Embedded Bluetooth Stack

# **Project Plan**

**Dean Camera**

**15550597**

Supervisor John Devlin

# Abstract

This project plan and proposal outlines the major components for the development, construction and testing of an embedded Bluetooth stack, allowing for wireless communications between an embedded design and other Bluetooth enabled consumer goods. The production of a flexible stack designed for resource constrained environments will allow for specialised short range wireless devices to be produced with minimal additional cost to the system, making it an attractive alternative to existing short range wireless communication protocols (e.g. *Zigbee*) which do not have typical consumer implementations.

# Introduction

In modern society, we value connectedness highly; the advent of the internet has given us a way to remain connected to one-another while still being able to perform our daily tasks. One crucial requirement developed in recent years is the advent of consumer wireless goods; we now use WiFI connected devices to surf the web from almost any location, and Bluetooth devices to connect our personal gadgets together to make our lives easier and more pleasant.

With the advent of wireless personal devices has come the need for open source, low power and high performance Bluetooth host stacks, so that new niche products (such as wireless system monitoring and sensor networks, push-advertising, etc.) can be developed within a reasonable timeframe in a low-cost and low complexity embedded system. Typically most devices incorporating Bluetooth make use of one of three approaches:

1. Proprietary Bluetooth stacks for full operating systems (see Microsoft, Toshiba)
2. Open source Bluetooth stacks for full operating systems (see Linux, BSD)
3. Embedded single-profile Bluetooth ASIC modules (such as Bluetooth Serial Ports)

However a missing segment here is an open source, simple general purpose stack designed for use in embedded environments outside of a full operating system, on low cost hardware. By providing such a stack, the need for external Bluetooth modules or complicated processors is removed, allowing for lower system design costs.

Such a design would be required to incorporate both Bluetooth Host and Bluetooth Device roles, allowing for stack-initiated connections to be made to other devices, as well as accepting connections initiated externally. Special considerations will need to be taken into account to minimise the amount of processing time and external hardware needed, as well as abstract out the physical Bluetooth HCI data transport so that different physical modules (or, in the case of microprocessors with built in Bluetooth hardware, direct register manipulations) can be substituted. A final important consideration will be to optimise for static over dynamic memory allocation, so that memory fragmentation issues prevalent in many embedded system environments will not be a problem in resource limited environments.

The system will be split into four logical components:

1. **The physical transport layer**, which will provide the physical data transfer between the stack internals and the Bluetooth transceiver
2. **The core stack**, which will provide device connection and data channel management between the local stack and external Bluetooth devices
3. **The Bluetooth profile implementations**, providing services which will sit on top of the core stack to provide high level functionality (e.g. wireless serial port)
4. **The demonstration application**, making use of the implemented profiles to implement a practical demonstration of a functional device using the stack

Due to time constraints, only a very limited subset of the defined Bluetooth profiles will be implemented, although the core stack API should allow for future expansion.

# Project objectives

The project objectives are split into two groups; those targeted for completion by mid-year, and those targeted for completion by the end of year (and thus the end of the project).

**Mid-year Objectives:**

* Investigate and purchase an appropriate model car to adapt for the project
* Decide which are the most feasible/appropriate Bluetooth services to implement
* Design a suitable schematic and PCB for the reference hardware
* Manufacture PCB and order appropriate components
* Write suitable hardware drivers for the reference design

**End-year Objectives:**

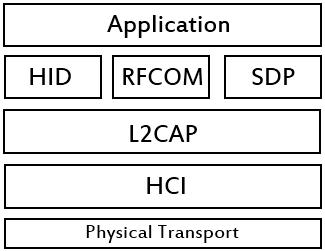
* Construct the PCB and implement it into the model car
* Implement the base Bluetooth stack core and Service Discovery Protocol
* Implement additional services on top of the Bluetooth stack core
* Program and test the completed model prototype

# Project details:

The design of the software will be covered in two sections, software and hardware. The hardware portion of the project will provide a reference design using the stack, to demonstrate its capabilities in a practical and useful manner.

## Software Stack

The crux of the system will be the embedded Bluetooth software stack design. The proposed Bluetooth stack consists of the following layers:



With the optional **HID** (Human Interface Device) and **RFCOM** (Virtual Serial) profiles being implemented at a minimum in the final design. These two services are the most useful in most applications, in addition to the mandatory **SDP** (Service Discovery Protocol) profile which is present in all Bluetooth stacks so that the available services can be enumerated by other devices.

Special considerations will need to be made in the design for minimal memory and processor requirements. As the stack will be targeted towards designs not making use of full operating systems, it will be crucial that it is designed in a manner which will allow for optimal deployment on processors lacking full memory management – ideally, dynamic memory allocation will be avoided altogether to prevent fragmentation problems in favour of exclusive use of static allocations. As the Bluetooth specification requires devices to be able to process packets of up to 64KB, for simplicity the stack design will require a 64KB block of statically allocated memory for packet fragmenting and reconstruction in addition to any other memory requirements for the stack management.

It is proposed that the design accommodate pluggable services, with each layer being properly decoupled from the ones below it, communicating through the use of proper APIs. While this approach will be less size optimal than a tightly coupled approach, it is nevertheless good practise and will allow for future expansion without invasive changes to the underlying architecture.

## Hardware Reference Design

As the software stack itself is inherently abstract, a physically constructible and testable system must also be developed in order to demonstrate the firmware and provide a real-world implementation, in order to test for correct operation. Because of this, a basic “Explorer Car” design will be implemented.

An off-the-shelf remote controlled toy car will be used, modified to remove the existing control system and replace it with a manufactured PCB containing the new system control hardware. This hardware will consist of a main system microcontroller (an Atmel AT90USB1287) connected to a low-cost USB Bluetooth module, an external 128KB SRAM module (for packet buffering), motors and associated circuitry, a microphone, speaker, illumination lamps and optional additional sensors such as accelerometers and magnetometers via an I2C compatible bus.

While serial USART connected Bluetooth HCI modules are available, these are specialised and hard to source in small quantities. As a result, a USB Bluetooth dongle will be used instead as these are widely available and very low cost. The lowest layers of the stack should be able to accept packets from either source.

To speed up development of the hardware and free up more time for software development, existing ready-made modules are planned to be used in the final design; a *Micropendous* board for the microprocessor, external SRAM and USB interface, an *Atmel Pressure-One* and *Intertial-One* boards for the additional sensors. The main PCB will be designed to accommodate these modules, as well as an appropriate power supply, microphone, speaker (with basic audio amplifier), motor control and LEDs.

# Timing to meet plan objectives

The proposed project timing Gantt Chart is attached to the end of this document. The bulk of the project time will be spent in the development of the core stack and services, however additional time will be used for the design, construction and testing of the system hardware. The time-span of the project should allow it to be completed by the end of the second semester, in time for its presentation.

# Budget

The vast majority of the hardware components have already been purchased or sourced, with only a few parts required to be bought for the final construction. A departmental budget of $200 is more than sufficient for this project, allowing for possible expansion if time allows with additional features such as a serial GPS module for positioning.

By purchasing a ready-made remote control car toy, the mechanical design of the system is already complete, and will remove the need to select appropriate motors and a battery for the system.

|  |  |
| --- | --- |
| Item | Estimated Cost |
| Micropendous-1287 Board | *Already Sourced* |
| Atmel Inertial One Board | *Already Sourced* |
| Atmel Pressure One Board | *Already Sourced* |
| PCB Manufacturing | $40 |
| Remote Controlled Car | $50 |
| Miscellaneous Discrete Components | $35 |
| Bluetooth USB Module | *Already Sourced* |

# Work to date

A previous USB-Bluetooth stack implementation has been attempted by the author, however this is incomplete and suffers for several shortcomings (e.g. the lack of dynamic service discovery and packet fragmentation). This work will be referenced during the creation of the new stack for the project, however most of this code will not be reused, as the new design should be reimplemented from scratch to remove these issues.

Central to the Bluetooth HID class profile is the HID report system, which is identical to the USB HID class – this was a deliberate design decision made during the Bluetooth standardization process for compatibility. Consequently, it is planned that the HID report parser be reused from the Author’s own embedded USB stack (<http://www.lufa-lib.org>) to speed development.

As a USB Bluetooth module will be used in the prototype design to lower costs, the author’s existing USB stack will be utilized to manage the connection between the microcontroller and module.

# Foreseeable problems (risk management)

There are several possible problems which may be encountered during the design and production process of the outlined project:

1. Testing Issues: Not all Bluetooth hosts and devices support all Bluetooth classes, resulting in the possible difficulty of sourcing suitable equipment to test the implemented services of the finished stack.
2. Compatibility issues: not all Bluetooth hosts implement all the classes in the same (or even correct) manner. This may result in a stack which is functional on some systems, but not others.
3. Licensing issues: While the Bluetooth specification is freely available from the official Bluetooth website, it is not currently known what restrictions are placed upon software implementations.
4. Physical Injury: As with all physical construction projects, there is a small risk of personal injury whilst working in the laboratory. Proper safety equipment should be worn and safety guidelines followed to mitigate this risk.

# Conclusion

The proposed project meets the requirements of complexity, marketability, usefulness and other related criteria for a thesis project, and should provide valuable insight into the process of embedded software and hardware design, as well as other ancillary skills such as marketing and time management. This project will fill a market niche once completed to a satisfactory standard, allowing it to be commercialised if desired, or released to the community for further development if not.