CLASSPERT

MAKING EMBEDDED SYSTEMS – A course by Elecia White

FINAL PROJECT REPORT (draft)

A DIGITAL FM TUNER

Emmanuel PONCELET | May 2022

Table of Contents

Application Description

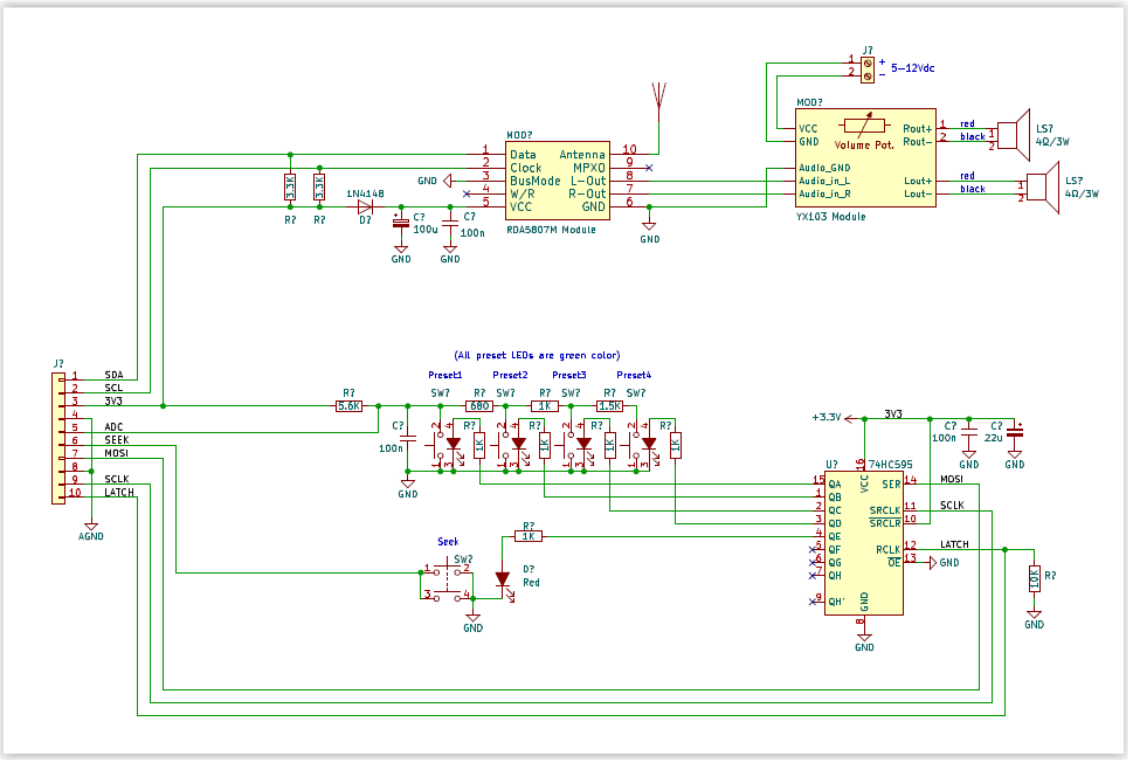
While web and DAB radios are commercially available at reasonable cost these days, I thought it would be nice to build a "kind of" old-fashion receiver based on a Chinese module comprising a digital FM tuner and a headset audio amplifier, both hosted on a printed circuit board not exceeding one square centimeter!

Radio stations use frequency modulation (FM) to modulate the radio-frequency signal and then transmit the data (music or speech). A FM receiver can be adjusted – or tuned - to certain frequencies and receive those signals that will later be converted into audio signals.

Hardware Description

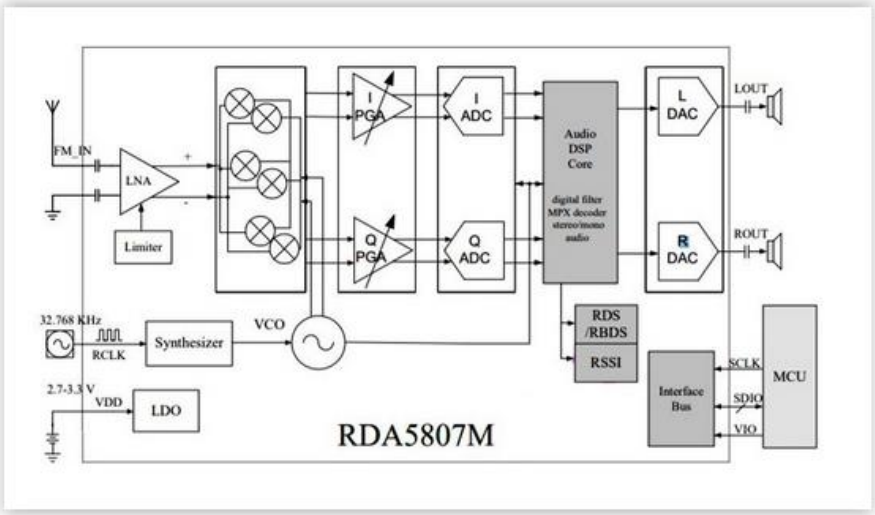
The complete system requires two interconnected boards to operate:

1. The first board is the STM32F429I-DISCOVERY designed by ST Microelectronics which is hosting a 32-bit ARM Cortex M-4 microcontroller and a 2.4" TFT LCD screen. The later will be used for displaying the station frequency and the received signal strength (RSSI). The layout of the board is shown below (*Source: ST Microelectronics*)
2. The second board is a prototype board that I designed. It is hosting a stereo FM tuner module, a 3-W stereo audio amplifier with adjustable output level, two loudspeakers, five push buttons and their associated signaling LEDs and a few discrete components and connectors. Jumper wires are used to pass the signals from one board to the other. The schematics is shown hereafter:

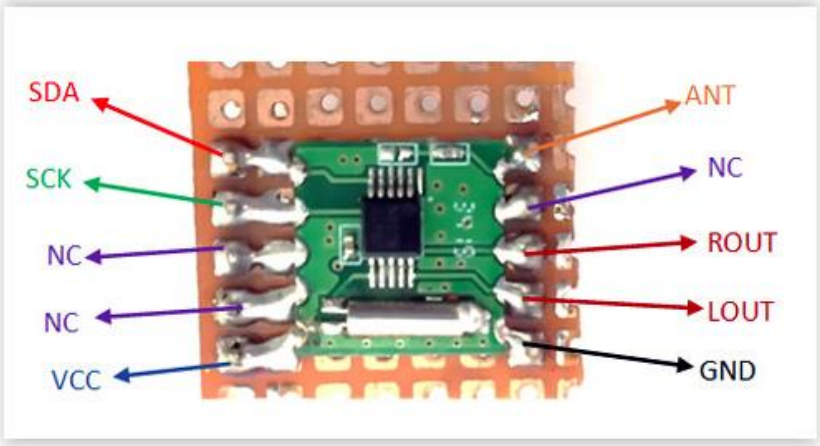


STEREO FM TUNER MODULE

The RDA5807 is a single-chip FM stereo radio tuner module with a fully integrated synthesizer. The module supports the worldwide frequency band of 50 – 115MHz, volume control and mute, programmable de-emphasis (50/75μs), received signal strength indicator and SNR, 32.768KHz crystal oscillator, digital auto gain control, etc.



It has digital low-IF architecture and integrates a low noise amplifier (LNA), which supports the FM broadcast band (50 to 115 MHz), a programmable gain control (PGA), a high-resolution analog-to-digital converter, and high fidelity digital-to-analog converters (DACs). The limiter prevents overloading and limits the number of intermodulation products created by adjacent channels. The PGA amplifies the mixer output signal and then digitized with ADCs. The DSP core manages the channel selection, FM demodulation, stereo MPX decoder, and output audio signals.



(*Source: https://circuitdigest.com/microcontroller-projects/arduino-fm-radio-using-rda5807*)

Accessing and configuring the RDA5807M registers is made by sending commands through the I2C bus.

DISCOVERY KIT STM32F429I-DISC1

This kit is based on a STM32F429ZIT6 microcontroller featuring a maximum clock speed of 180MHz, 2 Mbytes of Flash memory, 256 Kbytes of RAM in an LQFP144 package. It also comprises 64-Mbit SDRAM, six LEDs which two are for the user, a 3-axis MEMS gyroscope, two push-buttons (user and reset), I/O extension headers, one user USB, one USB-OTA, among other common peripherals such as ADCs, DACs, timers, USARTs, I2C, I2S, SPI, DMA, etc.

AUDIO AMPLIFIER MODULE

A 3-W stereo amplifier XY103 module is required to amplify the audio signals coming out of the FM tuner module. The output volume of both loudspeakers can be adjusted with a potentiometer located on the module. There is no L-R balance adjustment.

USER INPUTS

Five momentary, normally open, push-buttons are available to the user. A first series of four are used to memorize a station during the scan process or to select the station to listen. A button is active whenever its corresponding LED is lit. These memory buttons are polled regularly by the firmware.

There is a little trick here about how these stations buttons are used: rather than tying each of them to GPIO pin, a push shorts a resistor at some position of a resistive divider. Therefore, the voltage read on the measurement resistance is fed to the ADC. This configuration is commonly used with microcontrollers having very few I/Os available. In this case, it helps respecting the project's requirement about peripherals.

The fifth push-button is used for seeking stations emitting in the area. The seek mode only goes up and loops when the FM tuner highest frequency is reached. The seek button is connected to GPIO pin which responds to a falling edge interrupt.

The LEDs illumination is made with a binary pattern send in series to a 74HC595 shift register via the SPI bus.

LCD TFT DISPLAY

In addition to the LED attached to each button, the user can read additional information on the Discovery Kit LCD display. This information is the station frequency and the RSSI. Messages such as "empty preset memory", "erase preset" or "preset stored" will also appear // to do...

Software Description

GENERAL DESCRIPTION

The behavior of the software is pretty straightforward. Once the user switches the FM tuner on, three scenarios may arise:

1. No preset is stored → In this case, the first thing to do is to seek for FM stations and store them into the microcontroller Flash memory (EEPROM emulation).

2. Some presets are already stored → In this case, the new preset is either stored at a free location or in an existing location (in this case, the previous preset is erased). Note that during this step, stored preset LEDs are blinking helping the user to locate the free ones.

3. All presets are stored → In this case, the user can either erase all presets or erase one or more of them by applying a long push on the corresponding button(s).

SOFTWARE PARTS

The software was originally written for a Microchip dsPIC33F and has been deployed on that microcontroller. Not all the features are fully functional nor developed (e.g., the LCD

display code). My intention is to port it to the STM32F429 and add features (LCD display, CLI) and modify existing parts to better fit with new ideas or hardware.

Here is a list of the different software parts:

Name	Purpose
adc.c	To measure the voltage divider value to determine which preset button has been pressed
but_leds.c	To manage the LEDs by sending patterns through SPI bus in a shift-register driving them
flash.c	To emulate an EEPROM for storing the station presets in the microcontroller Flash memory
lcd.c	Graphic functions to write or draw on the LCD display
timers.c	Software timers // not sure to need them
main.c	Where everything starts. The state machine lies here too.
tuner.c	To manage the FM digital chip (configuration registers)
cli.c	To debug through a serial port using a command line interface // done as homework #5 but needs to be tailored to my final project, of course

RE-USED CODE AND LICENSES

// to do

Diagrams of architecture

- Hardware block diagram
- Software block diagram
- Hierarchy of control

Build instructions

HARDWARE

// to do

SOFTWARE

- How to build the system (including the toolchain(s) // to do
- How you debugged and tested the system // to do
- How you powered it // to do

Future

To make this project ready for production, two actions are necessary:

1. Design a dedicated printed circuit for hosting a microcontroller, a display, a tuner, and user input device(s). The Discovery kit microcontroller is way too powerful and has too many features and I/Os for this application. Therefore, another choice has to be made. Also, rather than using push-buttons for seeking, storing and selecting stations, a touchscreen display or a rotary encoder with axial push-button could be used. The tuner chip could also be replaced by a multi-band digital receiver supporting analog and digital radio standards.

2. The software will need to be upgraded according to the new hardware design. Moreover, depending on the new tuner chip complexity, it might be advisable to totally re-write this code part. Then extensive tests would be very useful for finding weird situations e.g., what happens if the user pushes two presets simultaneously?

Grading

// to do