A Hybrid Radio implementing RadioDNS

Summary

I plan to design and develop a web-enabled FM radio that implements new standards, enhancing listener experience by adding content delivered through IP technology.

Background Information

Taking the standard FM radio and giving it a connection to the Internet can empower the device to provide that extra content listeners want. “What track is currently playing?” and “What show is next?” are some of the typical questions that are asked. A new open technology called ‘RadioDNS’ allows devices to use existing meta-data sent within the FM transmission and discover web services provided by the broadcaster to provide extra information. Example services are ‘RadioEPG’, allowing the device access to the station’s schedule and ‘RadioVIS’, pushing images and text to the device in real-time showing relevant information to the listener.
With this Internet connection, access to a vast number of other online radio services is also possible, giving the listener more choice. Discovery and sharing using online social networks is also now possible, creating favourites, accessing podcasts and other non real-time broadcasts is achievable.

Project Objectives

At the end of the project, I hope to have built a hybrid radio that:-

- receives analogue FM radio,
- decodes RDS meta-data,
- discovers online content through RadioDNS,
- fetches and displays online content to enhance the listening experience,
- discovers online radio stations,
- streams online radio stations,
- provides an elegant user interface,
- is an easy experience for the listener.

I hope to provide feedback on my experiences to the RadioDNS community to help develop the RadioDNS standards.

Proposal Form:

This project aims to develop a web-enabled FM radio, or ‘Hybrid Radio’ that implements new technologies from the RadioDNS platform. It will be mainly software based, embedded on a PIC in the device. Initially I will use meta-data from the FM transmission to find online content and services, including real-time images and text. I will then go on to present these to the listener on a touch user interface. Following that, I will be looking to integrate online radio station discovery and streaming live audio to the listener. A final product will be then designed and developed, aimed to be small, low-powered and portable. I will be required to thoroughly understand the technical specifications of the technologies being implemented, as well as use and develop skills in C programming.

Main Hardware

Audio Output: 2 speakers and headphone jack to provide stereo output, audio amplifier to provide appropriate volume adjustment control.

Visual Display: Full colour display, OLED for power efficiency and quality of display with touch screen. Possible: http://uk.farnell.com/densitron/c0240qgla-t/display-am-oled-2-4/dp/1589472

FM RX: An IC that demodulates the FM signal and extracts the RDS data out. Possible: http://www.sparkfun.com/products/10227

Microcontroller: A PIC powerful enough to drive an OLED, use Ethernet and TCP/IP, stream audio and make requests to RadioDNS. Possible: PIC18F67J60 – see document http://ww1.microchip.com/downloads/en/AppNotes/01128b.pdf

Internet Gateway: Ethernet at least, possibly moving onto wireless connections.

Main Software

- Extract RDS data and create RadioDNS queries
- Service discovery through RadioDNS
- EPG – parse xml files to store and display station schedules
- VIS – STOMP client to receive images and text in real-time
- TAG – Interface with radio station web services for RadioTAG
- Internet Radio stream discovery – through user input or menu system
- User Interface I/O: menu navigation, output radio and online data
- Presets/Favourites system
- Podcasts and Listen Again features.

Detailed Specification

The following practical outcomes will be achieved:-

1. FM RX will be a standalone IC with RDS data out to PIC, this saves time working on an analogue FM receiving circuit.
2. Internet connection will be done via Ethernet.
3. Audio output will be by either 2 speakers or 3.5mm stereo headphone jack, it will be the listener's choice.
4. Audio amplification will be provided as necessary by analogue amplifier circuit.
5. A full-colour (touch) screen will be used, at least 240x320px as graphical display to provide a large and elegant user interface.
6. RadioDNS services will be discovered by DNS queries made through radiodns.org
7. At least the RadioEPG and RadioVIS specifications will be implemented allowing electronic station schedule, real-time images and text to be shown to the listener.

8. Internet radio stations with MPEG streams will be able to be streamed and heard, also implementing RadioDNS, as with analogue. This provides the listener with more choice, making use of the internet connectivity.
9. The final product will be fairly small in size to allow for portability.
10. Power will be from DC mains power adaptor, voltage regulation as required.

The following practical outcomes may be achieved, but are not a necessity:-

1. Wireless internet connectivity by Wi-Fi or 3G.
2. Non real-time broadcasts, or podcasts, may be discovered and files played.

Project Timeline

WK14	WK15	WK16	WK17	WK18
Introduction Lectures; Initial Component Research; RadioDNS query prototyping; Proposal & Specification;	RadioDNS Arduino Protype; Order Components; Background Reading;	PIC to FMRX interface; FM Antennae Design; Remote Control;	PIC to Screen interface;	PIC to Ethernet interface; HTTP requests;
WK19	WK20	WK21	WKE1	WKE2
RadioDNS parsing data; RadioVIS STOMP Server; RadioVis STOMP client; Images on Screen in real-time	Audio Output Circuit; MPEG streaming; Online Station Discovery;	Product Design; PCB Designs; User Interface;	RadioTAG implementation; Other Web services;	Podcasts and non-real-time content discovery;
WKE3	WKE4	WK22	WK23	WK24
Project build;	Project Build; Report Writing;	Testing and Fixing; Report Writing;	Testing and Fixing; Report Writing;	Testing and Fixing; Report Writing;

Component List

SCREEN [ORDERED]
DENISTRON - C0240QGLA-T 2.4” Display
http://uk.farnell.com/densitron/c0240qgla-t/display-am-oled-2-4/dp/1589472
Farnell - Order No: 1589472
£36.80 - **DISCONTINUED TODAY??**
so
SHARP - LQ035Q1DH01 - LCD, TFT, 3.5" QVGA LED TOUCH
http://uk.farnell.com/sharp/lq035q1dh01/lcd-tft-3-5-qvga-led-touch/dp/1685283
Farnell - Order No: 1685283
£56.04
or
KENTEC ELECTRONICS - K350QVG-V2-FF - LCD MODULE, 3.5", 320X240, TOUCH SCREEN
http://uk.farnell.com/kentec-electronics/k350qvg-v2-ff/lcd-module-3-5-320x240-touch-screen/dp/2075134
Farnell - Order No: 2075134
£29.95 (US Stock)

PIC [ORDERED]
PIC18F67J60
http://uk.farnell.com/microchip/pic18f67j60-i-pt/8bit-flash-mcu-18f67j60-tqfp64/dp/1332285
Farnell - Order No: 1332285
£4.51

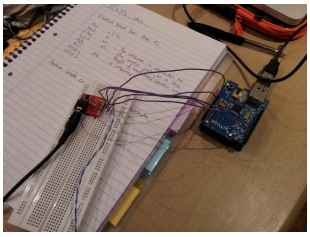
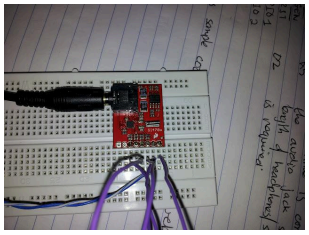
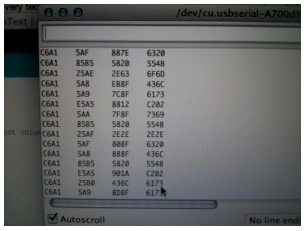
MP3 DECODER [ORDERED]
VS1011 MP3 Decoder Circuit
http://uk.rs-online.com/web/p/multimedia-misc/7165328/
RS - Part No: 716-5328
£18.44

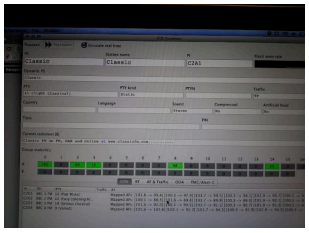
Blog Entries

Up to this week (before Christmas and during the holidays), I have been thinking about my project and have started tying down a detailed specification. I have been looking at various technical specifications that I plan to use RDS, RadioDNS and DNS. I have also started to put together a resource list, a list of the main components that I would like to get as soon as possible.
I have written an outline of the project so I can see any gaps of knowledge, what are my time constraints and laid out some weekly targets for the next semester.
I have agreed a weekly meeting time with Dr Kemp and we are now working on the project proposal form for submission on Monday WK15
I have made a prototype 'RadioDNS querier' to see how the process went from RDS data to getting today's show schedule from a radio station. It is available online at dyl.anjon.es/hybrid/playground

On Thursday I made an Arduino prototype to discover RadioEPG services and to then fetch the days EPG (station's schedule for the day) for given frequency, channel and country identifiers. I found that using the university network to make HTTP requests required to configure the requests to use the university proxy. Instead of this, I used my Laptop as a proxy to make it simpler.
Friday I finalised details of the main components ordered and had a meeting with Dr Kemp and Ed to get my list of main components ordered.
Monday, I held my weekly meeting with Dr Kemp where we went over the plan I formulated in WK14 and details of where I can focus my efforts on reading and information gathering to do to get the very technical content for the report as well as enhance my knowledge. Initial research was then done into RDS and the broadcast chain for FM radio. Further research into the chosen PIC was also done, to get more depth into the architecture of the device.
Tuesday I made some enquiries into University Radio York (where I used to be Chief Engineer) implementing some RadioDNS applications. I may assist in setting up some services that I can also then use to demonstrate my final product. I will also approach Leeds University's student radio station to see if I can also do the same.
I arranged a meeting with John Sheard to discuss how I wanted to use the campus network, we spoke on Wednesday and came to an agreement of having some form of private network that is then bridged to the internet using a PC in the lab or Laptop connected via Wi-Fi.
The Radio receiver chip (on breakout board) arrived Friday and I began experimenting with its functionality using the Arduino and serial interface. After trying out various methods of decoding the whole datastream of RDS, I decided I need not bother for the moment as the only thing actually required to make RadioDNS lokups is the programme identifier and the country code. As we will be using this prototype in Great Britain, I can hard code the country code in for now. The programme identifier (PI) is sent at the beginning of each block of data so is easily extracted. The breakout board comes along with a 3.5mm stereo jack to listen to the audio which is great for debugging. The best signal quality received currently is Classic FM, which is great as it is apart of Global Radio who implement RadioDNS standards including EPG and VIS so this station will be perfect for demonstration.
The antennae pin of the chipset is tied down to the ground of the stereo jack, I found attaching a longer (~2m) of wire (twin-pair) to the ground of the device improves signal quality greatly. The board even provides an RSSI which I will add to the UI of the final device.
I will interface the current prototype with the FM RX chipset this weekend and conduct tests ready for next week.

This weekend I have continued work on adding the FMRX to the Arduino prototype. Below are some images of that work.



The FM receiver circuit detected the RDS data from the transmission. The 3rd image shows how clear the programme identifier (PI) is - "C6A1" within the raw RDS. After decoding in some free software found, RDS Surveyor [rds-surveyor.softpedia.com], you can see in image 4 the identifiers and extra meta-data carried within RDS. I was hoping the the time would be decoded as well, but that does not appear to be included in the sample of data I took. I will have to use an alternative method for knowing the time, probably from the Internet.

Monday, I met with Dr Kemp to update him on my progress and discuss FM antenna design as the first hour in the lab I spent trying to get decent signal. I went on to do some in depth research on various antenna design principles for the FM band. Without having to construct a giant frame and getting specialist cabling in, I found a decent signal at my workbench eventually after shortening my wire considerably to the desired theoretical length. It works well as a flat (on the desk) oval shape.

I managed to converge the arduino prototype from last week with my weekend test code for the receiver and successfully managed to sample the RDS data suitably to make HTTP requests through my proxy web server, retrieving schedule data for the current playing station. I also sadly learnt that Capital FM in Yorkshire does not implement RadioEPG, even though Capital FM London does and it is pretty much the same schedule.

The PIC and MPEG decoder arrived today (Monday), both which will require fitting onto breakout boards for prototyping. The screen was not delivered and I am sure it won't be as the website says it has been discontinued. I will look into getting the second preference device ordered as soon as possible.

Further into Week 16, I worked on adding remote control capabilities to my prototype. I also adjusted the code to be more automatic rather than calling functions by serial commands. On startup, the radio protoype will now seek to a station and start looking for RDS data, then it will make requests to get data. The remote control is based upon a simple web server within the device that listens for commands from a client. I out together a simple HTML page with buttons that on press make HTTP requests to the radio with the command as the URL. You can control the radio's volume and frequency. The radio then returns all the current settings including volume frequency and PI code. After a station is found and the initial RadioDNS request has identified the station, the name of the station is returned as well.

After successfully finding a good antenna design, the next technical aspect I looked at was the analogue audio output. The FMRX chipset comes with the 3.5mm stereo headphone jack attached and I was looking at circuit designs that would allow the internal speakers-to-headphone switchover. A possible solution found was to use impedance matching. When the device is matched at the output the audio is 'sent' down the headphones, otherwise the audio is 'sent' to the internal speakers.

Towards the end of the week, I tidyed up the prototype code to simplify the functions and finialised a remote control protocol. I looked as using the Universal Remote Control API, a way to control all sorts of devices (Hybrid Radio actually given as an example). This white paper goes into a lot of details of service discovery and applications. At the moment I am not going to spend time on implementing this further with my prototype, however if time allows in the final product, I may look at using this so that it conforms to a possible specification.

Objectives for Week 17:

- Get the PIC soldered onto the breakout board
- Get the MP3 decoder soldered onto the breakout board

- Port the arduino prototype to C in MBLAB for execution on the PIC
- Create a small PCB with the ethernet jack socket and circuit protection components for linking to the PIC

Unfortunately, the PCB breakout boards for the PIC and MPEG decoder still have not arrived so I am continuing work using the PIC development board in the mean time. It has a built in OLED display so this will be great for prototyping the screen interface.
I continued refining the code in the Arduino prototype so that the PI code is automatically checked and updated 'in the background' rather than initialising a manual check. This is a lot faster and more efficient, when there is good quality signal, the RDS is decoded within milli-seconds and the PI code recorded in a variable.

Objectives for this week:

- Continue porting Arduino code to MPLAB for PIC
- Investigate services to implement for LSRfm

I met with the station manager of LSRfm on Tuesday to discuss the opportunity for me to help set up RadioDNS services for the student radio station here at Leeds. They have agreed to give me full access to the station to set up the software required etc to get a prototype working as soon as possible. This is a great opportunity for me to see the set up and speak to other broadcasters to see what they are doing to implement RadioDNS from the transmission end.
I have got the OLED display working on the PIC Development board and started interfacing that with the FMRX chipset, I aim to get real-time information up on the display this week as well as look at initial user interface designs.
I successfully added real-time data to the OLED display however, the FPC connector seemed to be dodgy and lines on the display started disappearing. This continued to worsen and now I cannot see anything on the screen although the serial interface seems to be responding normally.