Bluetooth Block

ECE441 Wearable Sensor for the Blind Project

Samuel Jia Khai Lee

ECE Senior Design

Oregon State University,

Corvallis, OR USA

leejiaks@oregonstate.edu

Keywords—Bluetooth; wireless; block description; design verification

# Introduction

The purpose of this document is to describe the Bluetooth block of the ECE441 Wearable Sensor for the Blind project to 4th year ECE students with the intention of these students being able to build and verify the block without further research. The Bluetooth block is implemented using a SH\_BT\_Board V1.3 HC-05 Bluetooth module. This document provides an overview of the overall block function including interface properties and a schematic, verification for the design in the form of a step-by-step testing process, and support for the validity of the design in the form of outside research and numerical justification addressing individual properties.

# Design Details

The wiring diagram (Fig.1) presents the block design, including the interfaces of the block. These interfaces are further elaborated through validation information given in Table 1, which demonstrates externally-supported validly for each of the block’s properties.

cntrllr\_cd\_bltth\_data

STATE

RXD

TXD

GND

VCC

EN

Bluetooth

bltth\_cntrllr\_cd\_rf

Pwr\_spply\_bltth\_dcpwr

1. Black Box Diagram of Bluetooth Block

# Block Overview

The Bluetooth transmit serial data from the system controller to the system feedback module. The block is powered from **pwr\_spply\_bltth\_dcpwr** interface usually around 5V with a nominal current draw of 50 mA. Fig. 1 shows the black box diagram. Data are sent via a serial port protocol over the **cntrllr\_cd\_bltth\_data** interface. Finally, the output is represented by the **bltth\_cntrllr\_cd\_rf** interface. The function of the Bluetooth block is to transmit data serially between system sensor module and system feedback module. This block is completed by Samuel Lee.   
  
A full listing of interface properties can be found in Table 1.

bltth\_cntrllr\_cd\_rf

Pwr\_spply\_bltth\_dcpwr

Bluetooth

cntrllr\_cd\_bltth\_data

1. Black Box Diagram of Bluetooth Block

`

| Interface | Properties |
| --- | --- |
| **pwr\_spply\_bltth\_dcpwr** | 1.Min Voltage: 3.6V  2.Max Voltage: 6V  3. Nominal Current: 30mA ±10% (During pairing)  4. Nominal Current: 8mA ±10% (After pairing) |
| **cntrllr\_cd\_bltth\_data** | 1.Language: Arduino  2.Transmission Rate: 38400 Baud  3.Program Size: 2050 bytes (Master code) / 2256 bytes (Slave code)  4.Extra bit: 1 Stop bit, 0 Parity bit |
| **bltth\_cntrllr\_cd\_rf** | 1.Protocol: Serial Port Protocol  2.Transmission rate: 38400 Baud  3.Range: At least 1 meter  4.Other: name – ece34\_M (Master mode) / ece34\_S (Slave mode)  5.Other: password – group34  6.Extra bit: 1 Stop bit, 0 Parity bit |

1. Bluetooth Block Interfaces and Properties

# Block Verification

Based on the interfaces for this block, a verification (testing) process needs to be indicated. This will allow the final constructed design to be tested verifying that all the interface properties have been met and that the block is ready for integration into the system.

## Power Testing (**pwr\_spply\_bltth\_dcpwr**)

This test will verify that the Bluetooth module operates after providing **pwr\_spply\_bltth\_dcpwr** interface.

1. Connect the Bluetooth block to power via the **pwr\_spply\_bltth\_dcpwr** interface at system microcontroller’s VCC pin OR use a power supply at 5V to the VCC on the Bluetooth.
2. Connect a multimeter to the **pwr\_spply\_bltth\_dcpwr** to check the voltage.
3. Connect a light bulb at pin 7 of the system’s microcontroller, TX pin on the system microcontroller to RX pin on Bluetooth; RX pin on the microcontroller to TX pin on Bluetooth; VCC and GND pins on microcontroller to VCC and GND pins on the Bluetooth respectively.
4. Detach VCC pin on the Bluetooth, then load a test program to the system microcontroller using “arduino.exe” via the USB Asp programmer. It does the following: when received data ‘1’, write to pin 7 to light up the light bulb for 2 seconds. Reattach VCC pin.
5. Connect a mobile device to the Bluetooth module and use a “bluetooth controller app”, *ArduDriod*, to send data to the system.

PASS: This test passes if 9 out of 10 individuals are able to successfully power up the Bluetooth by illuminating the LED on the Bluetooth and light bulb by using a mobile device.

[Link to Video](https://drive.google.com/open?id=1zA6jWgSdm-Rj7T7iHdYL0c3s9vWCM5t_)

## Bluetooth Configuration Testing (**cntrllr\_cd\_bltth\_data**)

This test will verify that the configuration of the Bluetooth module by using AT mode in Table 1. The communication interface **cntrllr\_cd\_bltth\_data** will be verified to the interface properties as expected.

1. Connect the Bluetooth block to power via the **pwr\_spply\_bltth\_dcpwr** interface at system microcontroller’s VCC pin OR use a power supply at 5V to the VCC on the Bluetooth.
2. Connect the TX pin on the system microcontroller to the TX pin on the Bluetooth; RX pin on the microcontroller to the RX pin on the Bluetooth; VCC pin on the microcontroller to VCC and EN pins on the Bluetooth; GND pin on the microcontroller to the GND pin on the Bluetooth.
3. Detach VCC pin on the Bluetooth, then load an empty program to the mimcrocontroller using “arduino.exe” via the USB Asp programmer. Reattach the VCC pin.
4. Open the Serial Monitor on the Arduino IDE and use AT Commands: AT, AT+UART?, AT+NAME?, AT+PSWD?, AT+ROLE?.

PASS: This test passes if all AT command returns “OK” and values of the parameters.

[Link to Video](https://drive.google.com/open?id=1zA6jWgSdm-Rj7T7iHdYL0c3s9vWCM5t_)

## Bluetooth to Bluetooth Testing(**bltth\_cntrllr\_cd\_rf**)

This test will verify that Master and Slave Bluetooth module will communicate to each other by sending data using **bltth\_cntrllr\_cd\_rf**.

1. Based on *Bluetoth Configuration Testing*, configure the first Bluetooth module to be in Slave mode and the second Bluetooth module to be in Master mode using **cntrllr\_cd\_bltth\_data** interface.
2. Configure Bluetooth module to Slave mode with the following setting and command:
   1. NAME = ece34\_S ; Command: AT+NAME=ece34\_S
   2. PASSWORD = group34 ; Command: AT+PSWD=group34
   3. UART = 38400 Baud Rate, 1 Stop bit, 0 Parity bit ; Command: AT+UART=38400,0,0
   4. ROLE = 0 (Slave) ; Command: AT+ROLE=0
   5. Returns ADDRESS (14:3:620fe) ; Command: AT+ADDR?
3. Configure Bluetooth module to Master mode with the following setting and command:
   1. NAME = ece34\_M ; Command: AT+NAME=ece34\_M
   2. PASSWORD = group34 ; Command: AT+PSWD=group34
   3. UART = 38400 Baud Rate, 1 Stop bit, 0 Parity bit ; Command: AT+UART=38400,0,0
   4. ROLE = 1 (Master) ; Command: AT+ROLE=1
   5. CMOD = 1 ; Command: AT+CMODE=1
   6. BIND = 14:3:620fe ; Command: AT+BIND=14,3,620fe
4. Detach VCC pin on the Slave Bluetooth module, then load a test Slave program to the system microcontroller using “arduino.exe” via the USB Asp programmer. It does the following: reads the serial input via **bltth\_cntrllr\_cd\_rf**, waits for data ‘1’ : Turns light bulb on for 100ms. Reattach VCC pin.
5. Detach VCC pin on the Master Bluetooth module, then load a test Master program to the system microcontroller using “arduino.exe” via the USB Asp programmer. It does the following: for every 2 second, ‘1’ is sent serially via **bltth\_cntrllr\_cd\_rf** and turns off local light bulb, send ‘0’ and turns on local light bulb. Reattach VCC pin.

PASS: This test passes if the light bulb in the Slave Bluetooth module lights up when the light bulb in the Master Bluetooth module turns off.

[Link to Video](https://drive.google.com/open?id=1zA6jWgSdm-Rj7T7iHdYL0c3s9vWCM5t_)

If the block passes all the listed tests, all interface properties have been verified and the block is ready for inclusion into the system.

# Design Validation

For this block, an off the shelf solution was chosen. The HC-05 Bluetooth module fits the needs of the block interfaces without needing to be modified except minor configurations. Table 2 includes the interface property validation for this block. All the interface properties have been addressed and the design meets or exceeds the properties.

| Interface | Properties |
| --- | --- |
| **pwr\_spply\_bltth\_dcpwr** |  |
| Min Voltage: 3.6V | SH\_BT\_Board V1.3 HC-05 Bluetooth has an input voltage range of 3.6 – 6V via **pwr\_spply\_bltth\_dcpwr** interface. BC417 Chip runs on 3.3V but a voltage regulator is used. (Section 1 SH\_BT\_Board Datasheet, Page 2[1]) |
| Max Voltage: 6V | SH\_BT\_Board V1.3 HC-05 Bluetooth has an input voltage range of 3.6 – 6V via **pwr\_spply\_bltth\_dcpwr** interface. BC417 Chip runs on 3.3V but a voltage regulator is used. (Section 1 SH\_BT\_Board Datasheet, Page 2 [1]) |
| Nominal Current: 30mA ±10% (During pairing) | Operating current during pairing is 30mA. (Section 1 SH\_BT\_Board Datasheet, Page 3 [1]) |
| Nominal Current: 8mA ±10% (After pairing) | Operating current after pairing is 8mA during communication. (Section 1 SH\_BT\_Board Datasheet, Page 3 [1]) |
| **cntrllr\_cd\_bltth\_data** |  |
| Language: Arduino | Controller code is programmed by using Arduino IDE to interact with the Bluetooth module. (Arduino [3]) |
| Transmission Rate: 38400 Baud | Serial baud rate is used by the system microcontroller to transfer data. |
| Program Size: 2050 bytes (Master code) / 2256 bytes (Slave code) | ATmega328P has 32K Bytes of In-System Self-Programmable Flash program memory which is sufficient for system code. (ATmega328 Datasheet Page 1 [4]) |
| Extra bit: 1 Stop bit, 0 Parity bit | The extra stop bit is useful to allow the Bluetooth module to know that the connection is still valid. |
| **bltth\_cntrllr\_cd\_rf** |  |
| Protocol: Serial Port Protocol | The HC-05 Bluetooth module uses Bluetooth SSP (Serial Port Protocol) for transparent wireless serial connection setup. (Overview of Serial Port Bluetooth Module HC-05 [6]) |
| Transmission rate: 38400 Baud | Supported baud rate: 4800, 9600,19200, 38400, 57600, 1152000, 230400, 460800, 921600, 1382400. Baud rate is the capability of the serial port to transfer a maximum of 38400 bits per second. (Section 3.1 AT Command Mode, SH\_BT\_Board Datasheet, Page 12 [1]) |
| Range: At least 1 meter | The Bluetooth module has a distance range of approximately 10 meters, it is sufficient for a head-to-waist distance of approximately 1 meter. (General Specification, HC-05 Bluetooth Module Breakout board Page 1 [5]) |
| Other: name – ece34\_M (Master mode) / ece34\_S (Slave mode) | The HC-05 Bluetooth module is Master and Slave mode compatible. Master role have paired memory to remember an address of a Slave Bluetooth module to pair with it automatically. (HC Serial Bluetooth Instruction Manual, Page 3 [2]) |
| Other: password – group34 | Password is set for security reasons. |
| Extra bit: 1 Stop bit, 0 Parity bit | The extra stop bit is useful to allow the Bluetooth module to know that the connection is still valid. |

1. Interface Property Validation for the Bluetooth Block

# Bills of Materials

Table 3 lists the bills of materials used for the Bluetooth block.

| Item | Price | Link |
| --- | --- | --- |
| SH\_BT\_Board HC-05 Bluetooth Module | $7.99 a pair | [Link](https://www.amazon.com/gp/product/B074GMQ6G3/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1) |

1. Bills of Materials for the Bluetooth Block

[1] SH\_BT\_Board Datahseet, <<http://www.ram-e-shop.com/ds/general/Bluetooth_TRx_Module_New.pdf>>

[2] HC Serial Bluetooth Instruction Manual, <<https://cdn.makezine.com/uploads/2014/03/hc_hc-05-user-instructions-bluetooth.pdf>>

[3] Getting Started with the Arduino Pro Mini, <<https://www.arduino.cc/en/Guide/ArduinoProMini#toc2>>

[4] Atmega328 Datasheet, <<http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf>>

[5] HC-05 Bluetooth Module Breakout board Page1 <<http://e-gizmo.com/oc/kits%20documents/HC-05%20Bluetooth%20Module%20breakoutboard/HC05%20Bluetooth%20module%20breakout%20board.pdf>>

[6] Serial Port Bluetooth Module HC-05 <<https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Master/Slave)_:_HC-05>>