ECE 445

Fall 2012

Design Review

Bluetooth Stereo Network

Team 7

Jeff Wheeler

Rishi Ratan

Jerry Sun

Prof. Andrew Carl Singer

TA: Justine Fortier

Table of Contents

1.0 Introduction

1.1 Project Overview

The goal of this project is to design and develop a Bluetooth-powered stereo network that accepts audio data from User-Equipment (UE) and seamlessly transfers this data wirelessly to speakers closest to the UE location. Numerous Bluetooth-powered speakers currently exist in the consumer market, however none of them provide location specific audio playback. The product we plan to develop over the course of this semester is unavailable in the wireless audio industry and would greatly streamline/improve the wireless audio playback experience for consumers.

1.2 Objectives

1.2.1 Goals

We intend to develop a stereo network in which a user can play music from their phone or other device over Bluetooth (BT) to a centralized hub, which will then forward the audio data to the speaker network, choosing the appropriate speaker to play depending on its proximity to the UE. This network will be composed of speaker adapters like those currently available in the market, but modified to observe how strongly they "see" the user equipment (e.g. by measuring attenuation on the BT signal from the streaming device) and relaying this information to the hub.

We seek a system in which the UE only need pair with the central hub and does not need to pair and subsequently un-pair with each speaker as it moves between rooms. The UE will pair with this hub (and *not* the individual speakers), which will then forward the audio data to the appropriate "speaker adapter" based on where it thinks the user is.

1.2.2 Benefits

- User-friendly with superior form-factor.
- No additional software/hardware needed for phone or other UEs.
- Seamless pairing with speakers without a need since the hub centrally connects all the speakers.

1.2.3 Features

- Compatible with any Bluetooth enabled audio device.
- Compatible with all existing speakers on the market that support 3.5mm audio jack.
- CD quality uncompressed audio.
- Audio range of up to 30 feet from hub to speaker and from speaker to UE.
- Dedicated audio streaming via TI PurePath wireless.
- Support for streaming on up to 4 devices.
- Microcontroller-Unit (MCU) for hub and speaker management.

2.0 Design Overview

2.1 Block Diagrams

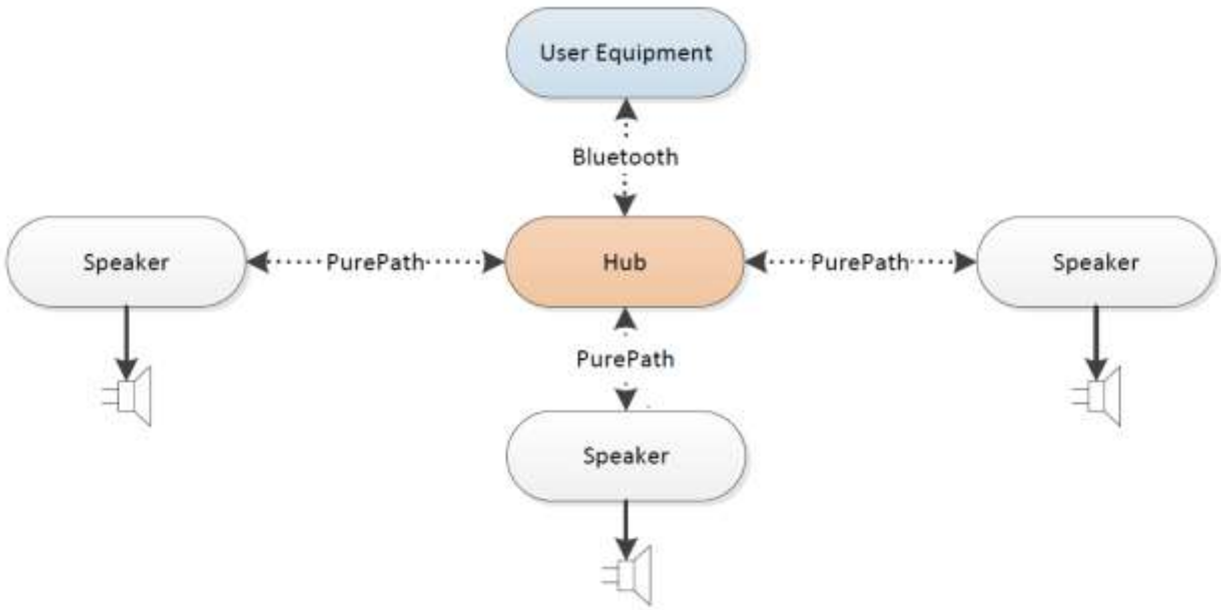


Fig 1: Overall System Setup Block Diagram

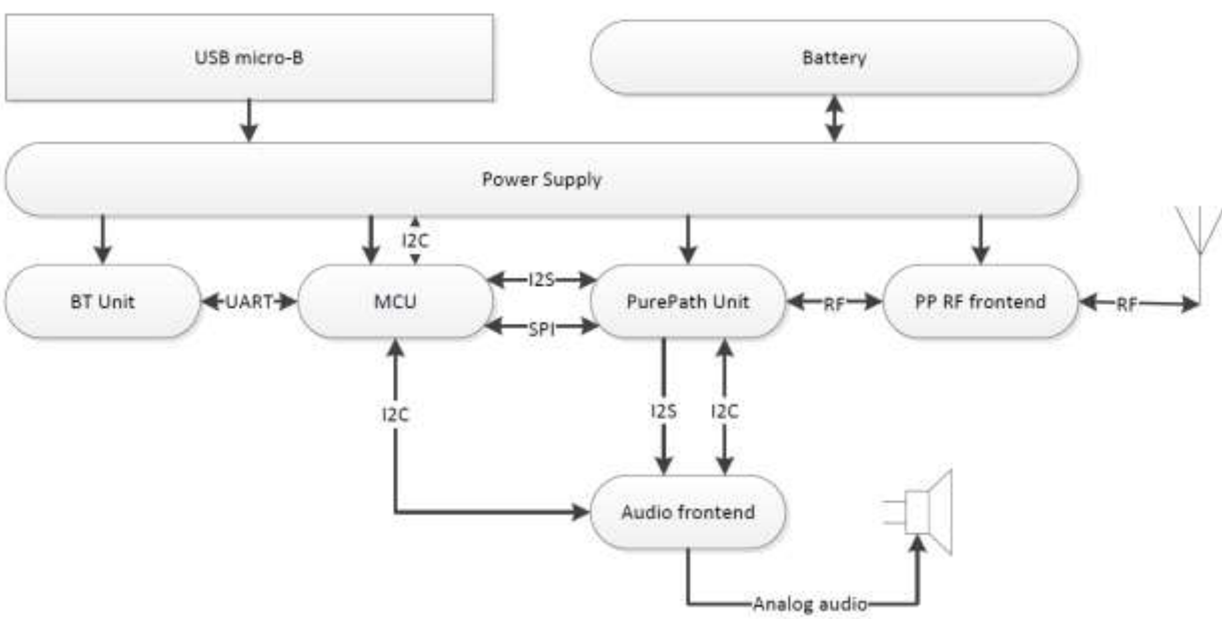


Fig 2: Speaker Adapter Setup Block Diagram

1

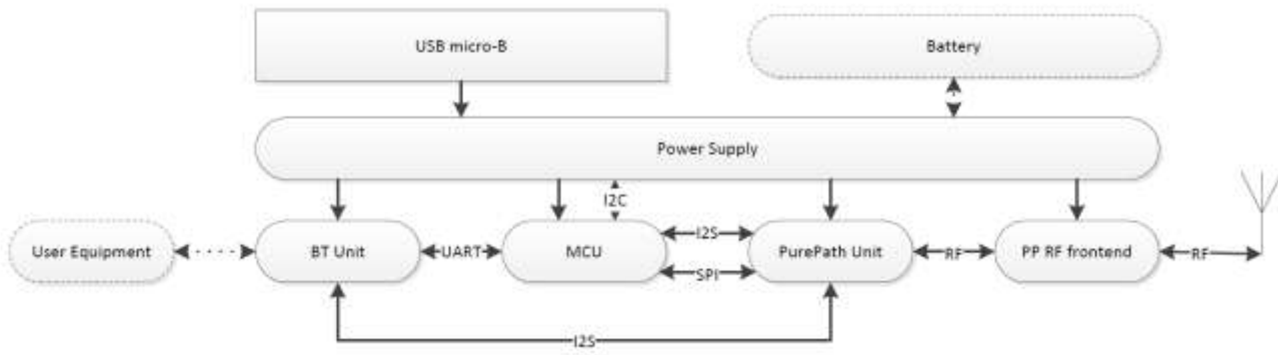


Fig 3: Hub Setup Block Diagram

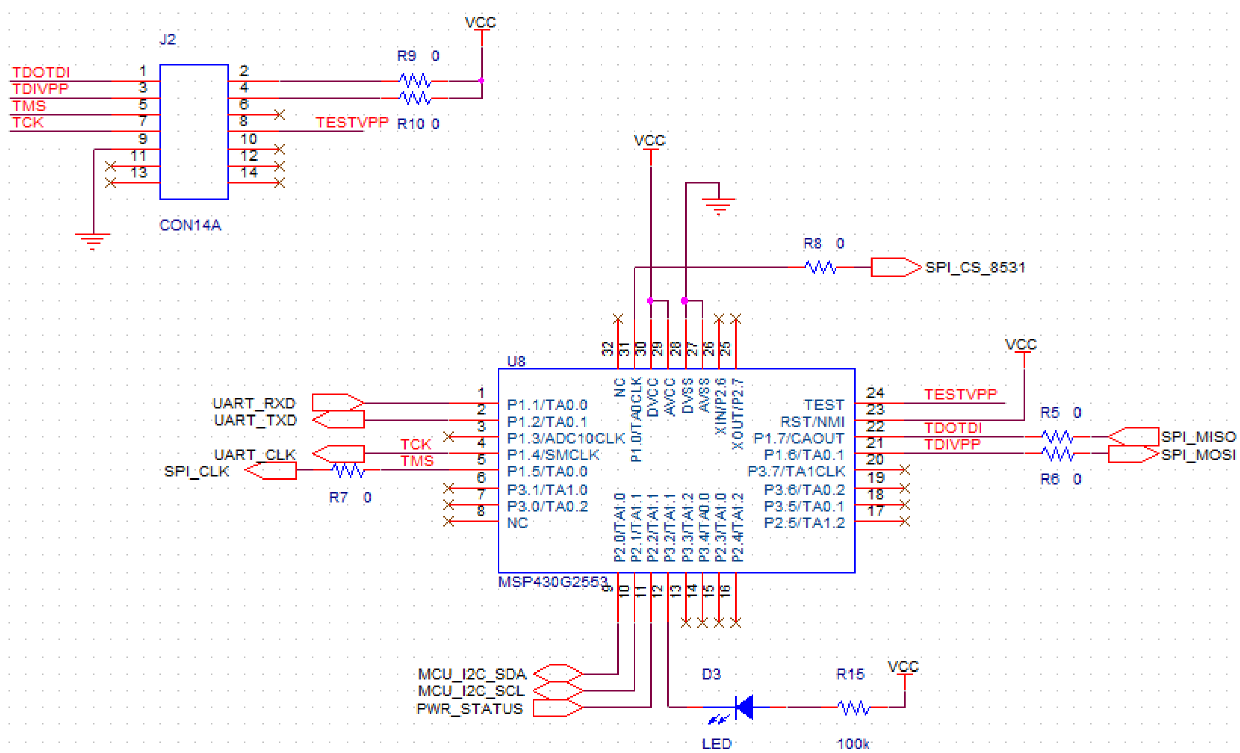


Fig 4: Microcontroller Unit

UART – Universal Asynchronous Receiver/Transmitter
MCU – Microcontroller Unit
I2S – Inter-IC Sound
SPI – Serial Peripheral Interface Bus
I2C – Inter-Integrated Circuit

2.2 Block Descriptions

2.2.1 Speaker Adapter

2.2.1.1 Microcontroller Unit

The MCU manages the operation of the speaker. To do this, it coordinates the Bluetooth Unit and the PurePath Unit, in addition to performing some initialization of ICs in the audio frontend (e.g. to specify the DSP settings via I2C)[1].

Most critically after initializing devices at boot, the MCU is responsible for encoding the signal strength data (as seen to the UE) available from the Bluetooth IC, received via UART, as datagrams to write to the host-controller SPI interface on the PurePath IC.

Furthermore, the MCU needs to initialize the PurePath Unit to pair with the master for the datagram return stream—i.e. the PurePath IC must connect to the master's PurePath IC in order to send all the datagrams with signal strength data provided by the Bluetooth Unit.

The logical flow of the MCU program is shown in Figure 5.

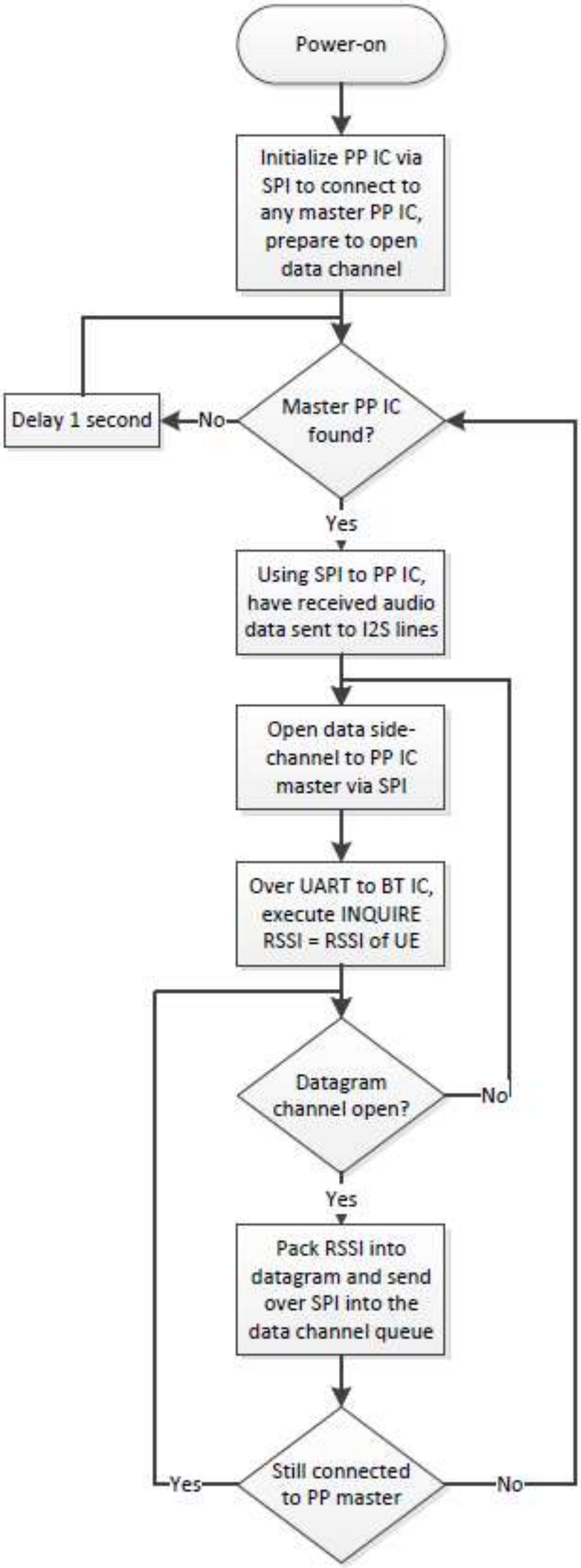


Fig 5: Speaker Adapter MCU Flow

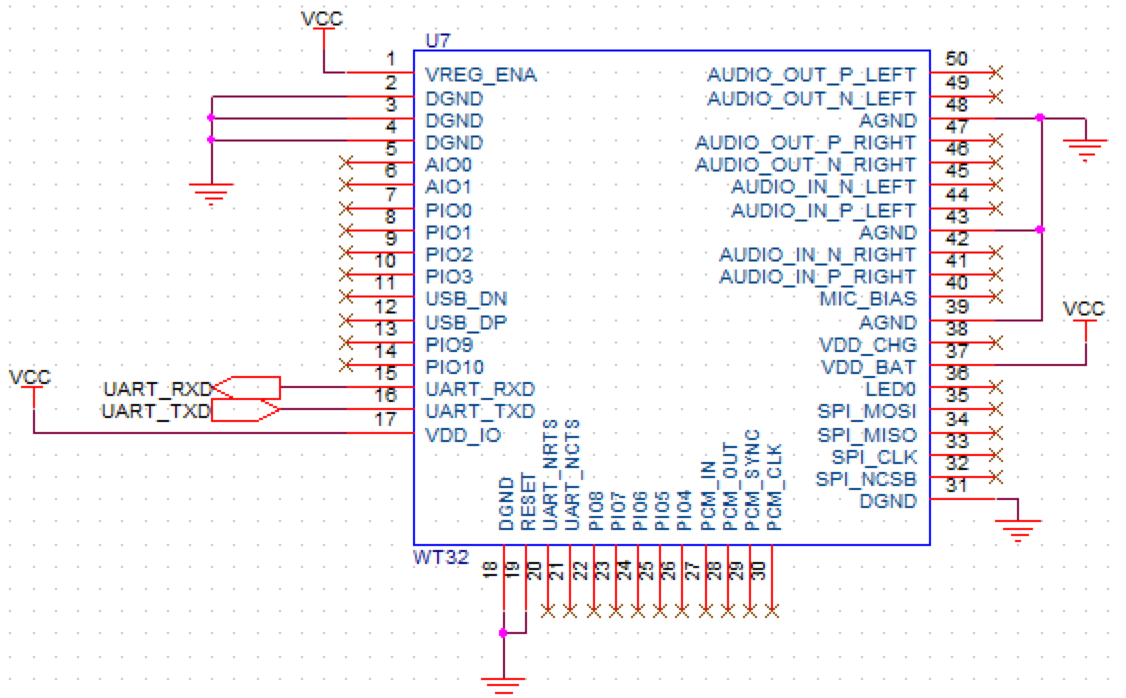


Fig 6: Bluetooth Unit

2.2.1.2 Bluetooth Unit

The Bluetooth Unit consists of the BT IC, an antenna, and a crystal oscillator providing the “slow clock” for the BT IC, all provided on an integrated PCB. The speaker BT IC never pairs with anything: it is used exclusively to discover the signal strength of the UE, and never transmits any data. The microcontroller requests the signal strength data via UART.

We are using the Bluegiga WT32, which has high-quality firmware, iWRAP4, to configure Bluetooth parameters and access functionality typically hidden behind a much more complicated UART interface. The module provides the Received Signal Strength Indication, or RSSI, command to acquire the signal strength. The RSSI ranges from -128 (Poor) to 20 (Good). Using the RSSI allows us to pick the speaker closest to the UE and play from the correct speakers.

The Bluetooth unit also provides the Inquiry function, which is used to find other Bluetooth devices in the area [2]. Combined with RSSI, the speaker units will send and inquiry and return the RSSI of the UE to the hub. The hub will compare the RSSI values sent by the speaker units and then use this information to determine which speaker plays the music.

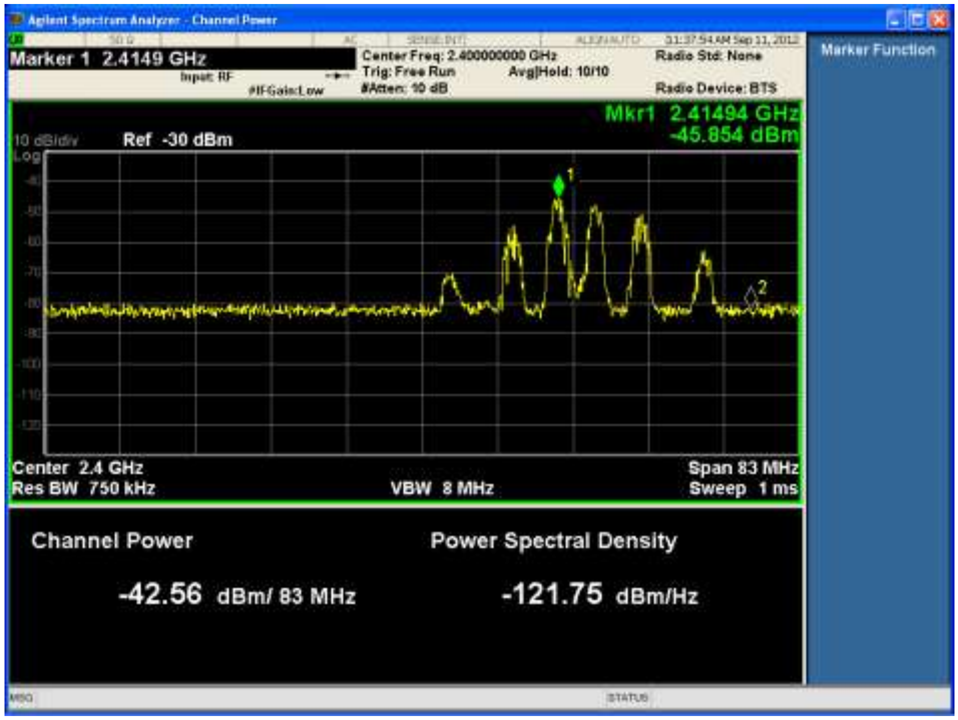


Fig 7: Bluetooth Spectrum

2.2.1.3 PurePath Unit

The PurePath (PP) Unit consists of the PurePath IC and a crystal oscillator. The PP IC in the hub is responsible for encoding and decoding the PP radio signals. The MCU will use I2C, speaking to the PP IC, to control which speaker adapter the device sends to. Finally, the BT IC will send audio data for broadcast to speaker adapters to the PurePath IC in the form of I2S.

The PP IC also receives the signal strength to the UE from each speaker adapter, in the form of a returning audio stream. This data is provided to the MCU as an I2S audio stream and will aid in determining the appropriate speaker to play audio data to.

In order to program IC we directly use a proprietary PP IC programmer and its associated software developed by Texas Instruments (TI), but certain broadcast settings are specified at runtime by the MCU, to control which speaker adapters play the audio.

The main advantage of using this chip is that is designed to interface seamlessly with the TI MSP430 MCU, TI CC2590 RF Front-End IC as well as the Audio Front-End chips. No additional development is needed to configure the chip with the external audio front-end as well as the RF front-end.

Additionally, the PPIC provides an average of 2Mbps over-the-air data rate, up to 11dBm output power when used with the CC2590 range extender circuit and -87dBm sensitivity as stated in the datasheet.

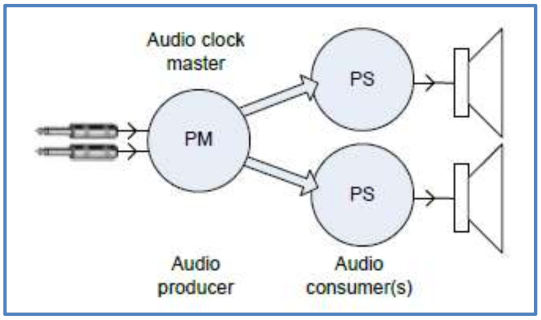


Fig 8:
Protocol Master
Slave Network Illustration

The TI CC8531 PurePath chip is unique as compared to traditional Bluetooth chips in that on top of operating within the 2.4 GHz ISM band the key feature supported by the PurePath technology is that it embodies a star-network topology. The star topology consists of one protocol master device along with up to four protocol slave devices. So in our Hub design the PurePath IC will be the protocol master device for the corresponding PurePath ICs in each of the respective speakers of the network. The protocol master PPIC of the hub will communicate with each of the children speaker nodes in our network over fixed time intervals of 2.5ms during which time the respective nodes can also communicate back with the master node over the same RF channels. Furthermore, for enhanced wireless audio streaming the PurePath chip employs adaptive frequency hopping over 18 RF channels that each has a 4MHz bandwidth. Furthermore, the protocol master measure transmitted energy on each RF channel in a listen-before-talk operation and then decides whether to transmit on the RF channel or not. Great advantage on the designer end is that we this chip has seamless integration with the CC2590 RF Front-End chip that acts as a range extended to further amplify the signal strength of our signals [3].

As seen in Fig 8, the Protocol master-slave network configuration of the PurePath IC is shown and it defines various features useful in our design. Some key parameters of interest in our system are the following:

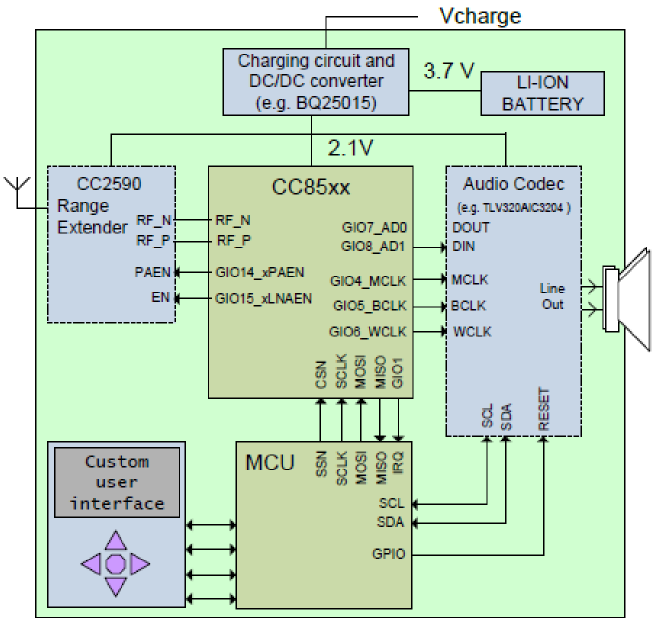


Fig 9: Overall Block diagram of system as referenced in the application note.

- Universal audio sample rate to be used by the entire audio front end is defined by the master node PPIC residing in the hub.
- Each slave PPIC nodes in the respective speakers send a certain manufacturing ID for authentication when trying to connect with the protocol master node.
- Power control features available in the design are phenomenal and reduce overall power consumption in our system. If a protocol slave's local audio input/outputs are silent the protocol master node automatically reduces audio device power consumption, thereby enhancing system battery life.

In Fig 9, the overall block level diagram for a typical wireless speaker network is displayed as seen in the application note for the CC 8531. We use this configuration as our base design for the Hub as well as the speakers. Though the CC8531 PurePath IC is capable of supporting a custom user interface we decided to not include this feature in our product as we don't anticipate the actual user actively interacting with the MCU unit in the hub. We intend the Hub as well as the speaker to be sort of like a black-box to the end user in that they merely need to configure their phone to the network only, with rest of the tasks performed in the back-end.

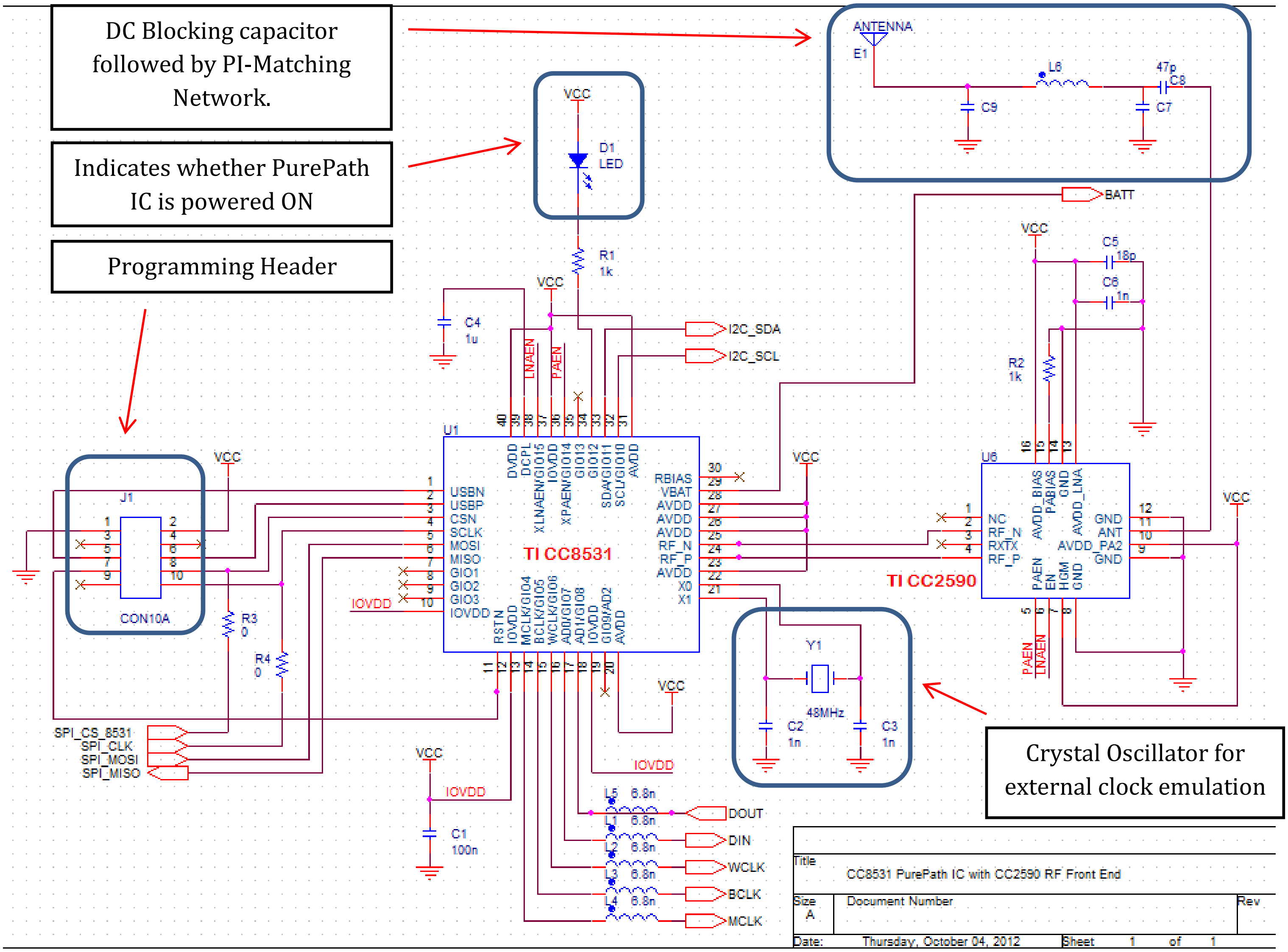


Fig 10: Chip-Level Schematics for the TI CC8531 PurePath IC and the TI CC2590 RF Front-End

2.2.1.4 PurePath RF Front-End

The PurePath RF Frontend includes an RF frontend amplifier IC and antenna. We hope to use the TI CC2590 chip as an integral part of our RF Front-End. The chip is optimal for low-power and low voltage wireless applications at 2.4GHz and has a seamless integration with the PP IC.

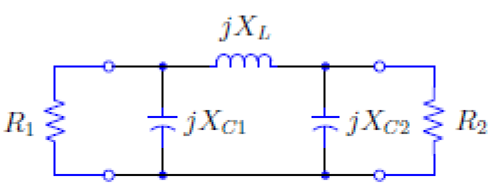
A key feature about incorporating this chip into our design is that it increases the link-budget as it contains an in-built power amplifier (PA) to enhance the output power and a Low-Noise Amplifier (LNA) with high gain and low-noise to enhance our receiver sensitivity.

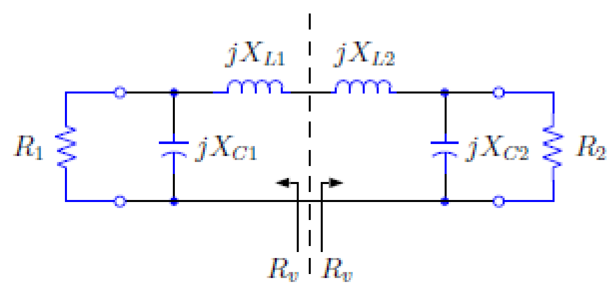
The output power of the CC2590 chip is controlled by the input power and the Power Amplifier embedded inside the chip is designed to work in Class AB mode, i.e. in cases of compression with the best efficiency being achieved whenever a strong input signal is received [4].

The values for discrete components such as the resistors, capacitors and inductors are based on the information stated in the reference designs of the respective TI application notes for the PurePath Wireless Audio Headset design and Family Guide [5][6].

The purpose of the DC blocking capacitor C8, as seen in Fig 10, is as the name suggests limiting the DC signal from leaking from the RF Front-End chip onto the antenna. We are assuming our entire RF section of the system works on a 50Ω reference thus matching considerations is critical to our design. Each respective block in the overall RF sub-system must be conjugately matched with the subsequent blocks in order to prevent losses within the circuit due to reflections caused due to an improper conjugate impedance match. The TI CC8531 PurePath chip has a reference output load impedance given by $Z_{OUT} = 70 + j30$ as per the data-sheet provided by the manufacturer and has to be thus conjugately matched with the effective input impedance as seen into the CC2590 RF Front-End chip and vice-versa. We anticipated this design requirement while brainstorming for parts, and therefore chose the CC2590 as the RF Front-End chip as it contains an in built matching network circuit that will conjugately match to the appropriate output impedance seen into the CC8531 IC. Though, we do not necessarily need to worry about designing the appropriate matching network circuit for the respective ICs amongst the RF section, we however need to design the matching network for appropriately matching the output impedance of the RF Front-End chip with the respective input impedance as seen by looking into the antenna port. We accomplish this task by designing a PI-Matching Network using the following approach:

The Pi-Network is basically like two-cascaded L-networks that collectively try to match both R1, R2 to a "virtual resistance" R_v. We use this topology as we anticipate the impedances to be matched to be very high



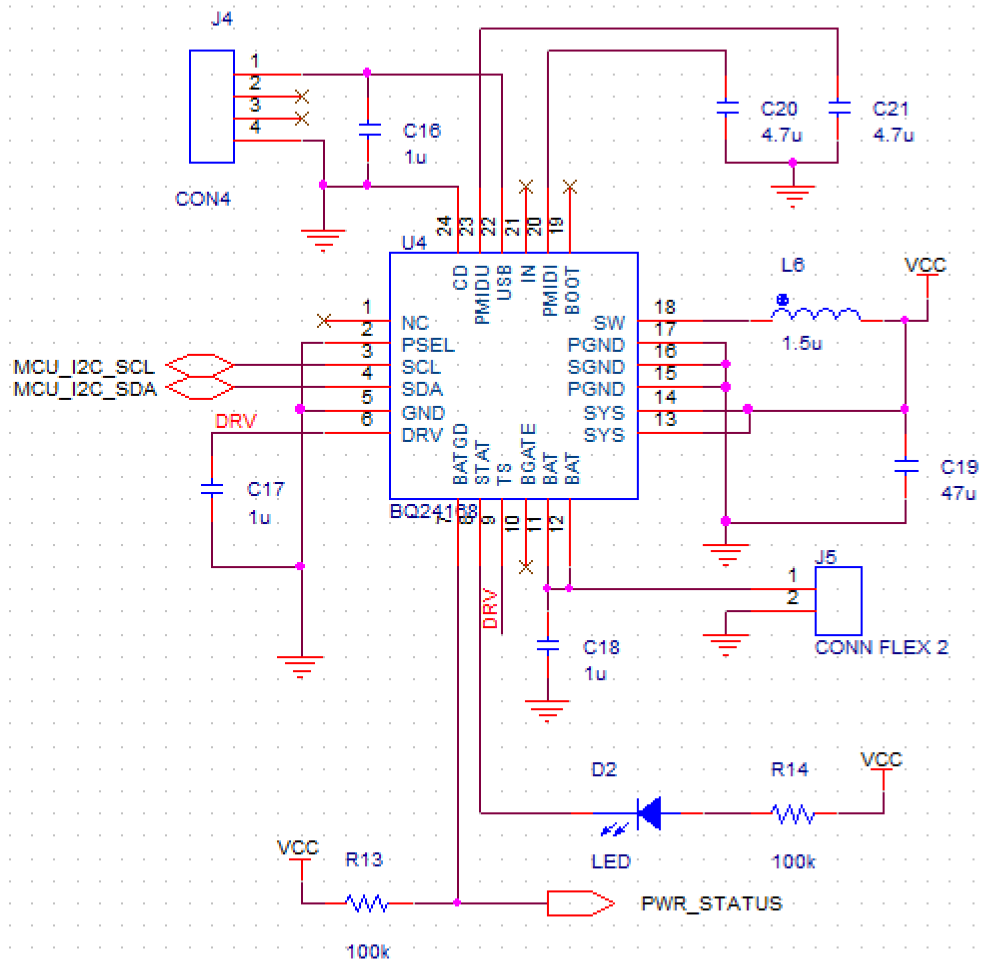


$$Q_1 = \sqrt{\frac{R_1}{R_v} - 1} \; and \; Q_2 = \sqrt{\frac{R_2}{R_v} - 1}$$

In all likelihood we will be matching to a situation wherein R2 > R1 so the overall Q (quality factor of our circuit) is given by Q_2. Whenever the impedances are complex we can use the absorption method to basically incorporate the reactive element of the respective load into jX_{C1} and jX_{C2} respectively after which the problem just becomes a regular resistive PI-Matching network design. After we calculate the effective values for X_L, X_{C1}, X_{C2} we can back solve for the actual MN element values.

Finally:

$$X_{C2,eff} = -\frac{R_2}{Q_2}, X_{Leff} = \frac{\left(R_2 Q_2 + R_2\sqrt{\frac{R_1}{R_2}(Q_2^2+1)}\right)}{Q_2^2+1}, and \; X_{C1,eff} = -\sqrt{\frac{R_1 R_2}{(Q_2^2+1) - \frac{R_2}{R_1}}}$$



<u>Fig 11:</u> Power Supply Circuit

2.2.1.5 Power Supply

The power supply will provide a regulated voltage Vcc to all of our ICs, which all operate at least in the region 3.3±0.3V. We selected a battery charging IC to control the regulation: it functions both as our voltage regulator and as our charging circuit from micro-USB (we preferred using one multi-function IC for the sake of increased circuit integration). We've selected TI's BQ24168 IC as it serves these functions, while also using very little power.

Critically, the BQ24168 IC can regulate the 5V DC rail from USB for the speaker's Vcc even when no battery is connected (a feature missing on many battery-charging and regulation ICs) [7].

In our power supply circuit, we've connected the Vbus, the USB 5V DC line coming in from the connector, to the USB pin on the IC, with a simple bypass capacitor C16 that functions to AC-ground the Vbus line.

The batteries we've elected to use are two-terminal Lithium-Ion batteries (i.e. there is no terminal on which to sense the internal temperature). Because of this, we've chosen not to use the temperature sense protection behavior on the IC, and therefore have tied that terminal to ground (via a bypass capacitor C17, per the application notes).

The voltage regulator is cleaned up significantly through the addition of the inductor L6, which completes the converter circuit. The application notes explain the choice of 1.5uH, but we then sought to verify that we could satisfy the current and temperature ratings of an inductor before placing it into our power supply circuit. The provided formulae are

IC	Avg Current (mA)
CC8531	25
CC2590	22
MSP430G2553	0.42
TLV320AIC3204	50.
TPA6130A2	2.5
WT32	4.5[2]
Total	104.

Table 1: Typical current consumption

$$I_{peak} = I_{LOAD} \times (1 + \frac{\%_{ripple}}{2})$$

We estimated the $\%_{ripple} \approx 0$, and calculated I_{LOAD} by summing the typical operating current for every IC (values shown in Table 1):

$$I_{peak} = I_{LOAD} = \sum_j IC_j \approx 104.\,mA$$

where IC_j is the typical current drawn by each IC, and pick an inductor with a current rating above that peak value.

This peak current can be used to confirm the anticipated battery life for our speaker adapter, too, through dimensional analysis:

$$t_{battery} = \frac{Q_{battery}}{i} \approx \frac{1000\ mAh}{104.\,mA} \approx 9.6\ hours$$

This compares quite decently to our target battery life of about ten hours.

[2] Accurate current measurements are not available on the WT32; we've estimated based on the supply current drawn when the internal battery charger circuit is enabled (even though we aren't using that circuit).

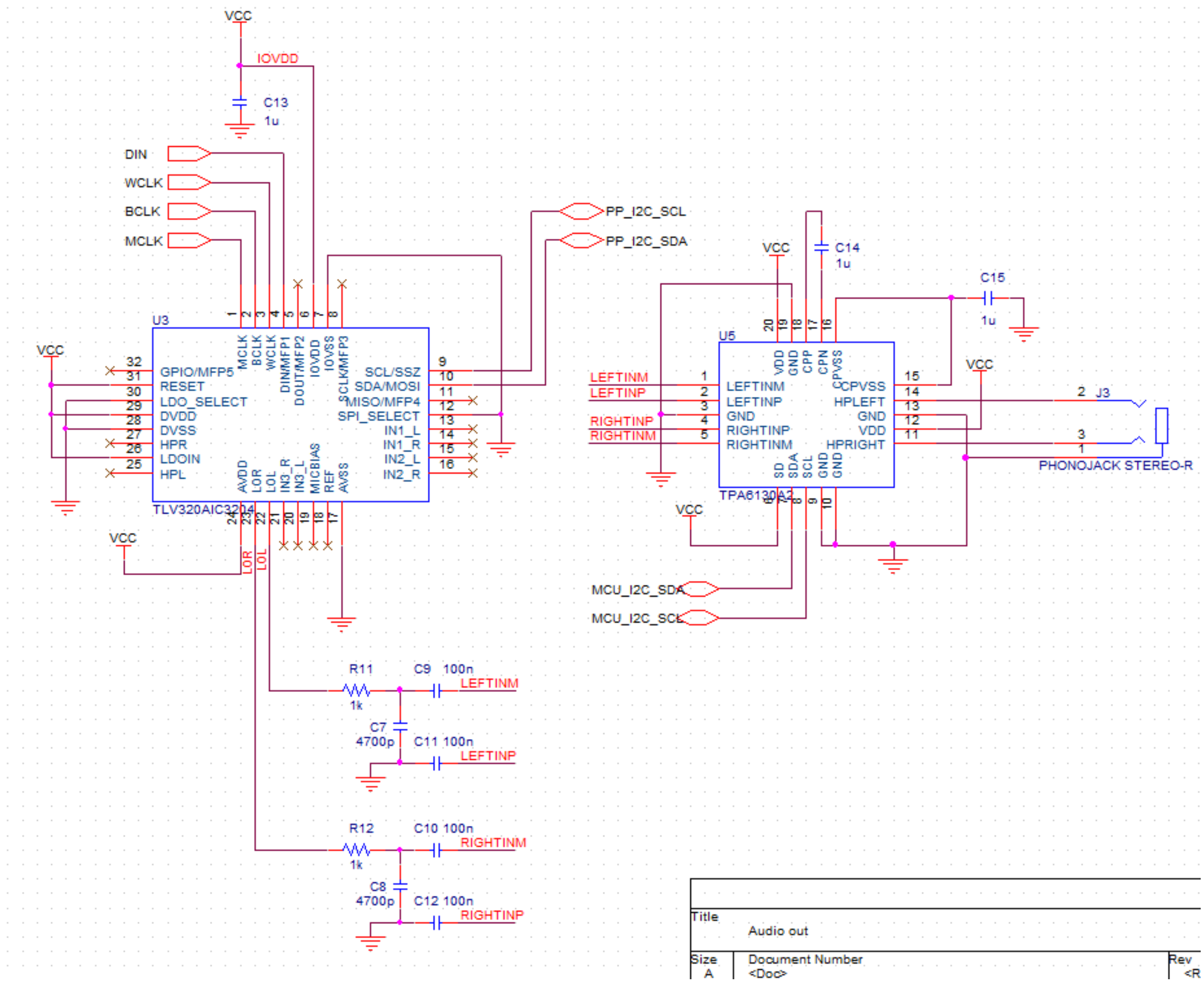


Fig 12: Audio frontend

2.2.1.6 Audio Frontend

The audio frontend of the speaker will include an audio codec that includes a DAC, as well as a high-power audio amplifier connected to a 3.5mm audio jack that a line-out wire can connect to. It takes in I2S data and outputs an amplified audio signal. The PurePath chip on the speaker unit controls the audio frontend via I2C.

The amplifier's operation is governed at boot by the MCU, which sends I2C directly to the amplifier. The amplifier's default mode is muted on the headphone line outputs so the MCU must tell the amplifier to work properly [8]. Overall, this unit serves to connect the PurePath unit, which outputs audio as I2S, to the system's analog output which connects to physical stereo speakers.

We selected the codec IC, the TLV320AIC3204, because it had an integrated LDO that allowed us to drive the entire IC at our only Vcc, 3.3V, which distinguished it form other codecs. We also sought an IC that could be controlled over both I2C and SPI, in case our circuit was misbehaving and we wanted an additional test point [6].

The critical circuit elements in the audio frontend compose the analog data's coupling between the two ICs. The codec is only able to drive single-ended audio data, so we have to be careful in the way that we connect this to the amplifier: in particular, we need to block DC voltages (capacitors C9-C12 do this) and to low-pass filter the output. Single-ended signals can be treated as differential signals by using ground as the paired signal.

2.2.2 Hub

2.2.2.1 Microcontroller Unit

The microcontroller unit (MCU) manages the operation of the hub. The hub is responsible for initializing the Bluetooth unit at boot to broadcast for pairing, and to manage the flow of pairing with the UE when the user attempts to pair. MCU is also responsible for choosing the appropriate speaker to forward audio data to, using the signal strength data it receives from the speaker (via the PurePath unit).

The circuit for the microcontroller in the hub is identical to that of the microcontroller in the speaker adapter.

The logical flow of the MCU program is shown in Figure 13.

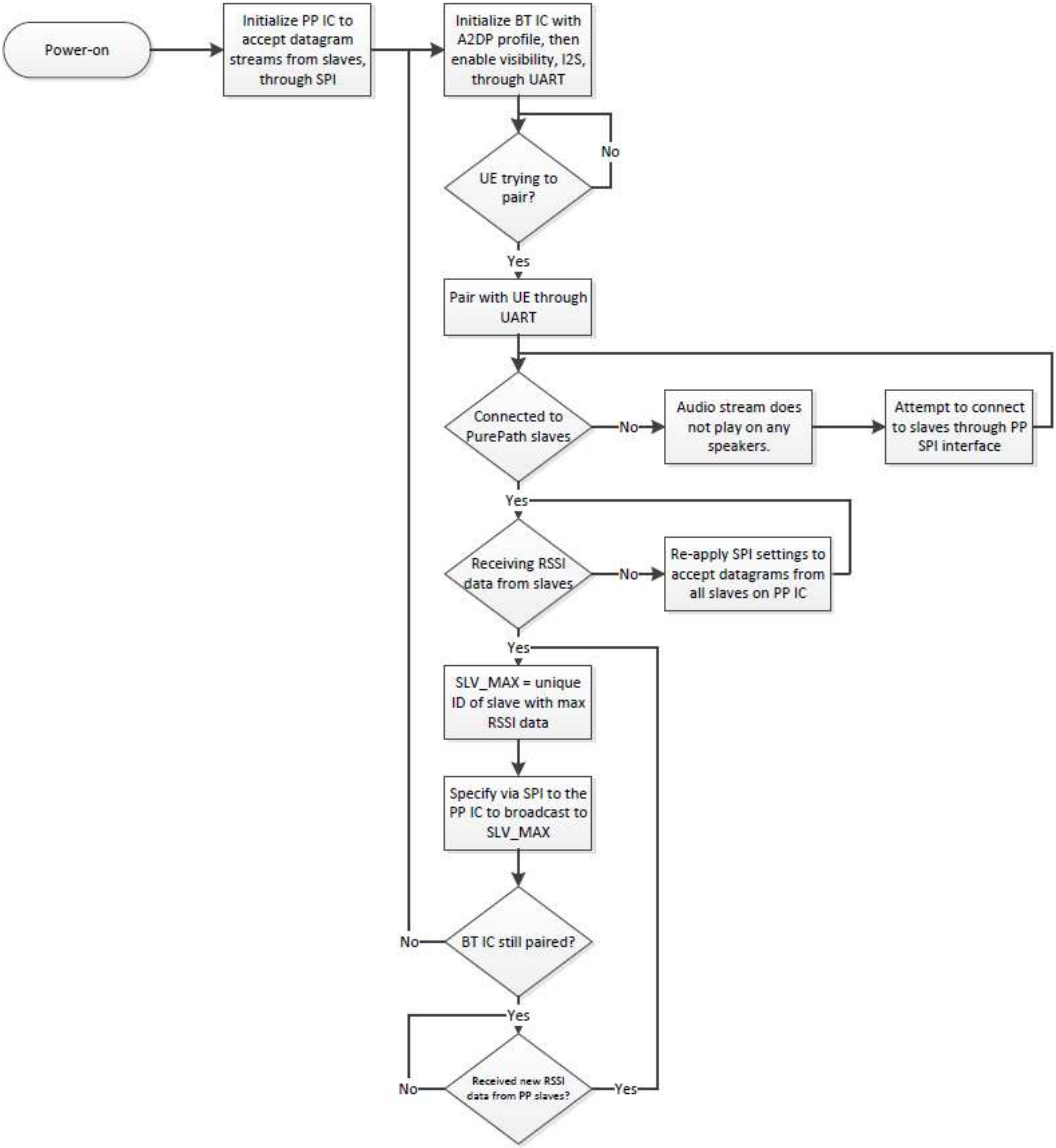


Fig 13: Hub MCU Flow

2.2.2.2 Bluetooth Unit

The Bluetooth (BT) Unit consists of the Bluetooth (BT) IC, an antenna, and a crystal oscillator providing the "slow clock" for the Bluetooth IC, all on one module. In the hub, the Bluetooth module will be used to pair with the UE, and A2DP (Advanced Audio Distribution Profile, a Bluetooth communication profile used to transmit stereo data) will be used to download the audio stream from the UE.

The Bluetooth IC speaks I2S directly to the PurePath unit, which will re-transmit the audio data to the speaker adapters. Additionally, the BT IC will speak UART to the microcontroller, which will be used to send the BT IC commands to enter pairing mode and set its visibility, as well as to enable the I2S output pins.

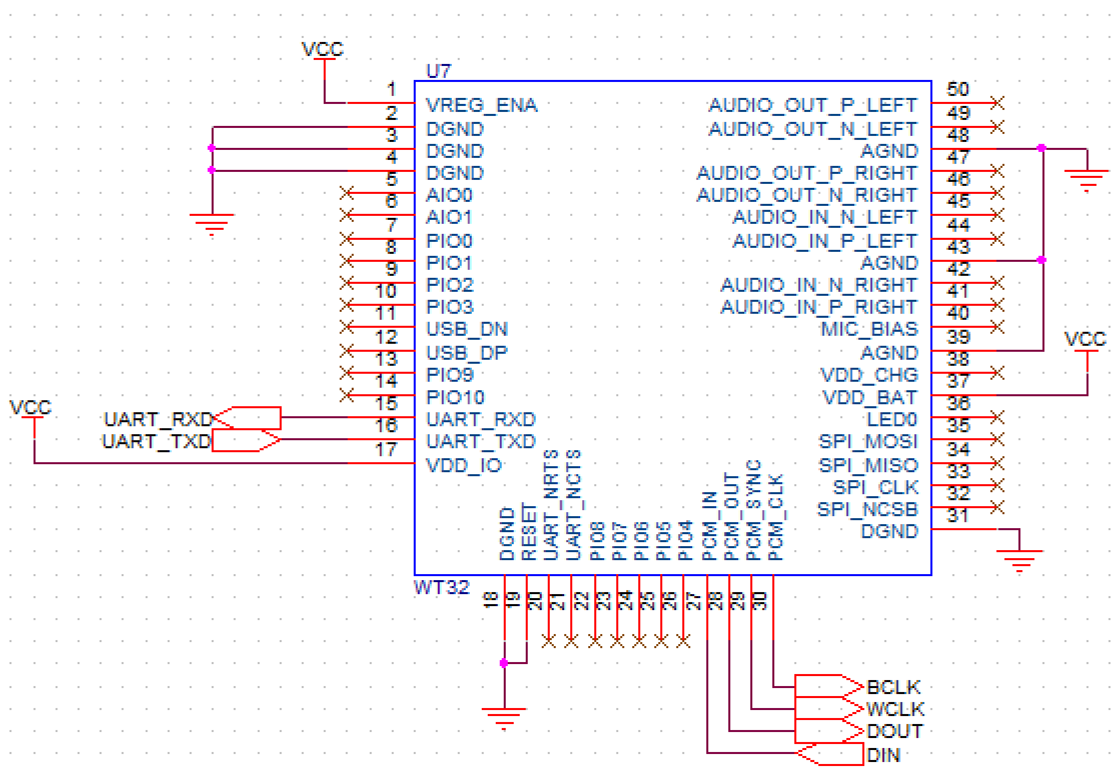


Fig 14: Bluetooth Unit

2.2.2.3 PurePath Unit

The PurePath Unit in the speaker consists of the PurePath IC and a crystal oscillator. In the speaker adapter the PP IC is responsible for encoding and decoding the PurePath radio signals. The microcontroller will send the BT signal strength data in the form of I2S, which will be broadcast back to the hub, while the received audio stream will be pushed to the audio frontend, also as I2S.

The PurePath IC is programmed directly using a PurePath IC programmer, but specific parameters will be configured at runtime by the microcontroller.

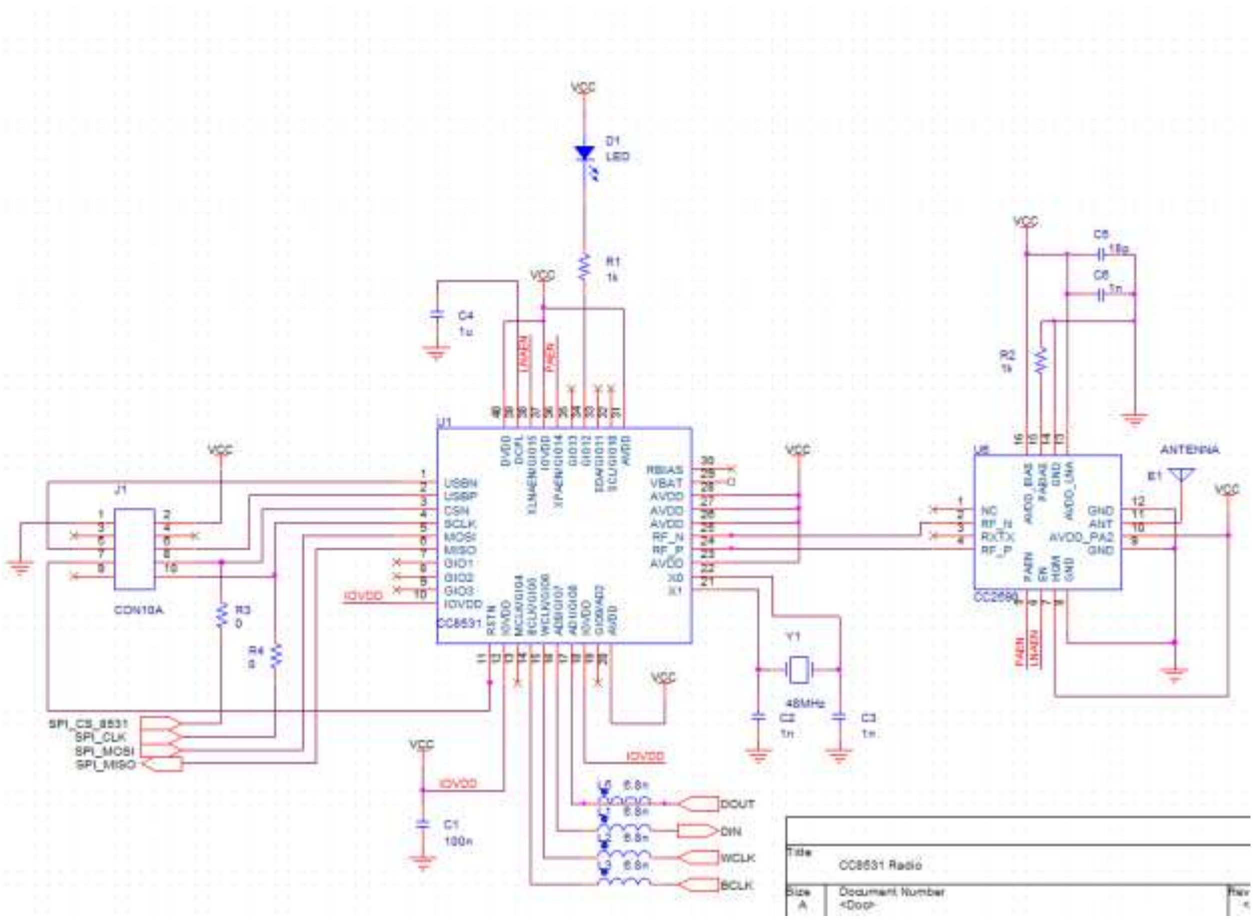


Fig 15: PurePath Unit

2.2.2.4 PurePath RF Front-End

The functionality of the PurePath RF Front-End will be similar to that used on the speaker adapter, so reference the description provided in the respective section.

2.2.2.5 Power Supply

The power supply in the hub will be identical to that of the speaker adapter, although we don't anticipate connecting the battery. For flexibility and commoditization reasons (to avoid making a new power supply circuit), we have left the capability to do so in the circuit, and therefore performed expected power behavior on the hub.

IC	Avg Current (mA)
CC8531	29
CC2590	22
MSP430G2553	0.42
WT32	4.5
Total	55.92

Table 2: Typical current consumption

The average current drawn by the ICs (other than the battery charger) is 55.92mA, as shown in Table 2. With a 1000mA battery, this would yield about

$$\frac{1000\ mAh}{56\ mA} \approx 17.9\ hours$$

This surprising result has led us to reconsider not using a battery in the hub. However, without accurate current data points for the Bluetooth, this may be misleading.

3.0 Requirements and Verification

3.1 Testing Procedure

Block	Requirement	Verification
Hub		
Microcontroller Unit (TI MSP430g2553)	1. Accepts signal strength datagrams encoded as I2S and picks a speaker based on that by sending SPI control packets addressed to the PurePath IC that specify the speaker to play to.	1. Drive test values on the SPI lines via the Aardvark SPI Host Adapter. The data will simulate the datagram stream returning from the PurePath IC with the signal strength data. The stream will include a speaker ID followed by the signal strength data. While sending stream strength data, cycle through letting each speaker be associated with the strongest signal strength values. The microcontroller must broadcast a correctly-formed SPI packet with an instruction to switch to the speaker with the strongest strength.
Bluetooth Unit (Bluegiga WT32)	1. Device must be visible over Bluetooth to UE by broadcasting its device name and UE must be able to connect to the Bluetooth module 2. UE must be able to stream audio to the BT unit using A2DP which is output from the unit as I2S	1. Develop test program that indicates, via UART, for the unit to be visible with a given name. Use a BT device (any phone) to pair with the Bluetooth unit 2. Develop test program that indicates, via UART, for the unit to present the A2DP BT profile. Then, connect to the unit through UE (any phone) and stream audio. Watch the I2S stream with a logic analyzer to verify continuous data stream.
PurePath Unit (i.e. TI CC8531 PurePath IC)	1. Convert provided I2S signal into PurePath proprietary signaling around 2.4GHz for broadcast over the air. 2. The output impedance of the PPIC chip must be Z = 70+j30 Ω as measured from the PNA. 3. Output power of the PPIC must not exceed 3.5dBm, while the receiver sensitivity ≈ -85dBm.	1. Program PurePath unit to always broadcast received I2S audio data. Generate I2S test signal using Aardvark Host Adapter and feed it into the audio input lines of the PurePath unit. 2. Characterize S-parameters on the PNA and verify that the output impedance is Z = 70+j30Ω. 3. Measure RF output signal from the unit on a VSA by verifying the signal resides at 2.4GHz with output power level of 3.5dBm.

PP RF Front-End (i.e. TI CC2590)	1. Combined maximum output power of PPIC + CC2590 must not exceed 11dbm. 2. Antenna must be conjugately matched to the output impedance of the Front-End.	1. Measure RF output signal strength on a VSA by verifying that the output power from the PPIC + Front-End doesn't exceed the specified limit of 11dBm. 2. Characterize the S-parameters of the device using a PNA and verify that the antenna port is conjugately matched to the effective output impedance as measured from the RF Front-End.
Power Supply (TI BQ24168)	1. Accept 5±0.5V line through USB micro-B receptacle and provides 3.3±0.3V rail.	1. Provide 4.5—5.5V input with bench power supply to USB micro-B receptacle via cut USB cable. Verify that at all valid input voltages yield valid rail voltages (3.0—3.6V) at the output of the regulator.
Speaker Adapter		
Microcontroller Unit	1. Must request signal strength data from Bluetooth unit over UART and publish that data in the form of datagram packets in the PurePath IC's host-controller interface format.	1. Power the Bluetooth unit and connect the UART lines to the microcontroller. Program microcontroller to request list of visible Bluetooth devices via UART. Observe SPI output lines (via Beagle SPI Protocol Analyzer) reflect the proximity of the UE to the antenna.
Bluetooth Unit (Bluegiga WT32)	1. Silently listens to Bluetooth signals but never actively broadcasts. 2. Able to list available BT devices within 20ft and their signal strength values	1. Enable unit execution and measure RF output signal from the unit on a VSA to verify that no new signals reside around 2.4GHz. 2. Program microcontroller to request list of visible devices over UART, and broadcast this on SPI. Listen to the SPI bus via the Beagle SPI Protocol Analyzer and verify that UE shows up in the list when Bluetooth is enabled within a 20ft radius. Verify that the signal strength value adjusts according to the proximity of the UE to the antenna.
PurePath Unit	*1. Same requirements as in unit in the hub.*	*1. Same verification steps as in the unit in the hub.*
PP RF Front-End	*1. Same requirements as in unit in the hub.*	*1. Same verification steps as in the unit in the hub.*

Power Supply	1. Accepts 5±0.5V line through USB micro-B receptacle and provides 3.3±0.3V rail.	1. Provide 4.5—5.5V input with bench power supply to USB micro-B receptacle via cut USB cable. Verify that at all valid input voltages yield valid rail voltages (3.0—3.6V) at the output of the regulator.
Audio Front-End	1. Accepts I2S audio signals and drives analog audio signals varying from 50Hz to 15 kHz into a 3.5mm receptacle.	1. Use Aardvark SPI Host Adapter to generate I2S audio signals ranging from 50-15kHz and observe on the oscilloscope that audio at receptacle reflects the input frequency.

3.2 Tolerance Analysis

This system needs to be incredibly robust at detecting the speaker and playing back to it. We do not want audio switching incessantly from one speaker to the other when a given speaker is on the cusp of range. The hub-speaker range is critical in the performance of our circuit, and we estimate that we will be able to playback on speakers at least 20ft away from the hub. In order to meet this tolerance, we envision that we need the RF frontend to output power at 14dBm with a noise figure of 4±1dB for our system work seamlessly as proposed. To ensure that we meet these tolerance specifications we will measure the output power of the RF front-end using a VSA.

4.0 Ethical Considerations

The purpose of the Bluetooth stereo network is to simplify and enhance the user's music listening experience when moving about a room. We strive to complete this project bearing in mind the IEEE Code of Ethics in Section 7.8 of IEEE Policies [9]. The most relevant policies pertaining to our project are detailed as follows:

3. *To be honest and realistic in stating claims or estimates based on available data;*

During each step along the design process, we have included calculations and a testing procedure that involves verification of all requirements. We aim to meet all our requirements, which are based on realistic comparisons with similar products as well as our own tests.

5. *To improve the understanding of technology; its appropriate application, and potential consequences;*

Throughout the course of this project we have learned much about Bluetooth and other technologies. In examining different wireless protocols and standards, we come to understand their appropriate usage. This statement of ethical considerations also represents our investigation into the potential consequences of using technology.

6. *To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;*

Our goal for this course is to create a project that meets or exceeds expectations the University of Illinois has of us in order to be professional engineers. Through the design process, each team member has improved his technical expertise in areas that were not taught in classes in addition to using the practical application of coursework. We will gain knowledge and training in any technological task we undertake.

7. *To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;*

We fulfill this part of the code of ethics via the design review process. Based on feedback and criticism from our peers, professors, and TAs, we go forth in our design addressing such concerns. We will properly credit the sources using the IEEE citation format so as to always give proper credit. This consideration also applies to communication between team members, as we must remain vocal in bringing forth issues and be open in sharing technical knowledge as we collectively strive towards a successful project.

9. *To avoid injuring others, their property, reputation, or employment by false or malicious action;*

Our project's express purpose is for location-based music playback. We will not use any malicious hacks on Bluetooth such as bluejacking and bluebugging, which can result in damage to reputation and employment if the user's cell phone is exploited [10].

In addition to the IEEE Code of Ethics, we must also satisfy FCC Part 15 requirements, which govern acceptable use of radio frequency devices. Under these regulations, our Bluetooth stereo network falls under Class B digital devices, which are marketed to the general public for home use. While part 15 is quite lengthy, we will make sure to adhere to sections 15.247 and 15.249, which govern the 2.4-2.483 GHz band we are using [11].

5.0 Cost and Schedule

5.1 Cost Analysis

5.1.1 Labor

Name	Hourly Rate	Total Hours Invested	Total = Hourly Rate x 2.5 x Total Hours Invested
Jeff Wheeler	$75.00	150	$28,125
Rishi Ratan	$75.00	150	$28,125
Jerry Sun	$75.00	150	$28,125
Total		450	$84,375

5.1.2 Parts

Part	Count	Unit Cost ($)	Total Cost ($)
MSP430G2553	3	2.73	8.19
CC8531	3	8.77	26.31
CC2590	3	5.18	15.54
BQ24168	3	5.83	17.49
TPA6130A2	2	2.30	4.60
TLV320AIC3204	2	5.83	11.66
WT32	3	29.67	89.01
Capacitors	63	0.05	3.15
Resistors	45	0.05	2.25
Inductors	18	0.10	1.80
LEDs	9	0.05	0.45
Connectors	15	0.50	7.50
2.4GHz Ant	3	4.00	12.00
PCBs	3	30.00	90.00
Parts Total			**289.95**

5.1.3 Grand Total

$$\boldsymbol{Total\ Cost} = Labor\ Costs + Parts\ Costs = \$84{,}375 + \$289.95 = \$\mathbf{84{,}664.95}$$

5.2 Schedule

Week	Jeff	Rishi	Jerry
9/17	Proposal: Finalize system block diagrams and Requirements/Verification for each block.	Proposal: Finalize system level block descriptions, introduction/features and defined RF Requirements/Verifications. Ordered initial parts.	Proposal: Developed initial proposal structure, supervised overall block descriptions, tolerance analysis and cost.
9/24	Design schematic for MCU and its interface to PurePath IC. Analyze I2C/I2S signals generated by MCU using NI-DAQ, working towards software testing for Design Review.	Design and simulate the power supply circuit. Conduct simulations in Spice and complete the circuit design. Characterize the RF chips used in the PurePath RF Front-End circuit to verify their compatibility as per the design requirements highlighted in proposal.	Evaluate Bluetooth chips and verify their programmability. Test I2C integration into the analog audio front-end. Research UART interface to Bluetooth IC.
10/1	Design Reviews and create MCU PCB footprints, then begin layout, optimizing for board size.	Design Review and learn about PCB layout and make footprints for RF and Power ICs.	Design Review and learn about PCB layout and make footprints for BT and PP ICs.
10/8	Layout PCB and verify that it matches schematic. Start to write MCU programs for each device, and program the PurePath ICs.	Prototype power-supply design on breadboard and test integration of RF front end circuit for PurePath ICs.	Prototype analog audio front-end design on breadboard and test integration of BT ICs and experiment with UART.
10/15	Adjust PCB layout according to discoveries from debugging breadboard circuits, and continue to write MCU programs.	Order new parts as needed (including full BOM for completed boards). Debug breadboard circuits from previous week. Consider ordering first-round PCBs if debugging is going well.	Continue debugging UART interface to BT ICs on hub and speaker adapters, and learn SPI interface for PurePath ICs.
10/22	Complete Individual Reports.	Complete Individual Reports.	Complete Individual Reports.
10/29	Order next set of PCBs, updated according to initial testing. Integrate UART and SPI interfaces into MCUs.	Validating design choices for PCB RF circuits and verifying RF front-end behavior. Analyze battery behavior with rechargeable circuit.	Assemble PCBs, mounting almost exclusively QFN ICs. Finalize UART and SPI interface with BT ICs, PP ICs, and MCU.
11/5	Prepare for Mock-up Demos	Prepare for Mock-up Demos	Prepare for Mock-up Demos
11/12	Prepare Mock-up Presentations and finalize final design.	Prepare Mock-up Presentations and finalize final design.	Prepare Mock-up Presentations and finalize final design.
11/19	Thanksgiving Break	Thanksgiving Break	Thanksgiving Break
11/26	Consolidate work and start writing the final project report.	Consolidate work and start writing the final project report.	Consolidate work and start writing the final project report.
12/3	Project Demo	Project Demo	Project Demo
12/10	Final Design Presentations!	Final Design Presentations and Graduation!	Final Design Presentations!

6.0 References

[1] Texas Instruments, "Mixed Signal Microcontroller," MSP430G2x53 datasheet, Apr. 2011 [Revised Aug. 2012].

[2] Bluegiga Technologies, "WT32," WT32 datasheet, Aug. 2009 [Revised Jan. 2012].

[3] Texas Instruments, "2.4 GHz RF SoC For Wireless Digital Audio Streaming CC8520, CC8521, CC8530 & CC8531 – PurePath Wireless," CC8531 datasheet, Jun. 2012.

[4] Texas Instruments, "2.4-GHz RF Front End, 14-dBm output power," CC2590 datasheet, Sept. 2008.

[5] Texas Instruments, "PurePath Wireless Headset Reference Design," CC85XXDK-Headset-RD Technical Documents, Apr. 2011.

[6] Texas Instruments, "Ultra Low Power Stereo Audio Codec," TLV320AIC3204 datasheet, Sept. 2008 [Revised Oct. 2008].

[7] Texas Instruments, "2.5A, Dual-Input, Single Cell Switchmode Li-Ion Battery Charger with Power Path Management and I2C Interface," bq24168 datasheet, Nov. 2011 [Revised Sept. 2012].

[8] Texas Instruments, "138-mW Directpath Stereo Headphone Amplifier with I2C Volume Control," TPA6130A2 datasheet, Nov. 2006 [Revised Feb. 2008].

[9] Institute of Electrical and Electronics Engineers, Inc. (2012, September 10) *IEEE Policies* [Online]. Available: http://www.ieee.org/documents/ieee_policies.pdf

[10] *Bluetooth Tutorial* [Online]. Available: http://www.radio-electronics.com/info/wireless/bluetooth/bluetooth_overview.php

[11] Federal Communications Commission. (2012, October 1). *Part 15—Radio Frequency Devices* [Online]. Available: http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?c=ecfr&sid=9a73adb8343432691773b02065ccf867&tpl=/ecfrbrowse/Title47/47cfr15_main_02.tpl