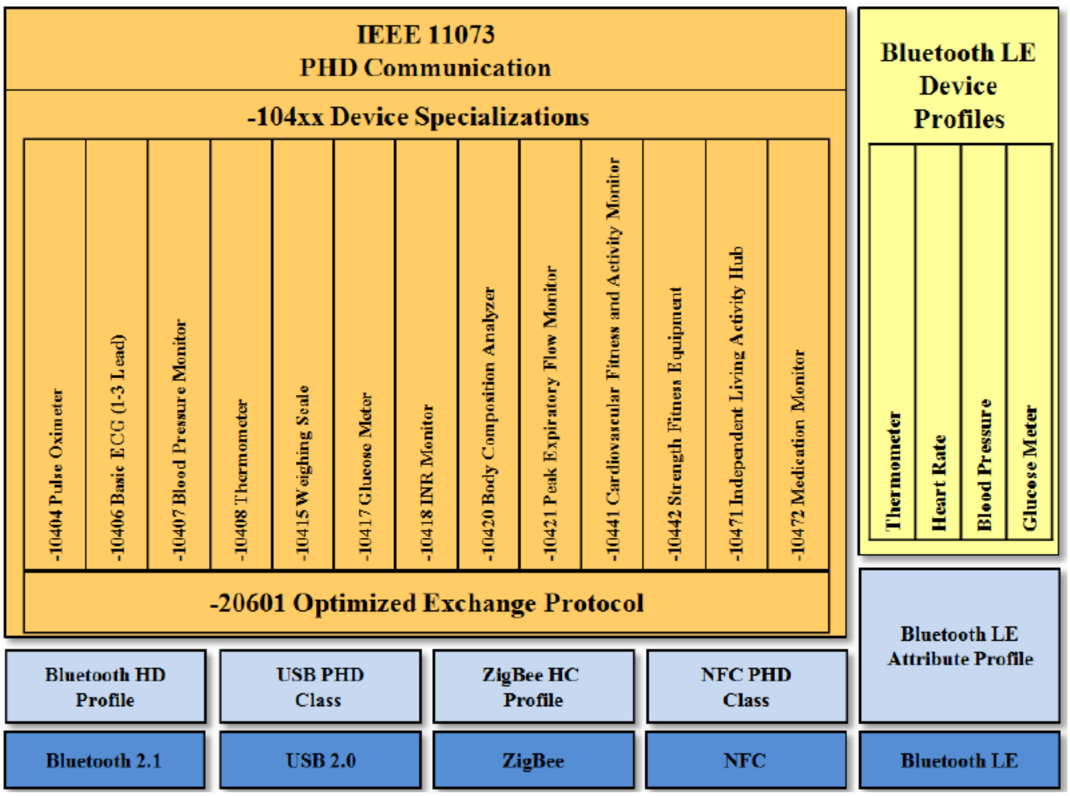

Figure 1 - Continua Protocol Stack

Continua devices in the personal area network interface. Guidelines in this clause recommend selection of USB personal healthcare device class or BT HDP or BT Low Energy (LE) services and profiles as transport protocol for the PAN-IF. For the BT HDP transport protocol designed in this thesis, requirements for device discovery, selection, pairing, data exchange and disconnection are listed in this clause.

2.3 Health Device Profile

BT is vastly used as a near-distance wireless technology in many applications and is very suitable for many medical device applications. Until 2008, the Bluetooth systems for medical application used proprietary implementations and data format. In most cases applications that run on top of the Serial Port Profile (SPP) were used. Such systems are – if they don´t come from one supplier – non-interoperable. Since the implementation is customized for just one vendor and/or device, data exchange between such systems is often difficult. Even BT interoperability with PC´s using different BT stack versions from different vendors is hard to achieve. To solve such issues, in 2008, the Bluetooth SIG released the Bluetooth HDP [19]. The BT-HDP supports a variety of in-home medical applications. The most typical use cases are portable sensors like oxygen saturation data transmitters, blood glucose level meters and blood pressure monitors that transmit the measurements to a monitoring system [10]. BT-HDP provides a wireless discovery method to determine device-type and supported data-type. BT-HDP uses the handshaking procedure for data and control channel establishment, link maintenance and retransmission of erroneous packets [10]. Other advantages of HDP are flexible data channel configuration, clock synchronization and resource optimization for constrained devices.

2.4 IEEE 11073-20601 and IEEE 11073 -10407 Standard

The IEEE 11073 is a family of standards established by PHD working group of IEEE to enable interoperability of manager and agent Medical Devices in telehealth systems. Among this family of standards, the IEEE 11073-20601 standard [20] defines an OEP [18] and modeling techniques to be used by implementers of PHD to create interoperability between device types and vendors. This standard establishes a common framework for transfer of personal health data over logical connections that are independent of underlying transport layer and transport layer protocol.

The IEEE 11073-10407 [21] establishes definition of communication between personal telehealth BPM devices and compute engines (e.g., cell phones, personal computers, personal health appliances, and set top boxes) in a manner that enables plug-and-play interoperability. While referring to the existing standards, 10407 specifies use of information models, application profile standards and transport standards for the data exchange between blood pressure monitor devices and manager devices/applications [21].

The designed LTP is compliant to all the necessary requirements of 11073-20601 and 11073-10407 [20] [21]. The information model, establishment of the logical connections, secure data transfer etc. are fully compliant to above described standards.

2.5 Local Transport Protocol

LTP [22] is a transport layer protocol designed to transfer data between MDH and MDC over an asynchronous serial interface. The need for the LTP comes from the fact that, medical applications expect only complete protocol packets are carried via a transport. For this purpose, the HDP allows application data frames up to a size of 64kBytes. However, most embedded "transport" solutions (e.g. BT modules) cannot store-and-forward this amount of data due to memory restrictions. So, HDP mandates use of Segmentation and Re-assembly (SAR) functionality of the lower BT layers [20]. SAR means, if there is large amount of data to be transported, the data shall be segmented in to multiple packets of protocol and re-assembled at the receiving end. So, HDP requires that packet boundaries of application data frames are maintained when they are transported via BT such that packets can be easily reassembled. LTP does this by making sure that once an LTP frame is started it will / has to be completed before any other action (e.g. indicating a disconnect event) is performed.

An LTP command is having following format.

<cmd><copmsk><lp>[P1][P2][P3]<payload>

Name	Size(in Bytes)	Description
cmd	1	Defines a command opcode that specifies the command included

		in a specific LTP packet. Each LTP packet includes exactly one command opcode
copmsk	1	Stands for “cmd optional parameter mask” and defines a field that is used to identify which optional parameters for the command are included in this LTP packet.
lp	2	Stands for “length packet” and defines a value that indicates the length of this LTP packet including all included headers, optional parameters and payload. This can be up to 64KB
P1,P2,P3	1 or 2	Optional fields. Which optional field is present in the given LTP message is identified by the copmsk
payload	N/A	Application data (APDU) contained in this LTP packet

Table: 1 – Software Packet Structure

The HDP mandates that all data exchanges have to happen on secure, authenticated and encrypted BT ACL connections. To handle the necessary security requirements, LTP defines messages such as User Confirmation Request, Authentication Request etc.

2.6 Components and Limitations of existing systems

The Continua developed and maintained CM windows GUI application uses LTP for communication with the MDC. Here, the CM GUI is developed in C++ programming language and the LTP library for interfacing with MDC (Stollmann adapter) is developed in C programming language. The C++ application interface to the C LTP library using the shim interface. The shim is an industry level interface standard defined by IBM for interfacing between different software modules. Using these interface standards, adds lot of memory and execution overhead and thus makes the protocol bulky and increases the delay in the communication. Typically, embedded systems have limited resources and it is necessary to avoid memory and execution overheads. To avoid such overhead, any proposed implementation needs to eliminate shim interface. In another implementation of the e-health system [23] a personal area network using ZigBee is developed. Different ZigBee enabled sensors such as thermometer, ECG Monitor, Heartbeat Sensor, are connected to a gateway developed using DR-ZHG. On the other side, the DR-ZHG connects to a Wi-Fi enabled broadband router using Ethernet interface to get connected to the global internet. Though this system has all the required elements of a telehealth system, the design is highly customized and doesn't support interoperability with other devices. This serious limitation is because of the fact that system doesn't follow any standard protocol for establishing the personal area network of different medical sensors.

2.7 Summary

This chapter described many telehealth systems and their limitations in the absence of industry standards and guidelines. This chapter provided an introduction to Continua Design Guidelines, other IEEE standards applicable for interfacing of different devices in telehealth system. This document also gave an introduction to LTP and HDP.

3. RESEARCH METHODOLOGY

3.1 Introduction

Implementing a Continua compliant LTP on the embedded systems running a RTOS has advantages of cost, speed and portability [24]. This chapter explains the details involved in the design and implementation of LTP on Unison RTOS [25]. In this research Unison RTOS is used since the project was in collaboration with Rowebots Inc. Unison is an ultra-tiny Linux and POSIX compliant embedded RTOS and is developed and maintained by Rowe Bots [26]. While implementing the protocol it is important to maintain the compliance towards all the applicable guidelines involved in the design of devices in personal connected health system[25] [26]. Since the protocol is implemented to run on embedded system, it is necessary to be lightweight in terms of processor and memory. Thus the end goals of the designed LTP protocol are:

1. To design and implement a Continua compliant LTP software library that can run on embedded system targets. The designed protocol shall be lightweight on processor and memory.
2. The designed LTP protocol library shall run on any POSIX compliant RTOS.

To be light on processor and memory, it is necessary that implemented LTP protocol be free from shim interface that other implementation of LTP are following. This is taken care by consciously avoiding use of shim interface standards wherever applicable. This chapter explains, in detail, the design and implementation methodology followed in achieving the above goals.

3.2 The Design and Implementation Methodology

The design and implementation of LTP SW library is carried out in following phases

1. Understanding and adaptation of standards applicable for implementation of a LTP
2. Design and architect the protocol
3. Implementation of SW library

3.1.1 Understanding and adaptation of standards applicable for implementation of a local transport protocol

In this thesis, the transport protocol is implemented over BT wireless communication link. Whenever BT is used as communication technology in telehealth devices it is necessary to use HDP [27] of BT specification [28]. The HDP is an extended specification of BT Core Specification 2.0 to be used for the implementations of devices used in telehealth system [29] [30]. This profile is used for connecting application data source devices such as BPMs, weight scales, glucose meters, thermometers, and pulse oximeters to application data sink devices such as mobile phones, laptops, desktop computers, and health appliances without the need for cables. Continua recommendation is that in addition to following the HDP for data transfer over BT connection, the device implementers must also follow the IEEE 11073 data exchange protocol and device specific standards for formatting of the data at the application layers of the communication protocol[27] [28] [29]. The development scope and standards coverage of LTP is as shown in Figure 2.

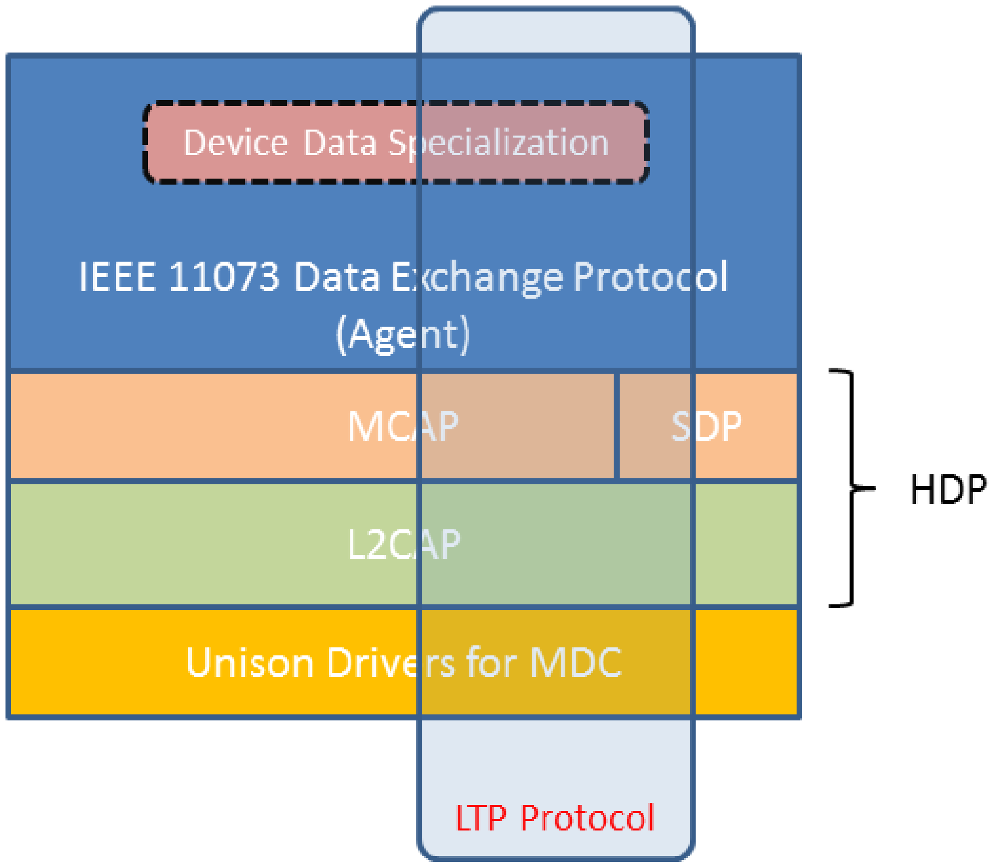

Figure 2 - Development Scope and Standards Coverage of LTP

3.1.1.1 HDP Protocol Model

The HDP makes use of the Multi-Channel Adaptation Protocol (MCAP) [31] and new Logical Link Control and Adaptation Protocol (L2CAP) features such as Enhanced Retransmission Mode, Streaming Mode and optional Flow Control [32] to define the interoperability requirements. The designed LTP can be used in either source or sink devices to add BT capability to the device. The LTP layer acts a glue layer of protocol between the application and physical BT layer. The Figure 3 shows the complete protocol stack as applicable to source and sink devices.

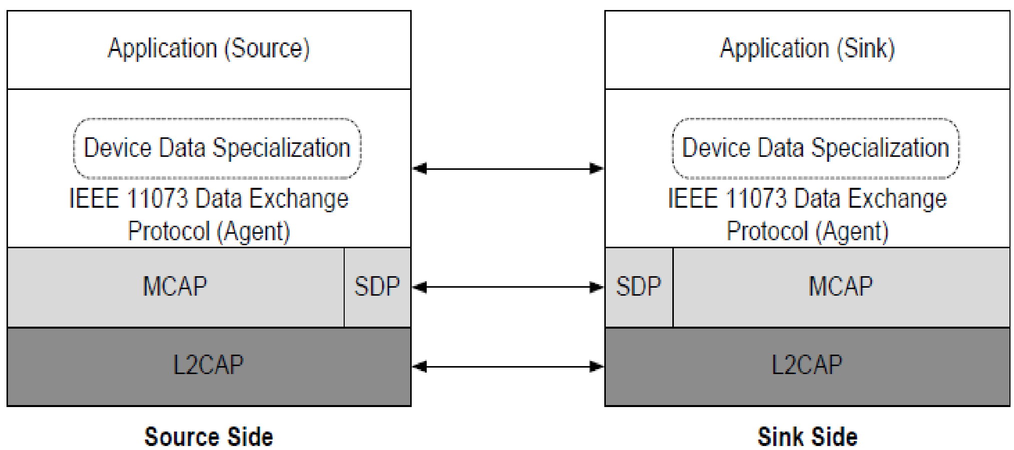

Figure 3 - Protocol Model

The MCAP defines the method by which data connections are established between two devices. Source refers to a source of data defined by the Data Exchange Specifications, and a Sink is a receiver for that data. The source may generate that data from sensors, or may relay data actually collected by some other device. The sink may be a display unit, a store-and-forward intermediary, or any other consumer of data exchange specification data. While the difference between source and sink is supported through flags that may act as indicators to the data exchange specifications, sources and sinks have the same observable behavior at the MCAP level. In MCAP, A control channel is used to facilitate data channel creation by establishing the desired logical endpoint for a new connection, or state reference for re-establishment of an old connection.

The control channel is the first L2CAP channel established between two instances of MCAP. This channel facilitates the creation of Data Channels, over which actual data (as defined in the data exchange specifications) can be exchanged. Data channels can be bi-directional, and carry data useful to higher layers. The control channel carries commands triggered by higher layers, but does not carry any data other than the commands defined in this protocol specification. The devices are “connected” (with respect to MCAP) when a control channel is established between those devices. So, as long as the control channel remains connected, the two devices will remain connected.

The control channel is used to identify the logical endpoints referred to as MCAP Data End Points (MDEP) to be used by a new data channel. Each MDEP is allocated an identifier (MDEP ID) that is locally unique to that MCAP instance, and can be used to refer uniquely to the corresponding function. Normally, MDEP IDs will be shared via the Service Discovery Protocol (SDP) by a profile using MCAP.

The data link established between two MDEPs as a result of a request from the control channel is referred as MCAP Data Link (MDL). Each MDL has an identifier (MDL ID) that is unique for the pair of devices involved. The MDL ID is very important in the context of reconnection operation. A MCAP Communications Link (MCL) identifies the full collection of L2CAP connections between two instances of MCAP, comprising a Control Channel and zero or more Data Channels. The channels formation and connections between data source and sink in BT communication using HDP are as shown in Figure 4.

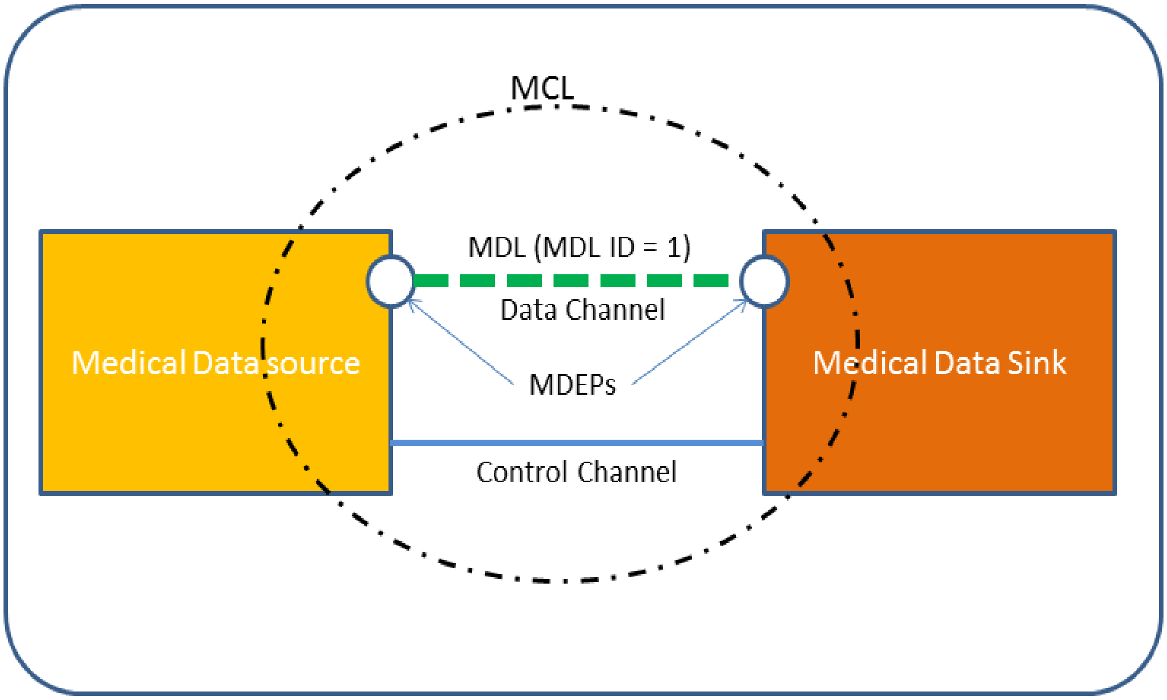

Figure 4 - Channel Formation and Connections

3.1.2 LTP Protocol Design

3.1.2.1 HDP Connection Establishment in LTP Protocol

In the implemented LTP, the establishment of the connection between a medical data source and sink involves MCAP control channel creation, identification of MDEPs and establishment of data channels. The request for connection establishment can get initiated from either side. The sequence of establishing HDP connection when the request is made from the medical data source device is as shown in Figure 5. To allow a remote device to discover and/or connect to the local device, the local device must be made “visible” and/or “connectable” on the air interface. In this context “visible” means that the local device can be discovered by an inquiry procedure of the remote device and “connectable” means that the local device at least allows a remote device to establish a point-to-point connection and responds to HDP connection attempts. The “visibility” and “connectability” of the local device can be controlled with the LTP command “RadioModeSetReq”. To allow a remote device to connect on HDP profile level, a corresponding MDEP that provides all information necessary to access these profiles has to be registered. This service registration procedure can be performed with the RegHDPMDEPReq messages. The first indication for an incoming HDP the target specific informal ACLStatusInfo and MCLStatusInfo messages, the CreateMDLInd message that has to be confirmed with a CreateMDLConfmessage. If the MDL creation is completed successfully, a ConnectMDLInfo message will be generated by the MDC that includes all necessary information to exchange data in that new MDL. If the MDL creation was not successfully completed due to reasons of the local or remote device, the newly created MDL will be indicated to be deleted with a DeleteMDLInfo message.

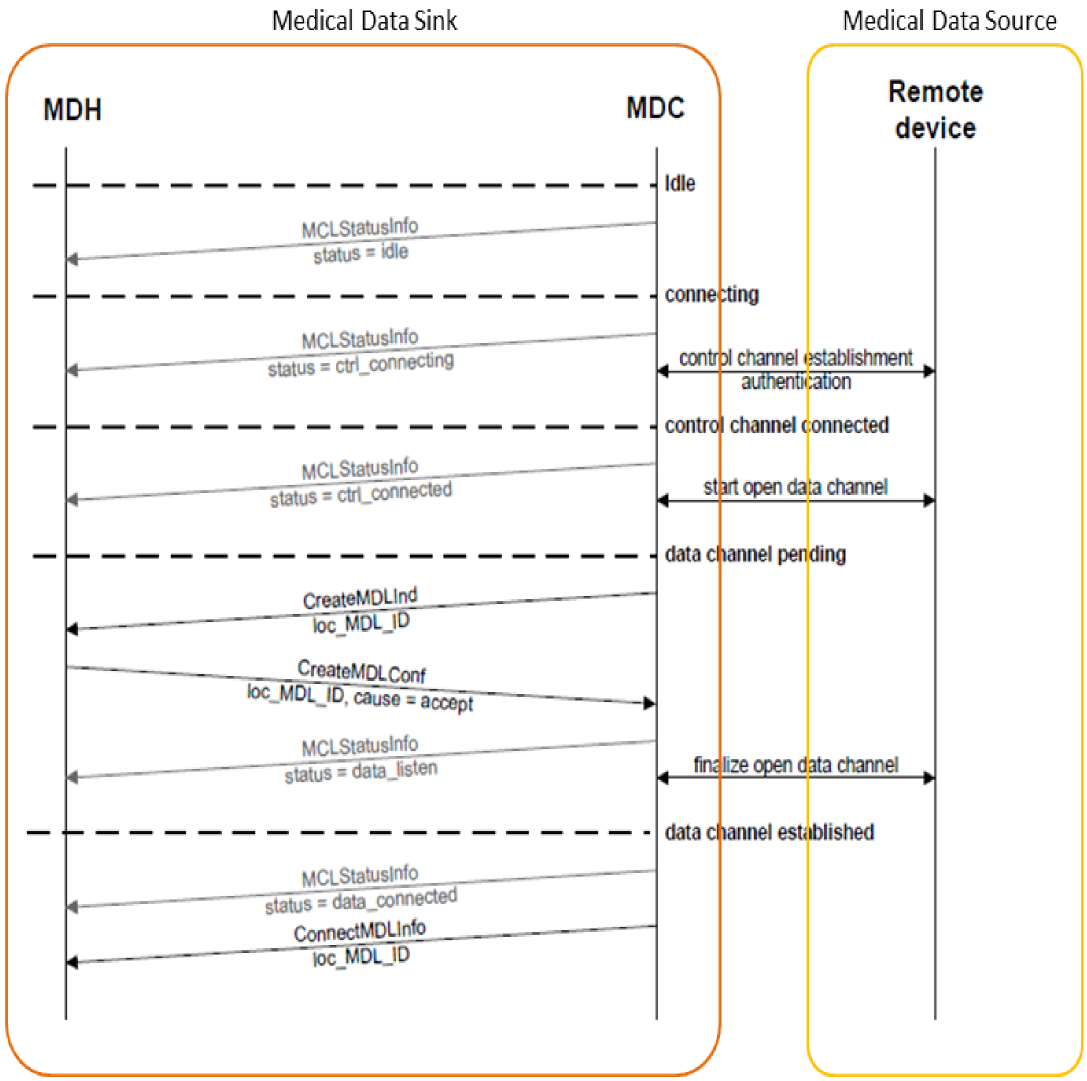

Figure 5 - Incoming HDP Channel Establishment

When the connection establishment request is initiated from the medical data sink, the remote device must be discovered first. This discovery procedure can be initiated with the InquiryReq message by the MDH. The sequence of this procedure is shown in Figure 6. The MDC performs a search for remote BT devices in the operation range and if the device is found, it will respond with the device information through one or more InquiryDeviceInfo messages. It also indicates the success or failure of the InquiryReq through an InquiryRsp.

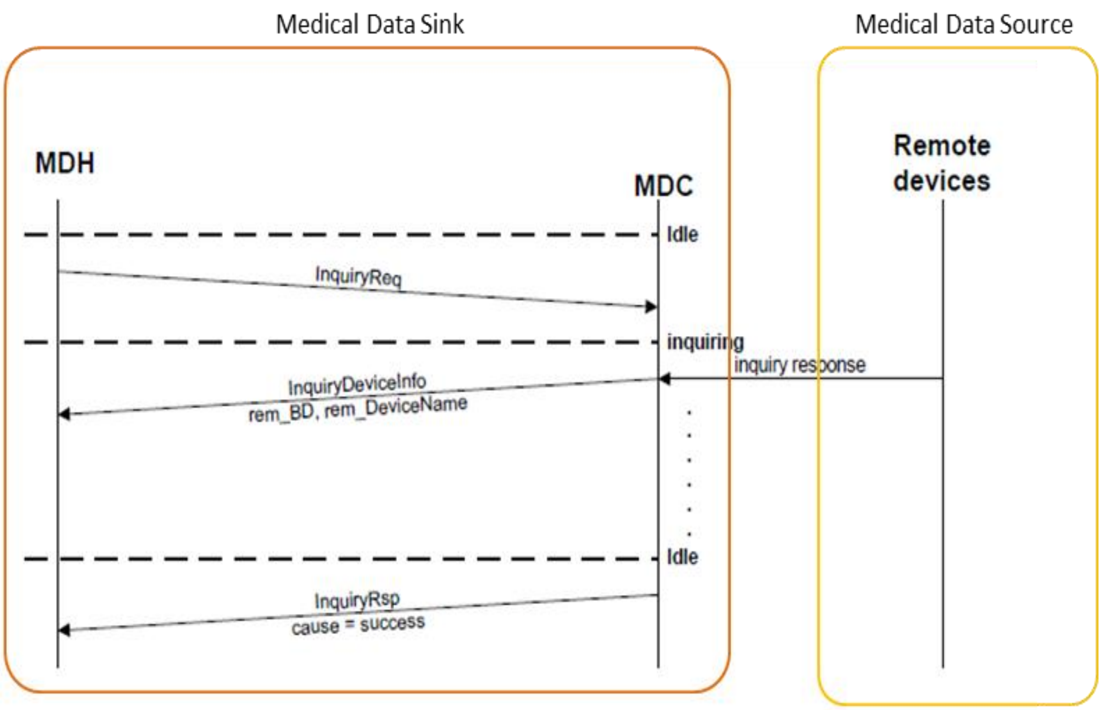

Figure 6 - Remote Device Discovery Procedure

Once a remote device is discovered or known otherwise, a service discovery has to be performed to gather all information necessary to perform a profile level connection to that remote device. This service discovery procedure can be initiated with the HDPDiscoveryReq message. The sequence of this procedure is shown in Figure 7. Once anHDPDiscoveryReq is received by the MDC, the MDC will perform a service discovery on the remote device identified by the <rem_BD> [6] [22] field. If the remote device supports HDP, all PnP specific information of that remote device will be indicated to the MDH using a DIDDeviceInfo message, all IEEE 11073-20601 relevant information of the device will be indicated with one or multiple HDPServiceInfomessages, and all data-endpoint relevant information of the device will be indicated with one or multiple HDPEndpointInfo messages. The success or failure of the HDPDiscoveryReq is indicated to the MDH through anHDPDiscoveryRsp message.

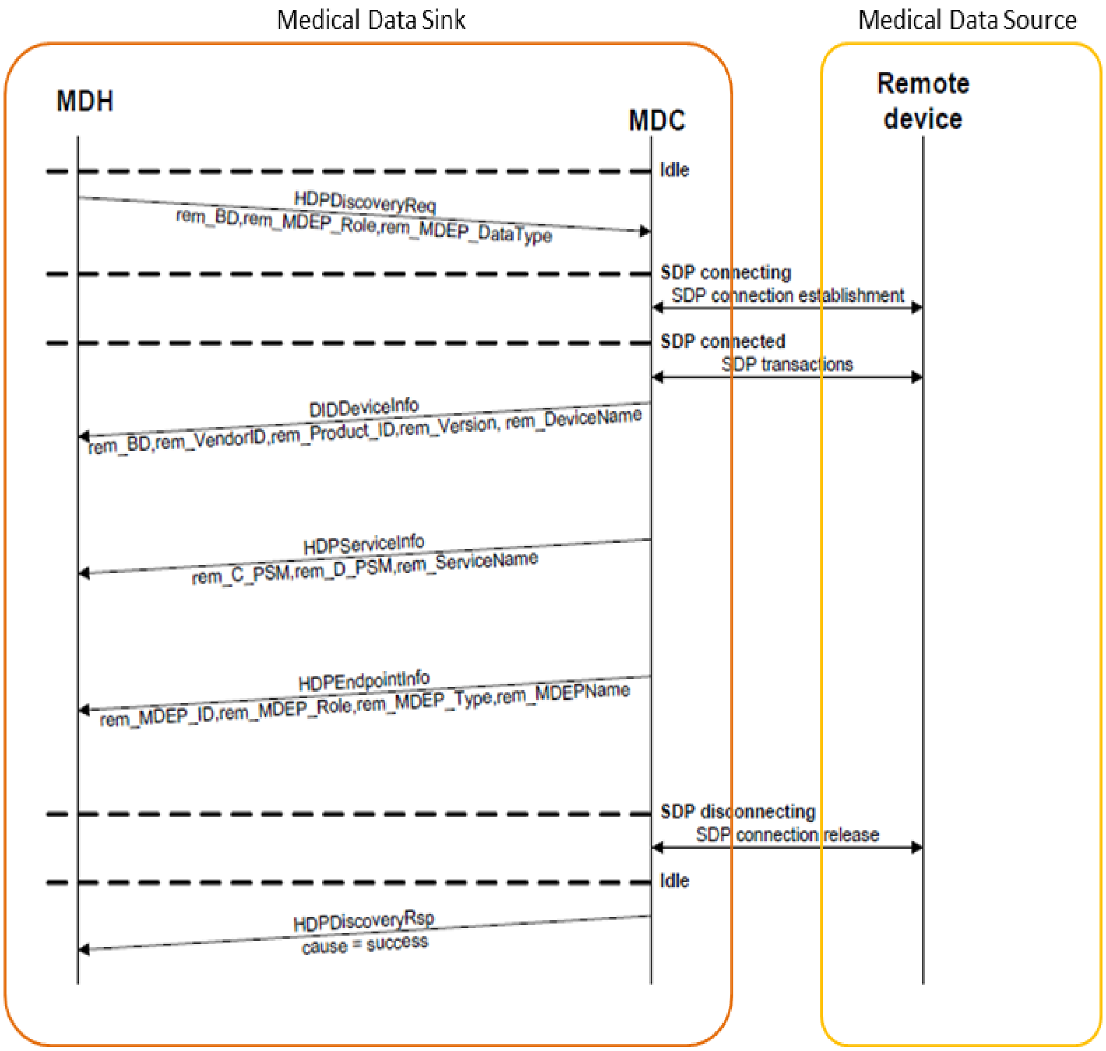

Figure 7 - Service Discovery Procedure

To initiate a HDP profile level connection for data exchange, the information gathered in the above procedure can be used to initiate a ConnectMDLReq message. The sequence of this procedure is as shown in Figure 8. Once a HDP MDL creation setup is initiated, the basic procedures to setup that new MDL connection are equal to the ones described for establishing incoming HDP connection with the only difference that at one point in time a ConnectMDLRsp message is generated by the MDC as a response to the initiating ConnectMDLReq message. Depending on the result of the profile level connection attempt this ConnectMDLRsp message will be either generated as a direct response to the initiating ConnectMDLReq message (e.g. remote device is out of range) or following the CreateMDLInd/Rsp procedure but before the terminating ConnectMDLInfo messages.

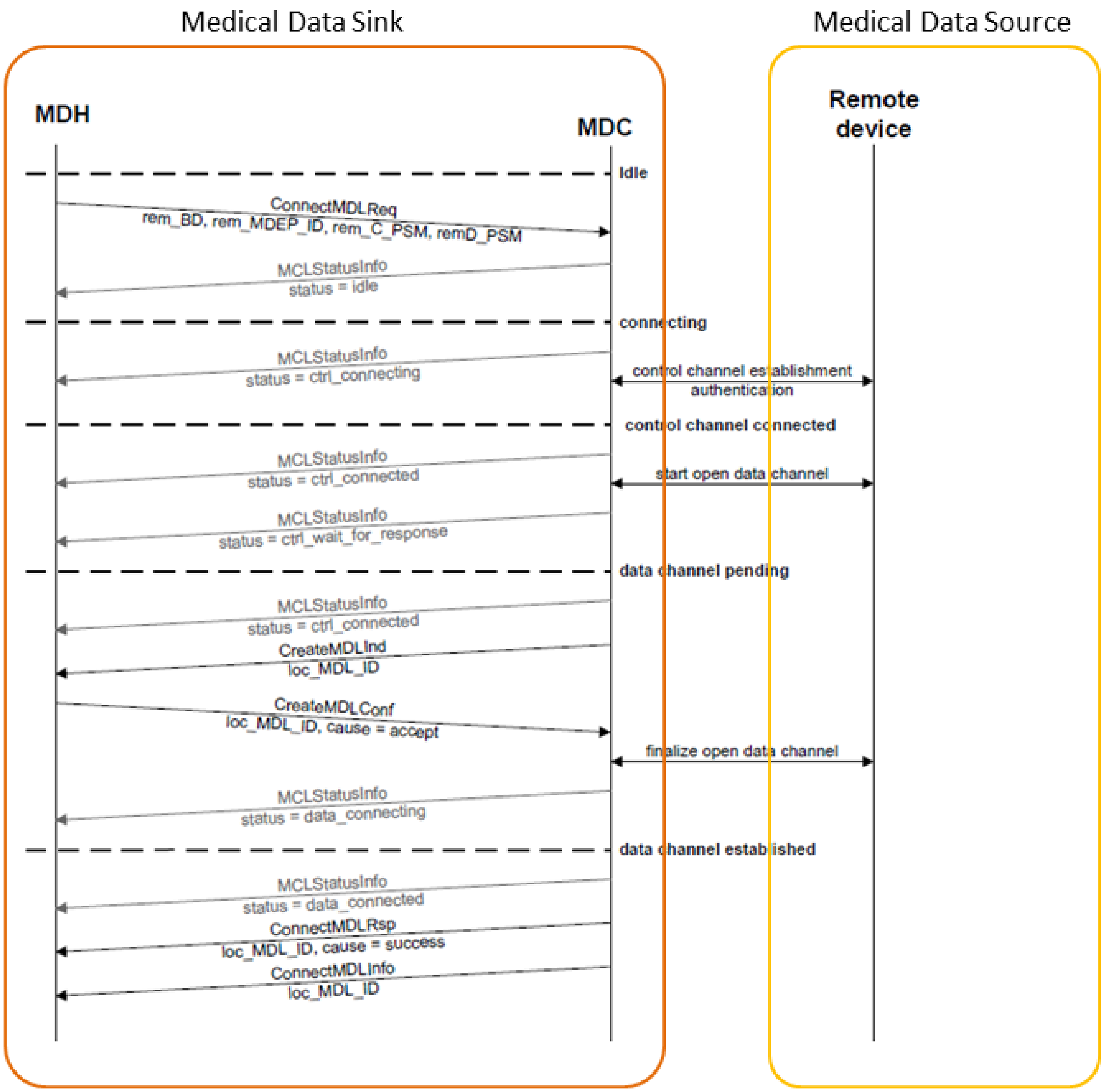

Figure 8 - HDP Profile level Connection Establishment in LTP

3.1.2.2 The APDU Data exchange

Continua standard mandates that the telehealth medical device implementers use IEEE 11073 Application Protocol Data Unit (APDU) standards for the exchange of data at the application layers of e-health communication protocols. For this purpose, HDP allows application data frames up to a size of 64kBytes. However, most embedded "transport" solutions (e.g. BT modules) cannot store-and-forward this amount of data due to memory restrictions. So, HDP mandates use of Segmentation and Re-assembly (SAR) functionality of the lower BT layers. Once a ConnectMDLInfo message is received, the signaled MDL connection is ready to exchange data but expects compliance to the MDL parameters indicated with the ConnectMDLInfo message especially regarding APDU size, Transport Protocol Data Unit (TPDU) size and flow control. To send such large 11073 APDU, that APDU has to be split into multiple data exchange messages that indicate if they provide the first data of a new APDU (start segment) continuation data of an already started APDU (continuation segment) or the last data of an already started or continued APDU (end segment). A pictorial representation of this segmentation is shown in Figure 9. Received large APDUs will be exchanged in segments with the same mechanisms. At the receiving side, the all segmented APDUs are assembled to form the complete APDU before passing on to the application layer. The data exchange request is started by medical data sink by sending the DataUnsegmented or DataStartSegment message based on whether the APDU of the current LTP command is being sent unsegmented or with multiple TPDU segments. When the data exchange is completed, a disconnect request is sent from the medical data sink. When the data exchange is initiated from the remote device (medical data source), DisconnectMDLInd message is received from the remote device. The complete sequence of communication for outgoing and incoming messages is shown in Figure 10 and Figure 11 respectively.

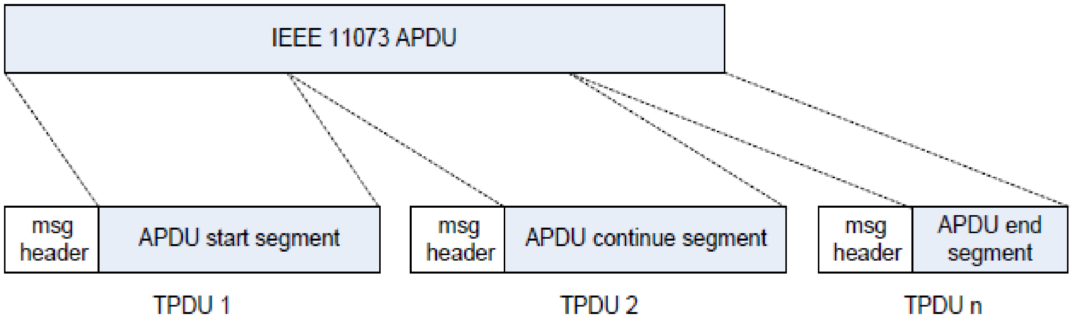

Figure 9 - APDU segmentation

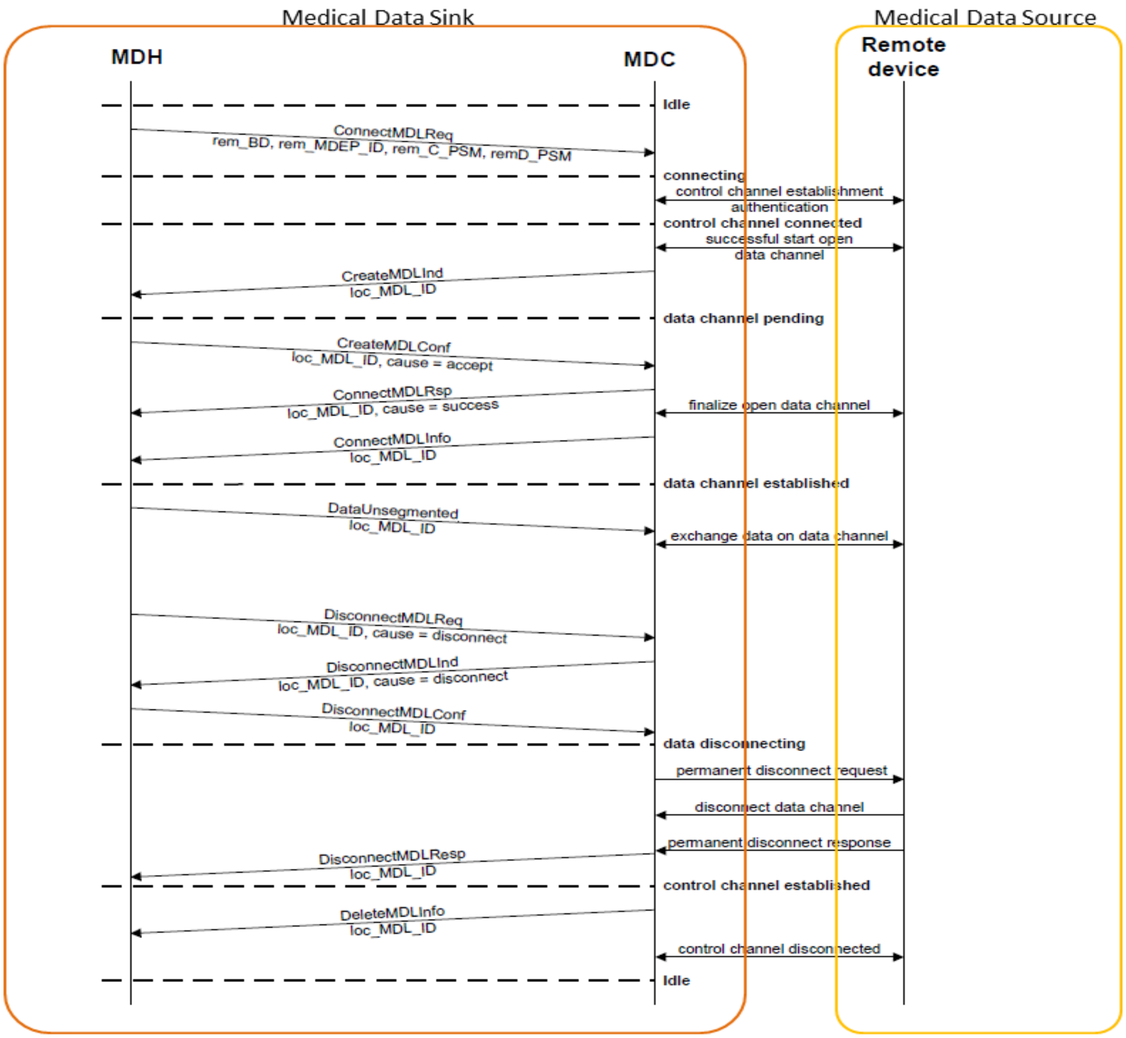

Figure 10 - Outgoing Connection Establishment with Data Exchange

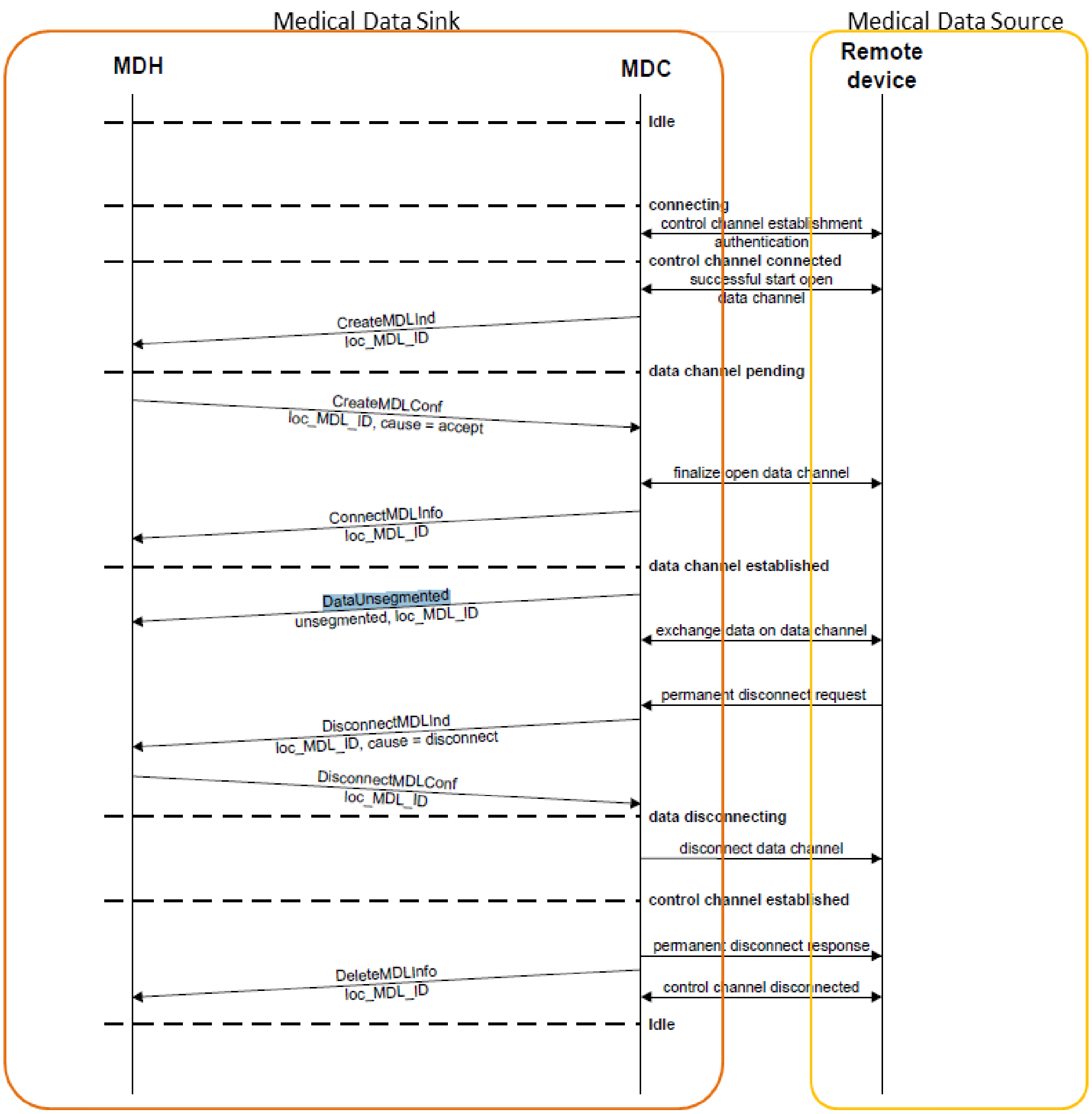

Figure 11 - Incoming Connection Establishment with Data Exchange

3.1.2.3 Message CRC checking

Each LTP message includes an optional CRC8 value that will be used by the MDC to check the LTP message header for correctness. If a CRC8 is part of an LTP message directed to the MDC and the message header is detected to be invalid by calculation of its CRC, the MDC assumes that communication with the MDH is corrupted and is out of sync. MDC then initiates an internalEventInfo message with event type

LTP_CommunicationOutOfSync and cause LTP_CauseConnectionLost. The CRC generator polynomial used in LTP implementation is x**8 + x**2 + x + 1.

3.1.3 Implementation of software library

The LTP SW library is written in C programming language using system calls of UNISON [25] [26] real time operating system. Unison is an ultra-tiny Linux and POSIX compatible embedded RTOS that is developed and maintained by RoweBots. It supports many general purpose microcontrollers, Digital Signal Processors (DSPs), FPGAs [33].Unison offers a single process, multiple threads, POSIX compatible embedded operating system with real time performance. Unison ensures multitasking and threads scheduling by employing round robin scheduling method, pre-emptive dynamic multithreading and cooperative multi-threading and poison pill approach for thread termination. Unison uses Interrupt Service Routines (ISR) to handle hardware interrupt requested by the processors [26]. The software architecture diagram of the Unison and the visualization of integration of LTP SW library is as shown in Figure 12.

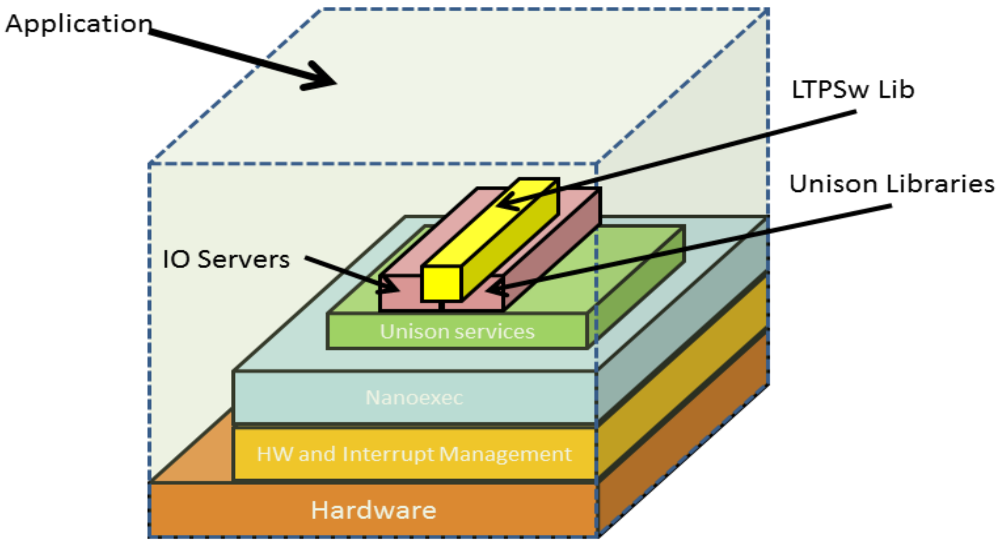

Figure 12 - Unison Software Architecture and LTP SW Library Integration

3.1.3.1 Implementation of LTP APIs

A list of LTP commands and the corresponding Application Programming Interfaces (APIs) that have the code implementation are listed in Table . Each API is written in C programming language and uses Unison system calls.

Table 2 - LTP Commands and Corresponding APIs

Command	**Description**	**Direction**	**LTP Library API**
Act Info	LTP Active Information Command. This command will be used by the MDC to indicate the activation of the LTP protocol handler	MDH←MDC	BTLTPTgtHandleLTPMessage function, checks if the ActInfo command is received. If yes, it sends the next command i.e. Register HDP/MDEP Request.
Register HDP/MDEP Request	This command is used by the MDH to request a HDP MCAP Data End Point	MDH →MDC	LTPLibSendRegisterHDPMDEPReq

Command	Description	Direction	LTP Library API
	(MDEP) entry in the SDP record of the MDC to be created. The MDC will respond with a RegisterHDPMDEPRspmessage specifying if the operation could be completed		
Register HDP/MDEP Response	This is the response by MDC to MDH for the Register HDP/MDEP Request command	MDH← MDC	BTLTPTgtHandleLTPMessage function, checks if the correct response to Register HDP/MDEP Request is received. If yes, it sends the next command i.e. Radio Mode Set Request.
Radio Mode Set Request	This command is used by the MDH to control the functionality of the local device on Bluetooth radio level.	MDH → MDC	LTPLibSendRadioModeSetReq
Radio Mode Set Response	This command is the response to Radio Mode Set Request	MDH← MDC	LTPLibSendRadioModeSetRsp
Pairable Mode Set Request	This command shall be used by the MDH to set the pairable mode of the local device.	MDH → MDC	LTPLibSendPairableModeSetRsp
Pairable Mode Set Response	This command is the response to Pairable Mode Set Request	MDH← MDC	LTPLibSendPairableModeSetRsp
Config Tunnel Request	This command is used by the MDH to access the configuration database of the MDC by initiation of configuration commands [22]	MDH → MDC	LTPLibSendConfigTunnelReq

Command	Description	Direction	LTP Library API
Config Tunnel Info	This command is the response to Config Tunnel Request	MDH← MDC	LTPLibSendConfig TunnelRsp
Inquiry Request	When an InquiryReq is received by the MDC, the MDC will perform an inquiry, indicate remote devices found with InquiryDeviceInfo messages and indicate the finalization of the inquiry process with an InquiryRsp.	MDH → MDC	LTPLibSendInquiry Req
Inquiry device Info	This command is used by the MDC to indicate a remote device found during an inquiry process to the MDH	MDH← MDC	LTPLibSendInquiry DeviceInfo
HDP Discovery request	This command is used by the MDH to request HDP/MCAP relevant information from a remote Bluetooth device in range.	MDH → MDC	LTPLibSendHDPDi scoveryReq
DID Device Info	This command is used by MDC to indicate the remote device's identification	MDH← MDC	LTPLibSendDIDDe viceInfo
HDP Service info	This command is used by MDC to indicate that the remote device specified in anHDPDiscoveryReq supports HDP and includes all service specific information for a service of that remote device.	MDH← MDC	LTPLibSendHDPSe rviceInfo

Command	Description	Direction	LTP Library API
HDP Endpoint info	This command will be used by the MDC to indicate that a remote device that was targeted by a HDPDiscoveryReq supports the HDP and includes all endpoint specific information for one endpoint supported by the service that was indicated via the last HDPServiceInfo message generated by the MDC	MDH← MDC	LTPLibSendHDPEndpointInfo
ACL status info	The purpose of this message is to inform the MDH about the status of a given ACL connection. Depending on the given Platform, only a subset of status changes will be indicated.	MDH← MDC	LTPLibSendACLStatusInfo
Connect MDL request	This command shall be used by the MDH to request creation of a new MDL connection to a specific MDEP_ID of a remote device by the MDC.	MDH → MDC	LTPLibSendConnectMDLReq
MCL status info	The purpose of this message is to inform the MDH about the status of a given MCL connection. Depending on the given Platform, only a subset of status changes will be indicated	MDH← MDC	LTPLibSendMCLStatusInfo
Create MDL Ind	This command will be used by the MDC to indicate a creation of a new MDL to the MDH.	MDH← MDC	LTPLibSendCreateMDLInd
Create MDL Conf	This command will be used by the MDH to response to aCreateMDLInd from the MDC.	MDH → MDC	LTPLibSendCreateMDLCnf

Command	**Description**	**Direction**	**LTP Library API**
Data Unsegmented Req		MDH → MDC	LTPLibSendDataUnsegmentedReq
Data Unsegmented	This command can be used to transfer an unsegmented application data packet APDU via LTP that means that the whole APDU is included within this single LTP packet.	MDH←→MDC	LTPLibSendDataUnsegmented
Data start segment		MDH←→MDC	LTPLibSendDataStartSegment
Data End Segment		MDH←→MDC	LTPLibSendDataEndSegment

3.1.3.2 Implementation of CP210x USB Driver

The CP210x driver is a highly integrated USB-to-UART bridge controller providing a simple solution for updating RS-232 designs to USB using a minimum of components and PCB space. Offering baud rates aliasing and supporting additional data formats, the CP210x is USB 2.0 full speed function controller, USB transceiver, oscillator and asynchronous serial data bus (UART) with full modem control signals [34] [35]. As the UART interface is used to transport the local transport protocol, this protocol enables the MDH and the MDC to communicate with each other, establish and terminate connections and exchange data frames while a BT connection is established.

The read and write system call implementation of the CP210x driver was having the limitation of the maximum data transfer to 8 bytes. This limitation was overcome by accessing the buffer present in the read and write calls of the driver. This buffer was set to a maximum possible of 64 bytes by accessing this buffer in the user space of the implementation. Segmentation and Reassembly (SAR) functionality of LTP is adopted in the user space to receive the data that exceeds 64 bytes in chunks of 2 to 3 messages and then reassembling it to one complete message. The BPM needed a reference date and time to set its clock from the Unison system through data packets well within the initial communication after successful channel establishment. This modification is also done in the CP210x driver level function by reading the time from the BPM and setting the communication time referring to this read time.

3.3 Summary

The LTP software library implemented in this thesis is written in C programming language using standard POSIX system calls. So, it can be used on any POSIX compliant operating system. Continua recommended standards are followed in the design of every layer of the protocol. The BT HDP standards, IEEE 11073-20601 OEP standards and BPM device data specialization standards are followed in the design of this transport layer protocol. The communication between medical devices implemented as per these standards requires establishing control channel and data channels; control channel for the initial handshaking and data channel for transfer of medical data. The sequence of establishing these channels and data transfer are explained in this chapter. This chapter also explains the different components of the LTP SW library and methodology involved in implementing them.

4. TESTING THE IMPLEMENTED LTPLIB

4.1 Introduction

The implemented LTP SW library is tested by using it in implementation of a Wireless Medical Gateway (WMG) that receives blood pressure values from a Continua compliant BPM device over BT interface. Since the BPM device is Continua compliant, it uses BT health device profile for communication with other medical devices. This chapter explains how the LTP SW library is used in establishing communication between a MDH and a MDC in real time data transfer between two medical devices. This chapter shows how a telehealth system can be designed to use LTP SW library to achieve communication with all Continua compliant devices.

4.2 Telehealth System with Wireless Medical Gateway

The block diagram of the telehealth system with wireless medical gateway that uses LTP SW library to communicate with blood pressure monitor is as shown in Figure 13. Different components of the WMG are shown in Figure 13. The BBB hardware is used as the base development board for this wireless medical gateway. In order to add the BT HDP interface to the BBB, a BT-HDP based adapter is used. This adapter is connected to BBB through a USB interface. This adapter is manufactured by Stollmann Inc, and will be referenced as "Stollmann adapter" further in this thesis. Through the Stollmann adapter, the BBB can connect to the blood pressure monitor to receive the data read from

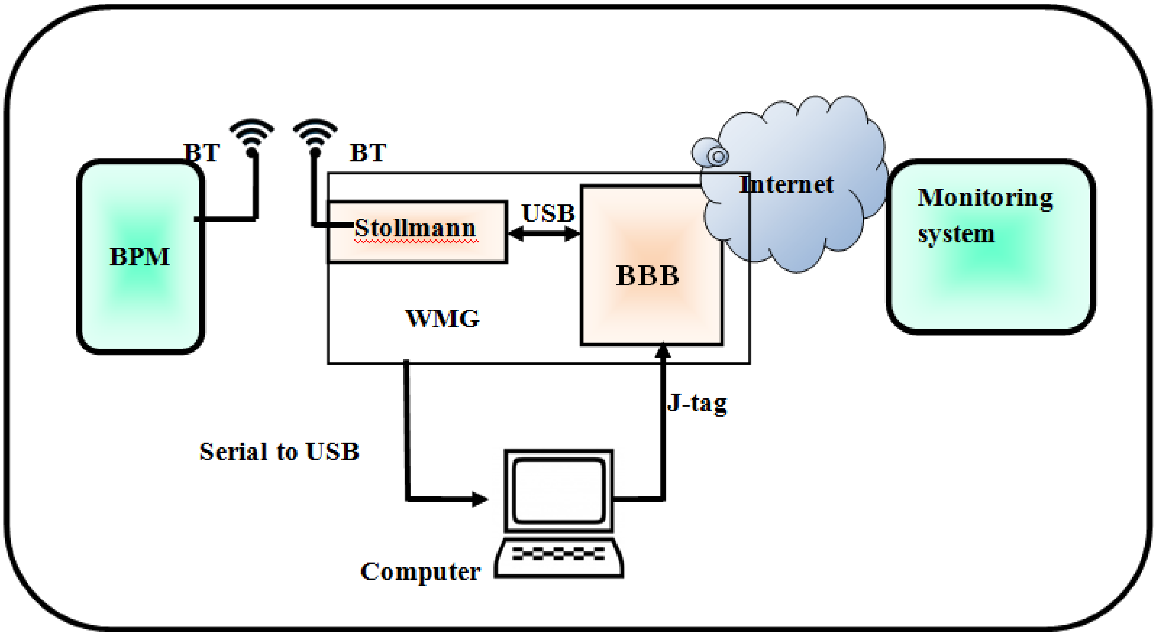

Figure 13 - Telehealth System with WMG

human body. A LAN cable is connected in the Ethernet port of the BBB which connects the gateway to the internet. The BBB has a surface soldered J-tag header to load the binary image of software that has LTP software library on to BBB. The BBB has a serial interface and can be connected to any host system though a serial to USB converter. This WMG is powered with 5V DC adapter.

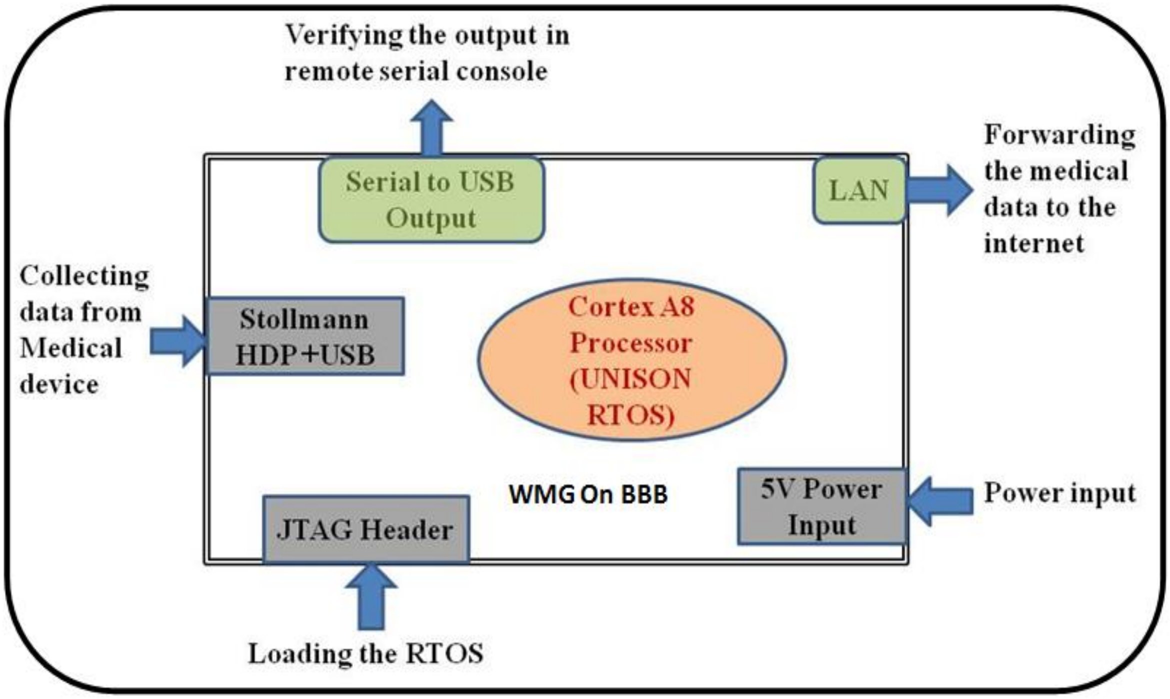

Figure 14 - Block Diagram of WMG

4.2.1 Beagle Bone Black

The BBB is a miniaturized energy efficient computer with high computational capabilities [36]. BBB is a low cost development platform, so, was appealing in selection as the implementation platform for implementation of this wireless medical gateway. BBB is built with a TI-Sitara AM335x ARM Cortex-A8 processor, has512MB DDR3 SDRAM, 4GB ROM with 8-bit embedded multi-media card, Ethernet, USB,HDMI, UART and mini USB interfaces [37]. The BBB is powered with 5V DC. This board comes with open source UNIX based operating system called Angstrom and supports other UNIX based OSs [38]. However, since this research was collaboration between an industrial partner and our research lab, real-time Unison OS had to be ported to the BBB and used as the OS in WMG implementation.

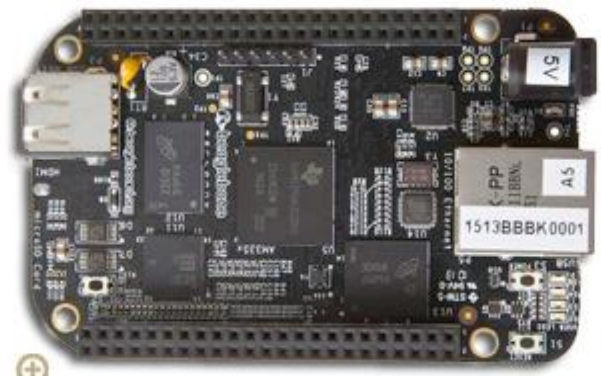

Figure 15 - BeagleBone Black

4.2.2 Stollmann Adapter

Stollmann is a Germany based company which has been developing serial and USB BT adapters and standard modules since 1999. The Stollmann adapter when connected to any PC on a USB port provides BT HDP functionality [6] [39]. By using this adapter any device can communicate with any other Continua certified medical devices like the BPM device used in this setup.

Figure 16 - Stollmann Adapter

4.3 Taking Blood Pressure Measurements

1. Step 1: Start the Code Composer Studio IDE installed on windows machine and build the working Unison user space solution [40]. Make the j-tag connection between the WMG (comprising of the BBB which has Unison loading capability) and the PC where WMG SW binary is available. Connect the serial port on WMG to the USB port of PC through a USB to serial converter.

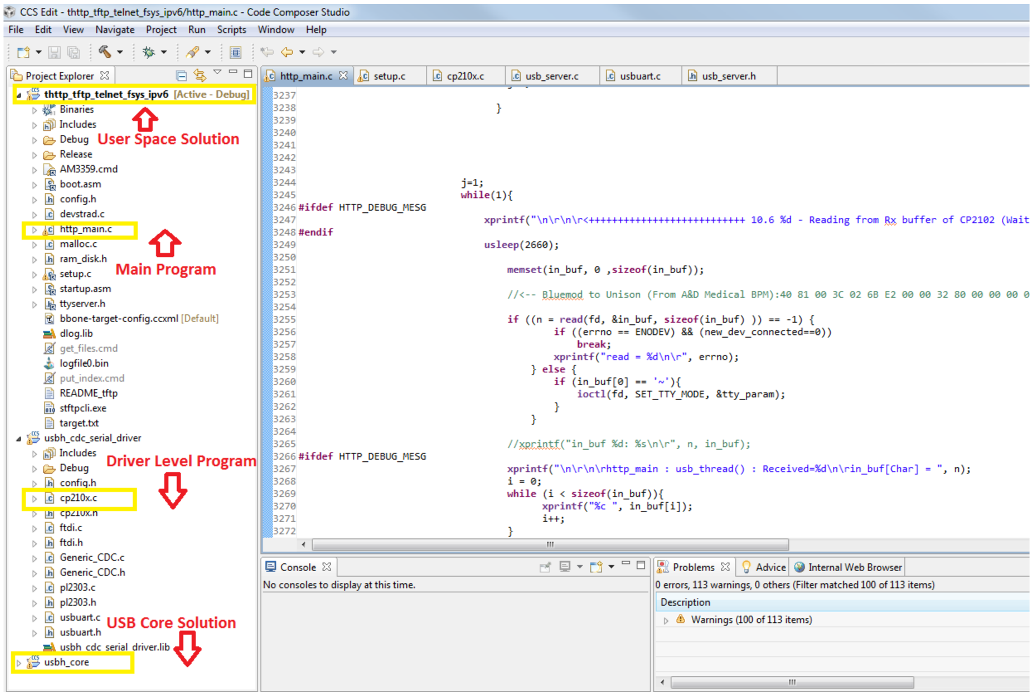

Figure 17 - Code Composer Studio Project

2. Simultaneously, open another program Tera-term, the Tera-term is an open source terminal application that can be used to send and receive data over serial port connection. On opening the Tera-term, click on setup and select the serial communication option; appropriately select the J-tag's COM port and baud rate to 115200. Now compile the built solution in CCS IDE and run as CCS debug session onto BBB.

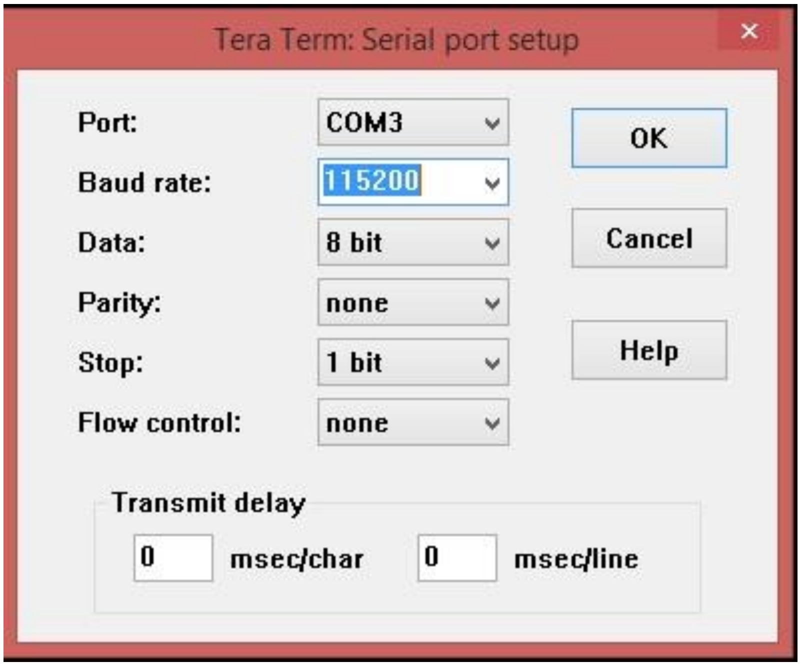

Figure 18 - Tera Term Settings

3. Step 3: The user space solution would start running and each step would be visible on the Tera-term screen. When the screen prompts as ‘telnet started’ connect the Stollmann adapter to the WMG.
4. Step 4: Power up the BPM. The designed WMG will connect with the BPM by sending the pairing and connection request. When the connection is established, the "END" will be displayed on the BPM screen [41].
5. Step 5: Now, attach the BPM cuff to the patient’s arm. Press the "start" button of the BPM to let the BPM measure blood pressure data. Wait till the BPM inflates the cuff and release the pressure to calculate the upper bound (Systolic) and lower bound (Diastolic) blood pressure and pulse of the user [41].

6. Step 6: Once the blood pressure is measured, the BPM forwards the data to the designed WMG. The WMG displays the received medical data on the Tera-term.

4.4 Comparison of Developed Gateway and Existing Continua System on Windows

The comparison between the existing continua compliant medical gateway and the newly developed medical gateway is provided in the Table 3. The comparison is provided in the three major aspects where the new development will make a substantial contribution which are cost, size of the system and average time to receive the data from the blood pressure monitor.

It can be clearly deduced from the comparison in Table 3 that newly developed medical gateway would be significantly cheaper than the windows based continua system. By keeping the average time nearly similar to that in the continua system, the developed gateway would be much less bulky and highly interoperable.

Comparison Chart		**Continua Medical Gateway**	**Developed Medical Gateway**
Cost	Operating System	$550 (Avg. cost of Windows OS + Laptop)	$145 (Avg. cost of Unison OS)
	Development Platform	Laptop includes the cost	$45 (Cost of Beaglebone Black)
	Blood Pressure Monitor	$350 (Continua & Bluetooth Compliant)	$350 (Continua & Bluetooth Compliant)
	Bluetooth Adapter	$95 (Stollmann Adapter)	$95 (Stollmann Adapter)

	Entire System Cost	$1000 approx.	$650 approx. (prototype only)
Size	Library	340 mb (with shim interface)	50 mb (non shim interface)
	Application	98 mb	not required (Code Composer Studio)
	GUI	150 mb (Continua Manager)	12 mb (Tera term)
Avg. Time	Discovery & Authentication	6 seconds	7-8 seconds
	Channel Management	3 seconds	3 seconds
	Data Display after BPM Reading	~ 1 second	~ 1 second
	Average Total Time	~ 10 seconds	~ 11 seconds

Table 3: Comparison between Continua Medical Gateway and the Developed Medical Gateway

5. CONCLUSION

5.1 Conclusion and Future Scope

In this thesis, a local transport protocol that can run on any e-health device that is running any POSIX RTOS is designed and implemented. The implemented software library is light weight and can be used on any embedded system running a POSIX RTOS. The implemented protocol is compliant to all Continua Standards and Specifications and assures interoperability with all other Continua compliant medical devices. The integration of this software library into any new device development only needs few simple steps such as adding the protocol source code into the target project and compiling it for the specific RTOS and hardware. The implemented protocol is used in the design of a wireless medical gateway and its usage to read blood pressure data from a blood pressure monitoring device is demonstrated in this thesis.

The designed protocol uses BT for transfer of data at the physical layer and uses health device profile of BT specification for handshaking, authentication and data formatting. The protocol can be easily extended to add support for other physical layers such as BT LE or Zigbee or USB. A limitation of this software library is that it cannot work with any generic BT adapter. It needs a BT HDP adapter to connect to other medical devices that are capable of BT HDP communication. Adding the ability to communicate over generic BT requires development of HDP in software and can be taken up as future work.

REFERENCES

[1] Continua Health Alliance (2015, Nov. 15). Personal Telehealth Overview [Online]. Available: http://www.continuaalliance.org.

[2] Dennis Tolstenko et al, "Continua Enabling Software Library (CESL) 'Transport Interface Layer (TIL)' ",Continua Health Alliance, rev. 59, Dec. 2008.

[3]Brian Reinhold and Barry Reinhold, "Continua Enabling Software Library (CESL) 'Programmer's Guide' ", Continua Health Alliance, rev. 0.5, Apr. 2012.

[4] M. Gondo et al, "Establishing a standard interface between multi-manycore and software tools – SHIM", in IEEE Symposium on Low-Power and High-Speed Chips - Proceedings for 2014 IEEE COOL Chips XVII, Yokohama, Japan. Apr. 2014.

[5]Beaglebone (2016, Jan 14) [Online]. Available: http://beagleboard.org/.

[6] Stollmann professionals (2015, Nov. 16). LTP / BlueAPI+ USB User Guide Release r02 [Online]. Available: www.stollmann.de.

[7] T. Oida et al., "A Security Gateway System for Medical Communication Networks –Inter-Hospital Use of the Secure Socket Layer", Proc. Joint International Conference on Autonomic & Autonomous Systems on Networking and Services, NJ, 2005, pp. 35-35.

[8] H. Yan Tung et al., "The Design of Dual ZigBee Homecare Gateway for Remote Patient Monitoring", IEEE Trans. Consum. Electron., vol. 59, pp. 756-764, Nov. 2013.

[9] N. Saranummi, "IT Applications for Pervasive, Personal and Personalized Health", IEEE Trans. Inf. Technol. Biomed., vol. 12, pp. 1-4, Jan. 2008.

[10] A. Zaheer et al., "Low-Cost Embedded Wireless Aggregation Manager for Telehealth and Telecare Solutions", in IEEE Conference on Biomedical Engineering and Sciences, Kuala Lampur, 2014, pp. 505-509.

[11] J. Yao and S. Warren, "Applying the ISO/IEEE 11073 Standards to wearable home health monitoring system", in Journal of Clinical Monitoring and Computing, vol. 19, pp. 427-436, Dec. 2005.

[12] V. Gay and P. Leijdekkers "A health monitoring system using smart phones and wearable sensors ", Int. J. of ARM, vol. 8, June 2007.

[13] J. Granados et al., "Towards Energy-Efficient HealthCare: an Internet-of-Things Architecture Using Intelligent Gateways", in International Conference on Wireless Mobile Communication and Healthcare, Athens, Nov. 2014, pp. 279-282.

[14] A. Sawand et al., "Multidisciplinary Approaches to Achieving Efficient and Trustworthy e-health Monitoring Systems ", in IEEE Symposium Privacy and Security in Commutations, Shanghai, Oct. 2014, pp. 187-192.

[15] M.J. Kirwan, "Introduction to the ITU-T H.810 Continua Design Guidelines", private communication, Continua Health Alliance, NJ, USA. Jul. 11, 2014.

[16] Bluetooth SIG, "HDP Implementation Guide White Paper", Dec 17, 2009.

[17] L. Schmitt et al., "Novel ISO/IEEE 11073 Standards for Personal Telehealth Systems Interoperability", in High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, Boston, July 2007, pp. 146-148.

[18] Randy Carroll et al, "Continua An Interoperable Personal Healthcare Ecosystem", Standards & Emerging Technologies, University of Florida, US, Oct –Dec, 2007.

[19]Sungwon Yang and Mario Gerla, "Personal Gateway in Mobile Health Monitoring ", in IEEE International Conference on Pervasive Computing and Communication, vol. 937, Seattle, 2011, pp. 636-641.

[20] "International Standard ISO/IEEE 11073-20601", IEEE Standard Reference number ISO/IEEE 11073-20601:2010(E), ISBN 978-0-7381-5826-6.

[21]11073-20601- 10407-2014 (2015, Nov. 20) Health informatics--Personal health device communication Part 20601: Application profile- Optimized Exchange Protocol. IEEE. Available: https://standards.ieee.org/findstds/standard/11073-20601-10407-2014.html

[22] LTP_Interface_Specification_r09.pdf (2016, Jan 22) [Online]. Available: www.stollmann.de

[23] Hoi Yan Tung and Kim Fung Tsang, "The Design of Dual Radio ZigBee Homecare Gateway for Remote Patient Monitoring", IEEE Transactions on Consumer Electronics, Vol. 59, No. 4, Nov, 2013.

[24] Quality RTOS & Embedded Software (2015, Nov. 18). Free RTOS. Available: http://www.freertos.org/aboutRTOS.html".

[25] Unison_Quickstart_Guide from RoweBots (2016, Jan 14) [Online]. Available: http://www.rowebots.com/

[26] RoweBots (2016, Jan 14) [Online]. Available http://www.rowebots.com/.

[27] Benefits of Bluetooth Technology, (2016, Jan 23) [Online]. Available:http://blog.bluetooth.com/10-benefits-of-bluetooth-technology

[28] Am SukhOh et al, "Personal Health Care Management System Developed under ISO/IEEE 11073 with Bluetooth HDP", Int. Journal of Smart Home, Vol. 8, 2014.

[29] Malcolm Clarke et al, "Building Point of Care Health Technologies on the IEEE 11073 Health Device Standards", IEEE Point of care Healthcare Technologies(PHT), Bangalore, India, Jan., 2013.

[30] Specification of the Bluetooth System and Bluetooth Specification Core, ver. 4.2, Dec. 2014.

[31] Bluetooth Architecture Review Board, "Multi-Channel Adaptation Protocol", Bluetooth Specification 1.0, Jun. 2008.

[32] Bluetooth Architecture Review Board, "Logical Link Control and Adaptation (L2CAP) Architecture", Bluetooth Specification 1.0, Nov. 1999.

[33] UNISON programmer's Guide, 5thed, RoweBots research Inc, Waterloo, ON, 2010.

[34] Silicon Laboratories Inc., "CP2102 USB to UART Bridge Controller", (29 Jan, 2016) [Online]. Available: www.silabs.com/

[35] R. Cravotta, "USB-to-UART Bridge Supports Legacy Upgrades", IEEE EDN, vol. 49, Texas, US. Nov. 2004.

[36] Beaglebone professionals (2015, Nov. 18). Beaglebone Black website [Online]. Available: http://beagleboard.org/black.

[37] Elinux professionals for Beaglebone black (2015, Nov. 18). BeagleBone Black [Online]. Available: http://elinux.org/Beagleboard:BeagleBoneBlack.

[38] A. Chianese et al., "Designing a Smart Multisensor Framework Based on Beaglebone Black Board", in The 9th International Conference on Embedded and Multimedia Computing, Ostrava, Aug. 2014, pp. 391-397.

[39] Product developments (2015, Nov. 21) [Online]. Available: http://www.stollmann.de/en/home.

[40] R. Heil, "MSP430 software development with code composer studio v5.3", SEI 2013 – 104, Julich, Germany. Mar. 2013

[41] Putul Saha, "Design and Implementation of Continua Compliant Wireless Medical Gateway ", MASc thesis, Dept.Elect.AndComp.Eng., Univ. Windsor, ON, 2015.

VITA AUCTORIS

NAME:	Sanam Mehta
PLACE OF BIRTH:	Bhavnagar, India
YEAR OF BIRTH:	1990
EDUCATION:	Fatima Convent High School, Bhavnagar, India, 2008
	Gujarat Technological University, India 2012
	University of Windsor, M.A.Sc, Windsor, ON, 2016