WICED : Wireless Internet Connectivity for Embedded Devices

Authors: Leo Schultz, Bryan and Vikram
Audience:

This document is intended for Tele communication and wireless networking engineers who are interested in current generation wireless security technologies. The challenges that wireless users are facing with security issues when compared with wired network.

This paper will discuss on various topics of wireless security standards and new concepts that are available to help secure communication via non wire.

Document Revision history

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description/section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feb 22nd 2014</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>2</td>
<td>March 17th 2014</td>
<td>Add content to section 3</td>
</tr>
<tr>
<td>3</td>
<td>March 17th 2014</td>
<td>Added flow diagram and charts</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Contributors and Acknowledgement:

The information included in this document represents the expertise, feedback and suggestions of a number of people. The author would like to recognize the support and thank the following reviewers of this document:

- Prof. Ming Wang – Santa Clara University.
# Table of Contents

2 Introduction .................................................................................................................. 4  
   Objective ..................................................................................................................... 4  
   The problem ............................................................................................................... 4  
   Why this is a project related to this class ................................................................. 4  
   Why other approach is not good ............................................................................. 5  
   Why you think your approach is better ................................................................ 5  
   Statement of the problem ....................................................................................... 5  
   Area or scope of investigation ............................................................................. 5  

3. Theoretical bases and literature review ....................................................................... 5  
   Definition of the problem ....................................................................................... 5  
   Theoretical background of the problem ................................................................. 6  
   Related research to solve the problem .................................................................. 6  
   Advantage/disadvantage of those research .......................................................... 6  
   Your solution to solve this problem ..................................................................... 7  
   Where your solution different from others ............................................................ 7  
   Why your solution is better ................................................................................... 7  

4. Hypothesis (or goals) hypothesis .............................................................................. 7  

5. Methodology ............................................................................................................ 7  
   How to generate/collect input data ....................................................................... 7  
   How to solve the problem ..................................................................................... 7  
   Algorithm design ..................................................................................................... 7  
   Language used ......................................................................................................... 8  
   Tools used ............................................................................................................... 8  
   A prototype (optional if time permit) .................................................................... 8  
   How to generate output ......................................................................................... 8  
   How to test against hypothesis ............................................................................ 8  

6. Implementation ......................................................................................................... 8  
   Code (refer programming requirements) ................................................................. 8  
   Design document and flowchart ......................................................................... 8  

7. Data analysis and discussion ..................................................................................... 12  
   • Output generation ............................................................................................... 12  
   • Output analysis .................................................................................................. 12  
   • Compare output against hypothesis ................................................................. 12
• Abnormal case explanation (the most important task) .............................................. 12
• Discussion .................................................................................................................. 13

8. Conclusions and recommendations ........................................................................... 13
   Summary and conclusions ........................................................................................... 13
   Recommendations for future studies ........................................................................... 13

9. Bibliography .............................................................................................................. 13
10. Appendices ................................................................................................................ 13
    Program flowchart / input/output charts ...................................................................... 13
    Program source code with documentation .................................................................. 16
    Input/output listing ...................................................................................................... 19
    Other related material .................................................................................................. 20
2 Introduction

Wireless communications offer organizations and users many benefits such as portability and flexibility, increased productivity, and lower installation costs.

With the proliferation of digital content and the expanding variety of connected and IP-enabled consumer electronics (CE) devices, consumers are increasingly seeking ways to efficiently stream high definition (HD) video amongst their home networking devices. The fully-connected home is enabling exciting new applications and easy digital media sharing between CE devices, mobile handsets, set-top boxes (STBs), and personal computers (PCs) that has changed the way people interact and share HD video in the next-generation digital home.

The key means for enabling these connectivity technologies to work together is to bring together proven and established network and multimedia standards into a single consistent and reliable framework that serves developers, consumers and service providers alike. To this end, Digital Living Network Alliance (DLNA) has developed robust, technical guidelines defining device classes and networked home use cases which, when built into digital TVs, STBs, Blu-ray players and other devices allow high-quality streaming of multimedia content over wireless and wired connections.

WICED is well placed to drive the Internet of Things and connecting everything to Internet. (Pronounced "wik-id") eases development effort and simplifies the implementation of Internet connectivity in an array of consumer devices, especially those without existing support for networking.

Objective

We plan to implement audio streaming over WIFI using WICED Module.

The problem

Current solutions that we have in market are wired and wireless solution available is at premium cost. The existing solution with Bluetooth is having latency issues and hasn’t met the user demand while streaming audio. We have never been able to stream audio so cheap due to memory or high RAM related issues in previous solutions. Synchronous streaming has been a problem. We believe we can do much better with the advancement over the wireless cards that are available now.

Why this is a project related to this class

We believe Computer Networks is related to communicating effectively and efficiently between devices and what we are trying to do or accomplish in this project is about sending audio streams over WIFI networks which is related to this session or class.
Why other approach is not good

The current solution has issues with cost, latency or synchronization and each one is these solutions are mutually exclusive.

Why you think your approach is better

The Broadcom technologies have come up with modern solution named WICED WI-FI module and which is the best solution that’s

- Cheap in cost.
- SDK kit that can enable to program for plug and play related testing.
- Opened to more advanced features.
- Synchronization, Auto tuning or self-adjustment channels.
- Low latency can be achieved.

Statement of the problem

Current Bluetooth solutions have synchronization problem and are not cost effective for streaming audio. Users were able to feel the lag while audio streaming via Bluetooth. We plan to implement a low cost quality solution that enables consumers to synch their audio streaming over WIFI using Broadcom WICED module that confirm to DLNA standards, while keeping the cost of this solution minimal

Area or scope of investigation

- Learning WICED SDK
- Understand the DSP ( Digital signal processing for Audio )
- Transfer Formats.
- QT- Framework understanding.

3. Theoretical bases and literature review

Definition of the problem

Audio Streaming over Bluetooth has ran into numerous issues with connectivity, streaming etc. especially lag issues etc. Bluetooth which is a a wireless connectivity technology, is designed to be relatively easy to use, and typically it requires little more than entering a four-digit PIN to instantly pair, say, a keyboard with your iPad. And yet, for all its purported simplicity, online forums are full of users complaining about Bluetooth problems. While the connectivity is good at the lower power level there are streaming issues observed and its affecting the user experience very badly particular delay or lag while streaming the audio is very bad. Our solution is to use Wireless 802.11 for good audio streaming while keeping the cost at minimum we should be able to achieve better quality than Bluetooth.
Theoretical background of the problem

Far the most popular short-range wireless technology for device-to-device data transfers. And it's everywhere—in smartphones, laptops, cameras, televisions, car dashboards, and home audio systems. You can use it to stream music wirelessly from your PC to a Bluetooth-enabled smartphone or MP3 player. Many people use Bluetooth for hands-free calling, and Bluetooth headphones are also a popular choice for listening to music on the go.

This technology ships in more than 19 million devices every week, according to the Bluetooth Special Interest Group, a trade association dedicated to advancing Bluetooth technology. The Bluetooth SIG has more than 15,000 member companies, including major consumer technology firms such as Apple, Intel, Lenovo, Microsoft, Nokia, and Toshiba.

The most recent version of Bluetooth, version 4.0, includes new low-energy protocols for transmitting data from medical devices such as glucose monitors and stethoscopes, and other low-powered gadgets like pedometers and watches. Bluetooth 4.0 also includes the capabilities of its predecessor, 3.0 + HS, which improved Bluetooth's data-transfer capabilities thanks to its ability to pair two devices and then switch large data loads from Bluetooth to the 802.11 wireless standard, the same standard used for Wi-Fi connectivity.

Related research to solve the problem

With the emergency of need for transferring the audio over WIFI there were newer and advanced chip developed by Broadcom to address the consumer electronic devices to connect to wireless networks available. WICED module reduces the effort required to add wireless connectivity to embedded devices. The SDK enabled developers to quickly create network connected applications targeted for low-resources microcontrollers.

Advantage/disadvantage of those research

Wireless technologies acquire more important place in market every day. Their utilization is being investigated not only in telecommunication, but also in industry. The advantages that wireless technologies offer, lead to various number of possible applications and services, which may be used in industrial fields. Intensive research proceeds at IP based controlling over wireless networks, which will together with Internet relieve remote control.

Though the wires limit properties of many applications, they still guarantee the main issues of industrial requirements – real-time communication (speed) and reliable link which is an important feature for voice streaming as well. Bluetooth controlled devices can become easily susceptible to microwave that we normally use at home.
Your solution to solve this problem

- QT – Framework can help us to build the UI (user interface)
- WICED-Chip can help us to intercept audio data for streaming

Where your solution different from others

- Current available solution use 802.15 which has higher latency.
- WICED has the capability to use 802.11.
- Open sourced.
- Our solution is Cheaper than the existing solutions in the market.

Why your solution is better

- WICED is the cheaper and most advanced networking solution for Embedded Devices that comes with a SDK. These Libraries will enable us to do authentication, synchronization and other advanced functions.
- We use 802.11 instead of 802.15.
- QT is good because of cross platform support, the fastest that’s available than the traditional framework.

4. Hypothesis (or goals) hypothesis

We can achieve higher quality audio streaming over WiFi networks.

5. Methodology

How to generate/collect input data

- Input is primarily from consumer with following options: Audio type, Audio File, Grouping of speakers, Authentication information, Connection Info, Power State of the audio speakers. EQ data.
- Speaker level input data would be, volume, power state, audio data, channel (Left or Right), and Equalizer changes.

How to solve the problem

Algorithm design

Take the volume, EQ changes and Audio data and run through filter algorithm.
Language used

- Java Script
- QML – Query markup language
- C, C++

Tools used

- WICED SDK 2.3.0
- QT 5.2
- WICED Chip from Broadcom
- Real hardware: Speakers (Full range), Audio AMP (Mono), Wolfson Mono DAC (digital to analog converter)

A prototype (optional if time permit)

How to generate output

- Quality Audio Sound from the Speakers.
- User ability to control the speakers.
- Authentication
- Power level

How to test against hypothesis

By Measuring the latency and quality of audio with our solution verse Blue tooth and synchronization of Audio.

6. Implementation

Code (refer programming requirements)
Refer the last section of the for code.

Design document and flowchart

This below flow chart explains the wiring diagram of the speaker module integration.
Figure 1: solution of implement the low cost wireless audio streaming solution with WICED

Figure 2: Prototype: Mobile app control functions for scanning, monitoring and controlling
Functions from Mobile Apps:

User Action flow charts:

1) Create Account
2) Login the user
3) Main Screen
4) Request prompt
5) Request query
Speakers Related Flow charts

1) Speaker Scan
2) Speaker connection
3) Speaker authentication

Speaker Settings

1) Group setting
2) EQ settings
3) Power settings
7. Data analysis and discussion

- **Output generation**
  
  Live Streaming of Audio over speakers by using 802.11 standards.

- **Output analysis**
  
  - Low Latency and High quality of audio stream via speakers when compared with Bluetooth streaming which has perceived lag.
  - Synchronization and low cost comparison

- **Compare output against hypothesis**
  
  Hypothesis has 2 aspects one is quality of the audio when compared with Bluetooth and wireless and other one is the cost aspect of the solution.

  Cost Benefit analysis for this Customer solution is as follows:

  ![Cost graph]

  The Above figure explains affordability factor with WICED card usage.

- **Abnormal case explanation (the most important task)**
  
  Support of Bluetooth and Wireless together can be difficult, shifting the channels and streaming the audio will be an area to be concerned. We have to understand more in this area to see interworking between standards at the consumer end.

  WICED AUDIO as advertised didn’t have inbuilt speaker to route the audio signals.
• Discussion
Understanding power usage pattern and QoS parameters around audio quality can be explored in details.

8. Conclusions and recommendations

Summary and conclusions
This project indeed has lot of front apps from the mobile phones and backward integration of WICED module to work with any wireless Access Point. Mono, AMP speakers and audio board integration is also important. Understanding of various 802.11 Standards and DLNA specific standards is curtail to have a go to market strategy with low cost solution can be derived.

Recommendations for future studies

WICED can integrate seamlessly with other electronic devices. We can connect any micro controlled device to wireless LANs. Engineers can start to explore connecting everything to internet using WICED.

9. Bibliography

• An Open Source and Low Cost Solution for Consumer Electronics Middleware Validation
  Gustavo M. Calixto, et al. 2013 17th International Symposium on Consumer Electronics (ISCE)
• Characterizing Wi-Fi Link Performance in Open Outdoor Networks, 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks

10. Appendices

Program flowchart / input/output charts

For Mobile APPs
Create Account

Back Button – go to login view
Input Form
-First Name (Text)
-Last Name (Text)
-Email (Text)
-Password (Text, Pass)
-Password Check (Text, Pass)
-Agree to Conditions (Checkbox)
-Create Button – Save Info for User Account

Input Form
-Email (Text)
-Password (Text, Pass)
-Submit (Button) – Enter Account

Save (Button) – Holds email in device default
Switch User (Button) – Alternate to save function, breaks user email out of default.
Recover (Button) – Sends email with password reset info
Create User (Button) – Link to Create User View

Main Screen

Header:
Power (Button) – Turns Selected/Controlled Speaker On or Off
Mute (Button) – Mutes Selected/Controlled Speaker
Refresh (Button) – Refreshes App and Speaker Connection
Mic (Button) – Takes Voice Input Data from Phone and Sends it to the Speaker for output
Back (Button) – Brings Child view Back to Parent view

Body:

Outputs
Speaker Name
Single Speaker = Speaker Type or Speaker Group = Number of Speaker
Battery Life
Battery Time Left
Speakers Power State (On, Off or Out)

Inputs
Choose Speaker (Double Click) – Makes Speaker “in use” for Device Audio output, if not “in use” by someone else
Add Speaker (Button) – Link to Scanning for Speakers and then to Speaker Connection
Request Speaker (Button) – if speaker selected and in use request is sent to user, if selected and not in use then taken over, if none selected takes you to request query view
Edit Group (Button) – if group selected, brings user to groups editing view
UnGroup (Button) – if group selected, ungroups speakers to individual speakers
Group (Button) – brings speakers into group, button brings to grouping view for creating group

EQ Settings

Header:
Power (Button) – Turns Selected/Controlled Speaker On or Off
Mute (Button) – Mutes Selected/Controlled Speaker
Refresh (Button) – Refreshes App and Speaker Connection
Mic (Button) – Takes Voice Input Data from Phone and Sends it to the Speaker for output
Back (Button) – Brings Child view Back to Parent view

**Body:**
Settings Nav (Button) – Brings to Settings for Group or Speaker
Treble (Slider) – Adjusts Treble
Hi Mid (Slider) - Adjusts Hi Mid
Mid (Slider) - Adjusts Mid
Lo Mid (Slider) - Adjusts Lo Mid
Bass (Slider) - Adjusts Bass
dB (Slider) - Adjusts dB
Bass Boost (Spinner) - Adjusts Bass Boost
Balance (Spinner) - Adjusts Balance
Virtualizer (Spinner) - Adjusts Virtualizer
Presets (Drop-down) – Sets to Music Type (Custom, Rock, Hip-Hop, Dance, Country, Classical)
Save (Button) – Saves Changes to EQ
Reset (Button) – Resets and Save EQ settings to Default

**Group Settings**

**Header:**
Power (Button) – Turns Selected/Controlled Speaker On or Off
Mute (Button) – Mutes Selected/Controlled Speaker
Refresh (Button) –刷新es App and Speaker Connection
Mic (Button) – Takes Voice Input Data from Phone and Sends it to the Speaker for output
Back (Button) – Brings Child view Back to Parent view

**Body:**
Edit/Save (Button) – Saves or Edits the Group Info
EQ Nav (Button) – Brings to EQ Settings for Group

**Output**
Group Name
Group Image Name
Speaker Name
Speaker Channel (Left, Right, Center, Mono)
Speaker Type
Input – If Edit Button Chosen
Group Name (Text) – Renames Grouping
Group Image (Drop down) – Changes Grouping Image
Speaker Settings (Button) – Brings to Speakers Settings View
Delete Speaker (Button) – Takes speaker out of group
Program source code with documentation

/*
 * Copyright 2014, Broadcom Corporation
 * All Rights Reserved.
 * This is UNPUBLISHED PROPRIETARY SOURCE CODE of Broadcom Corporation;
 * the contents of this file may not be disclosed to third parties, copied
 * or duplicated in any form, in whole or in part, without the prior
 * written permission of Broadcom Corporation.
 */

/** @file
 * UDP Receive Application
 * This application snippet demonstrates how to receive a UDP packet
 * from a network client (and optionally send a response)
 * Features demonstrated
 * - Wi-Fi softAP mode
 * - DHCP server
 * - UDP receive (and response)
 *
 * Application Instructions
 * 1. Connect a PC terminal to the serial port of the WICED Eval board,
 * then build and download the application as described in the WICED
 * Quick Start Guide
 * 2. Ensure Python 2.7.x (*NOT* 3.x) is installed on your computer
 * 3. Connect your computer using Wi-Fi to "WICED UDP Receive App"
 * - SoftAP credentials are defined in wifi_config_dct.h
 * 4. Open a command shell
 * 5. Run the python UDP transmit script as follows from the udp_receive
 *    dir
 *    - Ensure your firewall allows UDP for Python on port 50007
 *    - The WICED application starts a softAP, and then regularly checks to
 *      see if a UDP packet has been received from a connected client.
 *      An optional response is sent directly to the client.
 * - The udp_transmit script sends a broadcast UDP packet from your computer

...
to the WICED application. The received packet contents are extracted by the
WICED application and printed to the terminal.
* To disable the UDP response, comment out the #define SEND_UDP_RESPONSE
*/

#include "wiced.h"
#include "platform_audio.h"

/******************************************************************************
* Macros
*******************************************************************************/
#define PORTNUM           50007           // UDP port
#define UDP_SERVER_THREAD_PRIORITY          (WICED_DEFAULT_LIBRARY_PRIORITY)
#define UDP_SERVER_STACK_SIZE   (60)

/******************************************************************************
* Function Declarations
*******************************************************************************/
static void process_received_udp_packet(uint32_t arg);
static void transmit_over_i2c(uint32_t arg);

/******************************************************************************
* Variables Definitions
*******************************************************************************/
static const wiced_ip_setting_t device_init_ip_settings =
{
    INITIALISER_IPV4_ADDRESS( .ip_address, MAKE_IPV4_ADDRESS(192,168,  0,  1) ),
    INITIALISER_IPV4_ADDRESS( .netmask, MAKE_IPV4_ADDRESS(255,255,255,  0) ),
    INITIALISER_IPV4_ADDRESS( .gateway, MAKE_IPV4_ADDRESS(192,168,  0,  1) ),
};

wiced_i2c_device_t wm8533_control_port =
{
    .port          = WICED_I2C_1,
    .address       = 0x1A, /* 0011010b */ //0x25,  /* 100101 */
    .address_width = I2C_ADDRESS_WIDTH_7BIT,
    .speed_mode    = I2C_STANDARD_SPEED_MODE, //I2C_LOW_SPEED_MODE,
};

static wiced_thread_t      udp_thread;
static wiced_thread_t      transmit_thread;
static wiced_udp_socket_t  udp_socket;
wiced_i2c_message_t* buffer[256];
wiced_packet_t* packet[256];
int currentRead;
int currentWrite;

/******************************************************************************
* Function Definitions
*******************************************************************************/
**void application_start(void)**

- **Initialise the device and WICED framework**
  
  ```c
  wiced_init();
  wiced_i2c_init(&wm8533_control_port);
  currentRead = 0;
  currentWrite = 0;
  ```

- **Bring up the network interface**
  
  ```c
  wiced_network_up( WICED_STA_INTERFACE, WICED_USE_EXTERNAL_DHCP_SERVER, NULL);
  ```

- **Create UDP socket**
  
  ```c
  if (wiced_udp_create_socket(&udp_socket, PORTNUM, WICED_STA_INTERFACE) != WICED_SUCCESS)
  {
    WPRINT_APP_INFO("UDP socket creation failed\n");
  }
  ```

- **Register functions to process and transmit UDP packets**
  
  ```c
  wiced_rtos_create_thread(&udp_thread, UDP_SERVER_THREAD_PRIORITY, "UDP Audio Receiver", process_received_udp_packet, UDP_SERVER_STACK_SIZE, NULL);
  ```

**static void process_received_udp_packet(uint32_t arg)**

```c
{  
    wiced_packet_t* packet;
    char* rx_data;
    static uint16_t rx_data_length;
    uint16_t available_data_length;
    static wiced_ip_address_t udp_src_ip_addr;
    static uint16_t udp_src_port;
    WPRINT_APP_INFO("starting thread\n");
    /* Wait for UDP packet */
    while(1)
    {  
        WPRINT_APP_INFO("Test %d\n", i);
        i++;  
        wiced_result_t result = wiced_udp_receive(&udp_socket, &packet, 1000  
        if (result == WICED_ERROR)
        {  
            WPRINT_APP_INFO("ERROR ...\n");
        }  
        if(result == WICED_TIMEOUT)
        {  
            WPRINT_APP_INFO("Timeout waiting ...\n");
        }
        else
        {  
            // Process the received packet

```
/* Get info about the received UDP packet */
wiced_udp_packet_get_info(packet, &udp_src_ip_addr, &udp_src_port);

/* Extract the received data from the UDP packet */
wiced_packet_get_data(packet, 0, (uint8_t**)&rx_data, &rx_data_length, &available_data_length);

wiced_i2c_message_t* i2c_message = malloc(sizeof(wiced_i2c_message_t*));
wiced_i2c_init_tx_message(i2c_message, rx_data, rx_data_length, 3, 1);
buffer[currentWrite] = i2c_message;
packets[currentWrite] = packet;
currentWrite++;
if(currentWrite > 255)
{
    currentWrite = 0;
}
}

WICED_END_OF_THREAD(NULL);
}

static void transmit_over_i2c(uint32_t arg)
{
    while(1)
    {
        while(currentRead != currentWrite)
        {
            wiced_i2c_transfer(&wm8533_control_port, buffer[currentRead], 1);
            wiced_packet_delete(packets[currentRead]);
            free(buffer[currentRead]);
            currentRead++;
            if(currentRead > 255)
            {
                currentRead = 0;
            }
        }
    }

    WICED_END_OF_THREAD(NULL);
}

---

**Input/output listing**

Audio file to play and quality sound
Other related material

- Registration to website is mandatory to get the login an user will have access to all the WICED API via SDK.
- QT: Mobile App development requires QT GUI to explore
- Python: Python Installation and executions
- 802.11ac: IEEE