Objective

The purpose of this project is to build an FM receiver with good reception, capable of tuning to frequencies in the FM broadcast range (88 – 108 MHz) and outputting the received audio signal through a speaker.

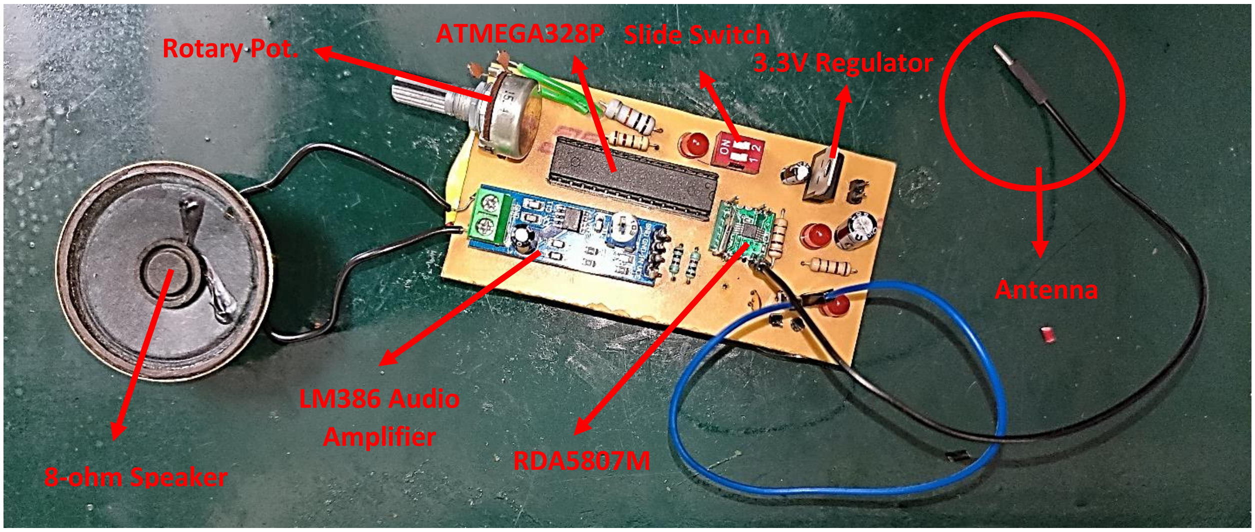
Components Used

Sr. No.	Component	Qty	Description
1.	**RDA5807M**	**1**	**FM Stereo Radio Module**
2.	**ATMEGA328P (PDIP)**	**1**	**8-bit microcontroller to tune RDA5807**
3.	**LM386 Module**	**1**	**Audio Amplifier**
4.	**Phenolic PCB board**	**1**	**Base for soldering the components**
5.	**8 ohm 0.5W Speaker**	**1**	**Audio output**
6.	**3.7V 1000mAh Li-Ion Battery**	**1**	**Power supply**
7.	**TP4056 3.7V Li-Ion Charger Module**	**1**	**Power supply charger**
8.	**3.3V Linear Voltage Regulator**	**1**	**To drop down from 3.7V to 3.3V**
9.	**100k Rotary Potentiometer**	**1**	**Knob to select desired frequency**
10.	2-way slide switch	1	Power on and Reset switches
11.	Red LEDs	3	Indicate power on and reset
12.	10k resistor	1	Reset pull-up resistor
13.	1k resistor	2	Stereo to mono conversion
14.	220 ohm resistor	3	LED current-limiting resistors
15.	100uF capacitor	1	Decoupling capacitor
16.	10uF capacitor	1	Decoupling capacitor
17.	0.1uF capacitor	1	Decoupling capacitor
18.	Pin headers (2.54mm pitch)	14	For antenna and power supply interface
19.	28 pin IC socket	1	IC base for ATMEGA328P
20.	Single core connecting wires	Multiple	For PCB bottom layer connections
21.	Jumper wires	Multiple	For antenna and other connecting wires
22.	**Soldering equipment**	**1**	**Soldering iron, solder wire, flux, wick**
23.	**ATMEGA328P ISP**	**1**	**To burn the program to microcontroller**

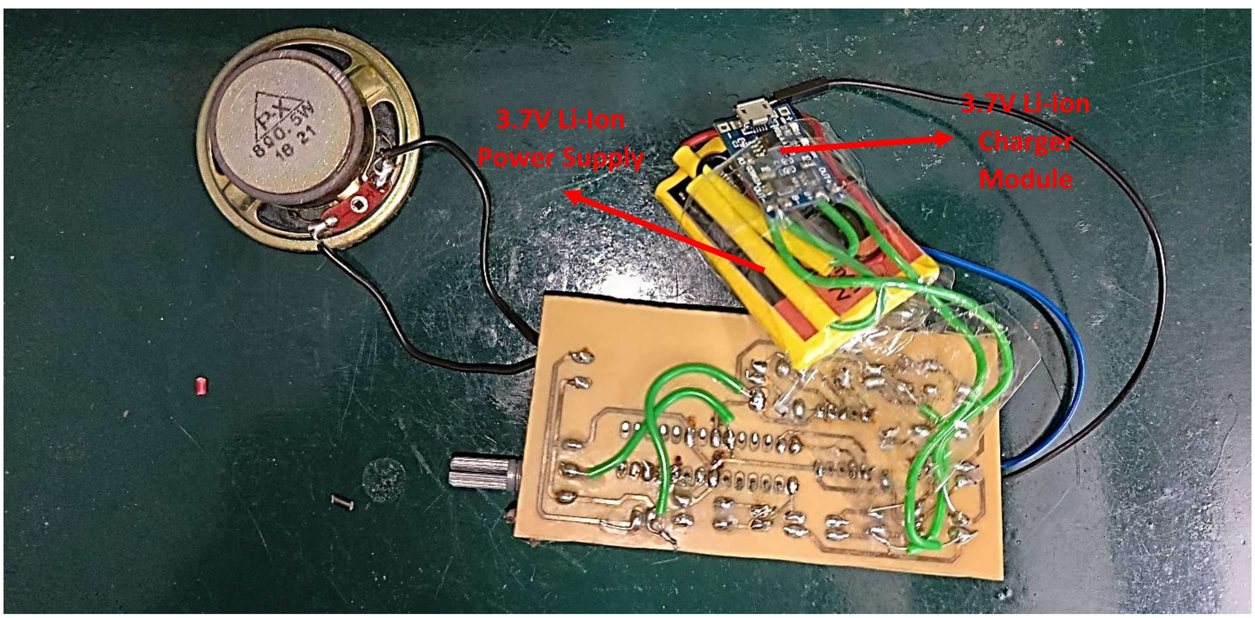
Software Used

Autodesk EAGLE (for designing the schematic and PCB layout), and Arduino IDE (for programming the ATMEGA328P)

Physical Prototype

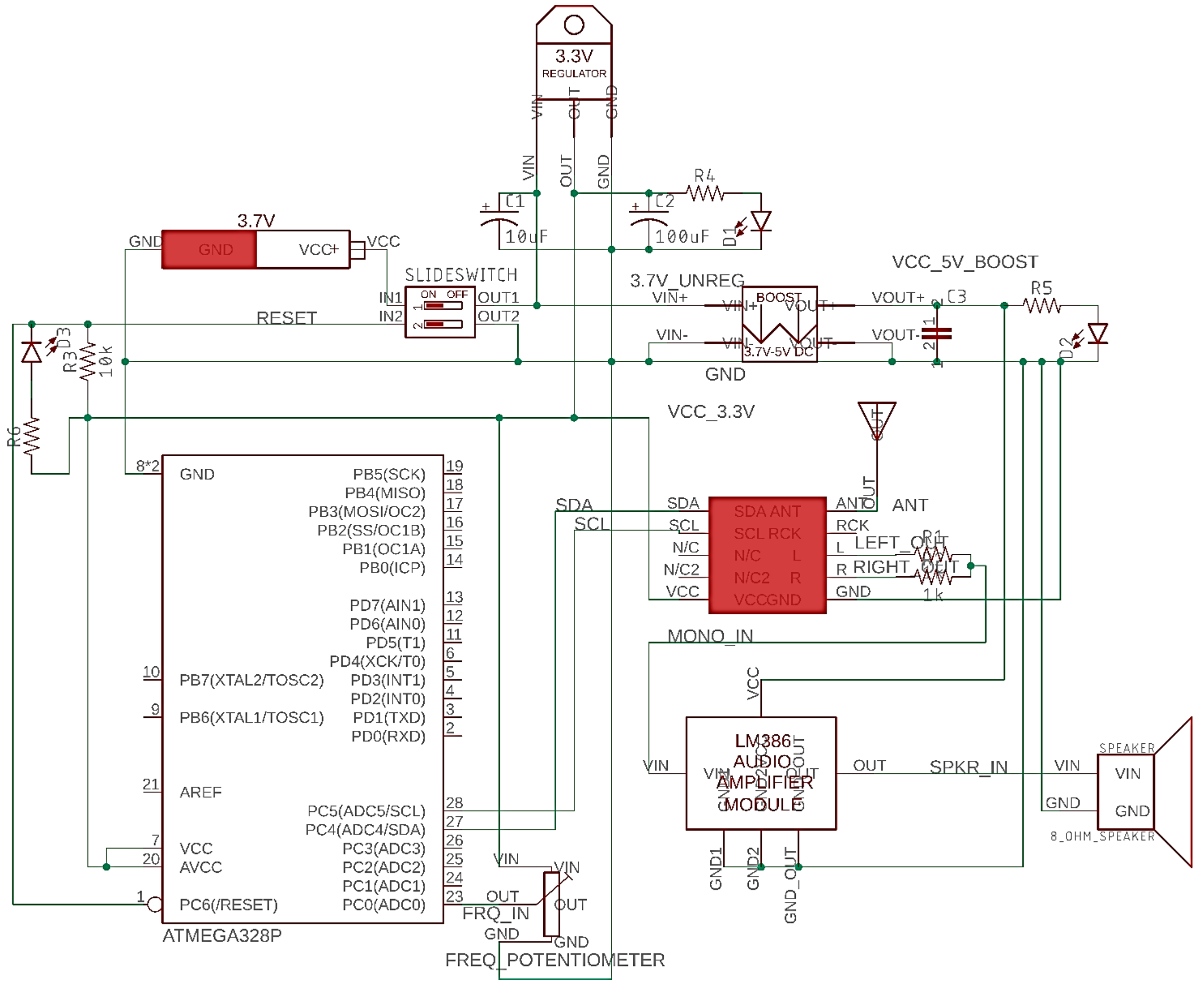


Top View



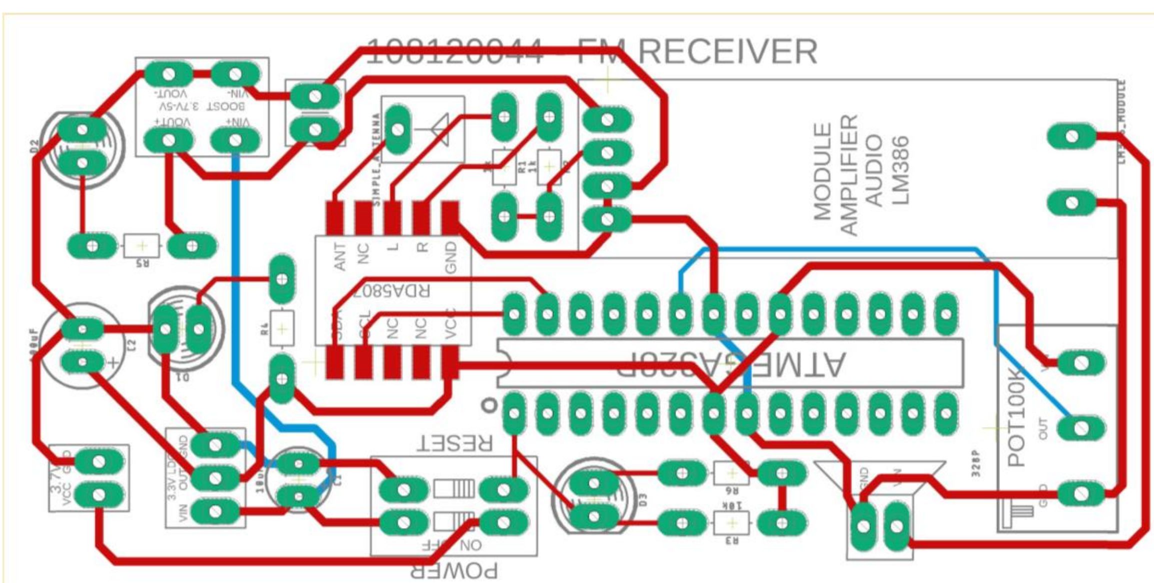
Bottom View

Schematic Diagram



3.3V
REGULATOR
VIN
OUT
GND
C1
10uF
C2
100uF
R4
D1
3.7V
GND
VCC+
VCC
SLIDESWITCH
ON OFF
IN1
IN2
OUT1
OUT2
RESET
D3
R3
10k
R6
3.7V_UNREG
VIN+
VIN-
BOOST
VOUT+
VOUT-
3.7V-5V DC
VCC_5V_BOOST
C3
R5
D2
VCC_3.3V
ANT
8*2
PB5(SCK)
PB4(MISO)
PB3(MOSI/OC2)
PB2(SS/OC1B)
PB1(OC1A)
PB0(ICP)
PD7(AIN1)
PD6(AIN0)
PD5(T1)
PD4(XCK/T0)
PB7(XTAL2/TOSC2)
PD3(INT1)
PD2(INT0)
PB6(XTAL1/TOSC1)
PD1(TXD)
PD0(RXD)
AREF
PC5(ADC5/SCL)
PC4(ADC4/SDA)
AVCC
PC3(ADC3)
PC2(ADC2)
PC1(ADC1)
PC6(/RESET)
PC0(ADC0)
ATMEGA328P
SDA
SCL
N/C
N/C2
RCK
L
R
LEFT_OUT
RIGHT_OUT
R1
1k
MONO_IN
LM386
AUDIO
AMPLIFIER
MODULE
GND1
GND2
GND_OUT
SPKR_IN
SPEAKER
8_OHM_SPEAKER
FRQ_IN
FREQ_POTENTIOMETER

PCB Layout



108120044 - FM RECEIVER
MODULE
AMPLIFIER
AUDIO
LM386
RDA5807
ANT
NC
L
R
GND
SDA
SCL
VCC
SIMPLE_ANTENNA
ATMEGA328P
RESET
POWER
ON
OFF
POT100K
BOOST 3.7V-5V
VOUT-
VIN-
VOUT+
VIN+
3.3V LDO
3.7V
VCC
GND
R5
R4
R6
R3
10K
R1
1k
D2
D3
C1
C2

Working of Circuit

1. **FM Reception:**

   The working of the entire prototype is centered around the **RDA5807M IC**. This is the IC responsible for the complete FM reception. The working of this IC and its internal functioning are mentioned later in the report. An **antenna** (quarter wavelength wire of about 75cm) is interfaced with the RDA5807M IC and is responsible for interception of the FM signals.

2. **Frequency Tuning:**

   The **ATMEGA328P** is an 8-bit microcontroller used widely in the DIY community for various electronics projects and has a vast programming library. The ATMEGA328P is interfaced with the RDA5807M using the “Two-Wire-Interface”, which is a digital serial communication protocol. Through this interface, the microcontroller programs the RDA5807M by manipulating its register contents. This allows the microcontroller to specify the required frequency digitally, and the RDA5807M takes care of tuning to that frequency.

3. **User Input:**

   The user can control the frequency being tuned to, with the help of a **rotary potentiometer** acting as a knob. The output of rotary potentiometer is fed to the ADC (Analog to Digital Converter) input pin of ATMEGA328P. Based on the position of the knob, a set of 7 predefined frequencies are tuned to in a sequential manner. The user also has a set of **two slide switches** at their disposal – the power and reset switch. By toggling the power switch, the user can turn on and off the FM receiver circuit. A red LED indicates whether the circuit is turned on or off. By toggling the reset switch, the user can reset the ATMEGA328P microcontroller, starting the program afresh. A red LED indicates whether reset is currently active.

4. **Audio Output:**

   Once reception of FM signals starts, the RDA5807M IC demodulates the signal and retrieves the analog baseband message signal. This is further processed to retrieve the stereo audio outputs. The left and right audio channels are outputted by the RDA5807M, but they have low signal strength. These two signals are first converted to mono using an averaging circuit comprising two 1k resistors. The output is then fed to the **LM386 Audio Amplifier**. The audio amplifier amplifies the signal, and outputs it through the **8-ohm Speaker**.

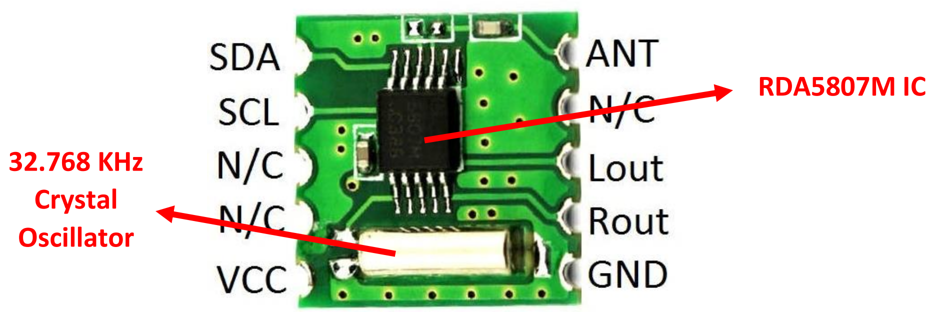
5. **Power Supply:**

The power supply circuit consists of a central supply to the entire board, from a **3.7V Li-Ion Rechargeable Battery**. A **3.3V Linear Voltage Regulator** is used to drop the voltage to 3.3V, which is the supply for RDA5807M and ATMEGA328P. Meanwhile, this same 3.3V supply can also be used as power supply for the LM386 Audio Amplifier. However, an interface to supply external voltages for the amplifier is provided. Additionally, the rechargeable battery is permanently interfaced with a **3.7V Li-Ion Charger Module**. Decoupling capacitors are provided wherever possible, to minimize power supply noise.

Working of Circuit Modules

RDA5807M FM Receiver

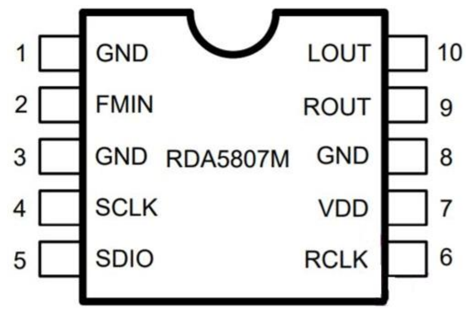
RDA5807M Module Pin Diagram



1. SDA – Serial Data line for TWI/I2C communication
2. SCL – Serial Clock line for TWI/I2C communication
3. N/C – No Connection
4. VCC – Positive rail of power supply
5. ANT – Antenna
6. Lout – Left channel of stereo audio output
7. Rout – Right channel of stereo audio output
8. GND – Negative rail of power supply.

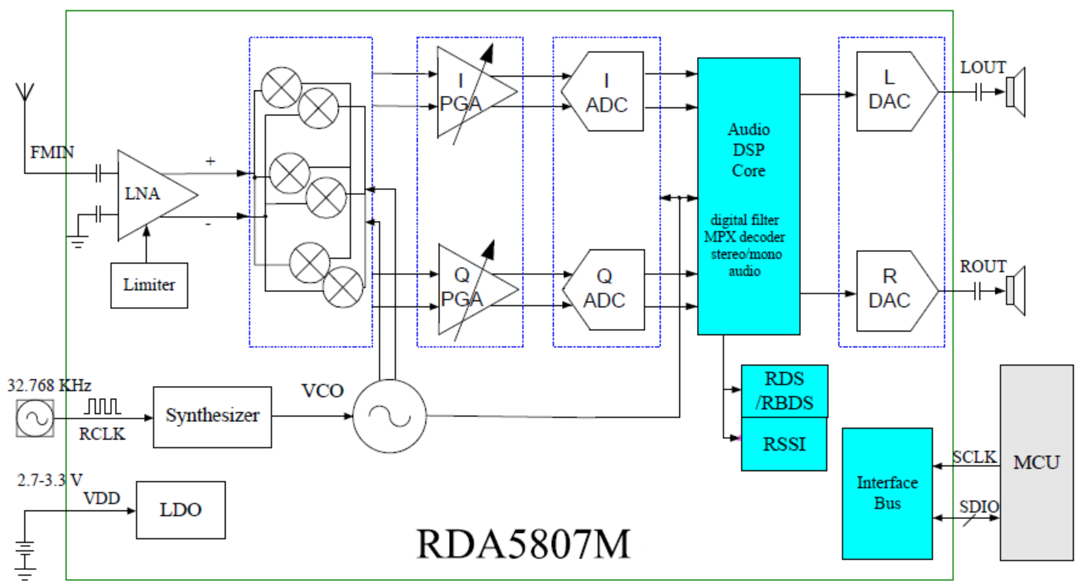
In the circuit, the SDA and SCL pins are interfaced with the ATMEGA328P's SDA and SCL pins. The Lout and Rout stereo outputs are averaged into mono and interfaced with the LM386 Audio Amplifier. The ANT pin is connected to the quarter-wavelength antenna wire.

RDA5807M IC Pin Diagram



1. GND – Negative rail of power supply
2. FMIN – FM Input
3. SCLK – Serial Clock line for TWI/I2C communication
4. SDIO – Serial Data line for TWI/I2C communication
5. LOUT – Left channel of stereo audio output
6. ROUT – Right channel of stereo audio output
7. VDD – Positive rail of power supply
8. RCLK – 32.768 KHz crystal oscillator and reference clock input

RDA5807M IC Working



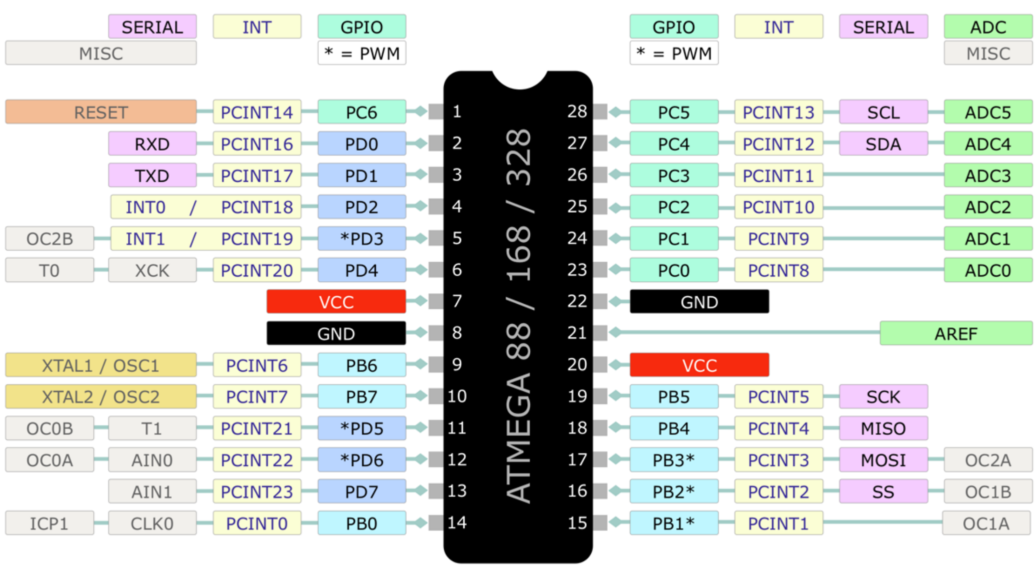
RDA5807M Internal Block Diagram

The receiver has a **Digital Low-IF (Intermediate Frequency) Architecture** that avoids the difficulties associated with direct conversion, while delivering lower solution cost and reduced complexity. It integrates a **Low Noise Amplifier (LNA)** supporting the FM broadcast band (50 to 115MHz), a **Multi-Phase Image-Reject Mixer Array**, a **Programmable Gain Amplifier (PGA)**, high resolution **Analog-to-Digital Converters (ADCs)**, an **Audio-DSP**, and high-fidelity **Digital-to-Analog Converters (DACs)**. The following gives a gist of the functioning of the IC:

- The **Limiter** prevents overloading and limits the amount of intermodulation products created by strong adjacent channels.
- The **Frequency Synthesizer** generates the local oscillator signal which is given to the multi-phase mixer array. The synthesizer reference clock is 32.768 KHz. The synthesizer output frequency ranges from 50MHz to 115MHz, depending on the desired frequency to be tuned.
- The **Multi-Phase Mixer Array** down-converts the LNA output RF signal to low-IF. It also has image-reject and harmonic-tones-reject functions.
- The **Programmable Gain Amplifier** amplifies the mixer output IF signal, which is then digitized with **ADCs**.
- The **Audio-DSP core** performs the channel selection and FM demodulation. It has a **Stereo MPX (Multiplex) Decoder**, which decodes the message into its left and right audio channels. The MPX decoder can autonomously switch from stereo to mono to limit the output noise. The left and right digital audio data are then fed to DACs.
- The **DACs** convert digital audio signal to analog and can also change the volume of the audio. The DACs have a low-pass characteristic (low-pass reconstruction filter), with a -3dB frequency of about 30 KHz.

ATMEGA328P Microcontroller

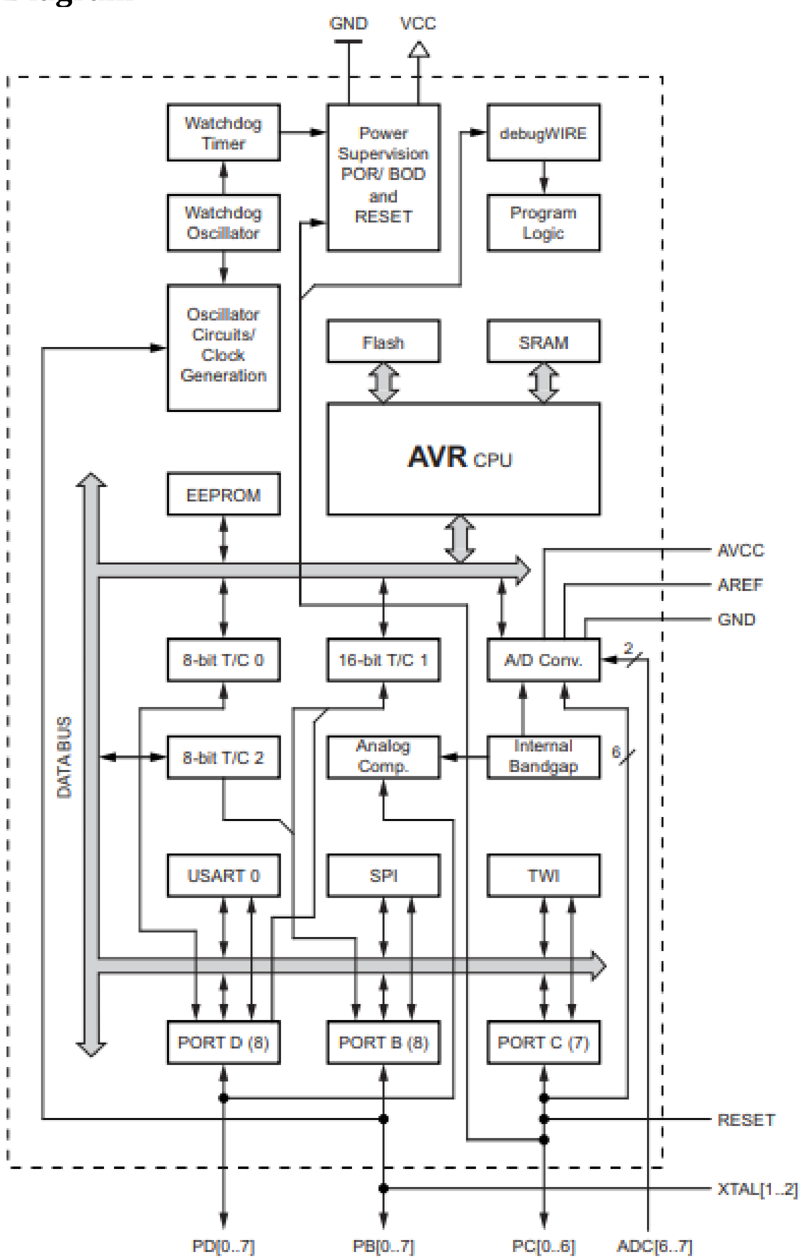
IC Pin Diagram



1. SDA (pin 27) – Serial Data line for TWI/I2C communication
2. SCL (pin 28) – Serial Clock line for TWI/I2C communication
3. ADC0 (pin 23) – Analog input pin, internally connected to a 10-bit ADC
4. RESET (pin 1) – To reset the microcontroller, and begin the program afresh

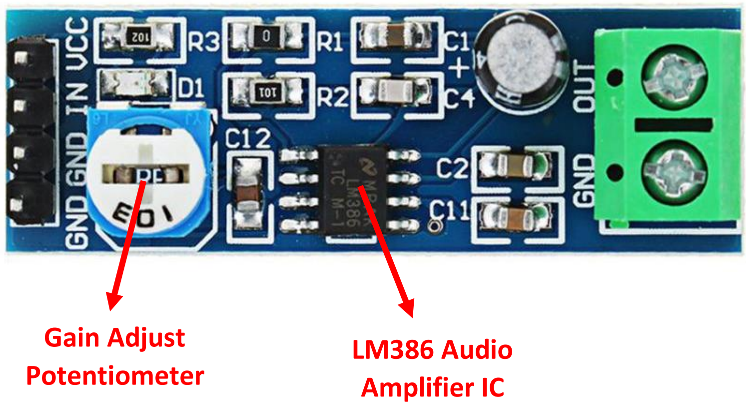
In the circuit, the SDA and SCL pins are interfaced with RDA5807M's SDA and SCL pins. The ADC0 pin is connected to the rotary potentiometer's output. The RESET pin is active low, hence is pulled high through a 10k resistor. The RESET pin is also interfaced with the reset switch, which when toggled, pulls the RESET pin low.

Internal Block Diagram



LM386 Audio Amplifier

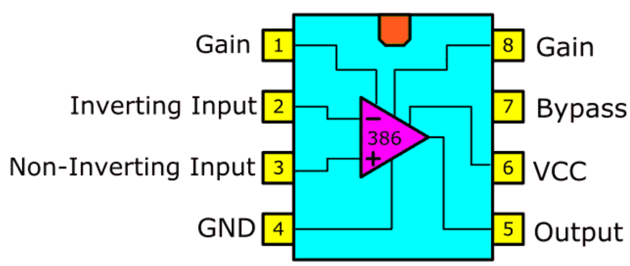
LM386 Audio Amplifier Module Pin Diagram



1. VCC – Positive rail of power supply
2. IN – Mono audio input
3. OUT – Amplified mono audio output
4. GND – Negative rail of power supply

In the circuit, the IN pin is connected to the mono audio output, the averaged stereo output from RDA5807M. The OUT pin is connected to the input of the speaker.

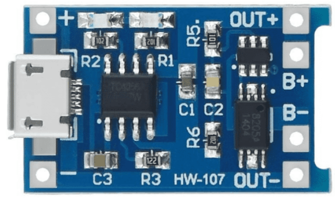
LM386 Audio Amplifier IC Pin Diagram



1. Gain – Used with pin 8, to adjust the gain with the help of a capacitor and resistor
2. Inverting Input – (-) terminal of differential input
3. Non-Inverting Input – (+) terminal of differential input
4. GND – Negative rail of power supply
5. Output – Amplified output
6. VCC – Positive rail of power supply
7. Bypass – To connect decoupling capacitor
8. Gain - Used with pin 1, to adjust the gain with the help of a capacitor and resistor

TP4056 3.7V Li-Ion Charger

Module Pin Diagram

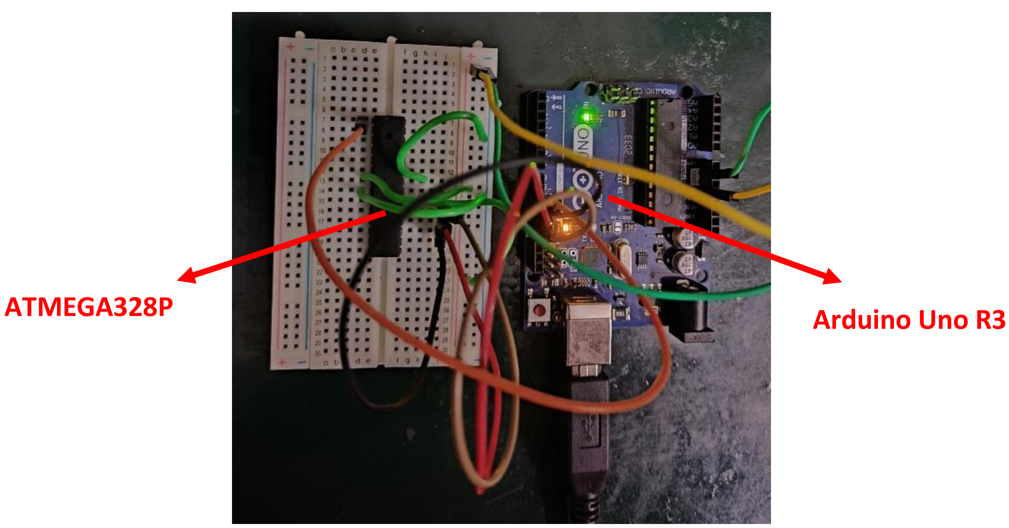


1. + – Positive terminal of 5V charging battery
2. - – Negative terminal of 5V charging battery
3. OUT+ – Positive supply voltage from 3.7V battery
4. OUT- – GND output from 3.7V battery
5. B+ – Positive terminal of 3.7V battery
6. B- – Negative terminal of 3.7V battery

In the circuit, the (+) and (-) terminals are left floating. Rather, the micro-USB port is used for charging, which would be connected to a 5V wall adapter (no load must be connected during charge). The B+ and B- are connected to positive and negative terminals of the 3.7V battery. The OUT+ and OUT- are used for supplying the entire circuit.

In-System-Programming of ATMEGA328P:

To program the ATMEGA328P, an Arduino Uno R3 module is used, which is itself a popular microcontroller board based on the ATMEGA328P, used for many DIY electronics projects, comprising a vast programming library. Arduino IDE software is used for this task.



First, the Arduino is programmed with readily available code, which converts it into an ISP for the ATMEGA328P. Then, the ATMEGA328P is interfaced with the Arduino, and the bootloader is burnt into it. Then, the ATMEGA328P is programmed with the following code (SPI protocol is used for programming), which is responsible for TWI/I2C communication with RDA5807M and receiving input from the rotary potentiometer:

rda_freq_tune_atmega

```
#include <RDA5807.h>
#define MAX_DELAY_RDS 40   // 40ms - polling method

const int pot = A0;
int freq_checkpoints[7] = {9100, 9350, 9500, 9750, 10050, 10210, 10640};
int voltage_checkpoints[8] = {0, 146, 292, 438, 584, 730, 876, 1024};
long rds_elapsed = millis();
RDA5807 rx;

void setup()
{
    Wire.begin();
    rx.setup();
    rx.setVolume(15);
    delay(500);
    rx.setRDS(true); // Enables SDR
    detectTuning(pot);
}

void detectTuning(int pot_pin)
{
  int curr_index = getVoltageIndex(pot_pin);
  rx.setFrequency(freq_checkpoints[curr_index-1]);
  delay(200);
  while(1)
  {
    int index = getVoltageIndex(pot_pin);
    if(index!=curr_index)
    {
      curr_index = index;
      rx.setFrequency(freq_checkpoints[curr_index-1]);
      delay(200);
    }
  }
}

int getVoltageIndex(int pot_pin)
{
  int pot_read = analogRead(pot_pin);
  int index = 1;
  for(int i=1; i<8; i++)
  {
    if(voltage_checkpoints[i-1]<=pot_read && pot_read<voltage_checkpoints[i])
    {
      index=i;
      break;
    }
  }
  return index;
}
```

Result and Conclusion:

The FM receiver was successfully implemented, and all circuit components work flawlessly. The working of the FM receiver is demonstrated in the following link:

https://drive.google.com/file/d/14Dg3HLB6xJ2poVqxr5RXImXUq9b5pR8t/view?usp=share_link

As can be seen in the demonstration, the FM receiver is capable of tuning to FM stations and receiving the broadcasts with high fidelity. It can be observed that the quality of the received signal is impeccable. The working of the FM receiver is thus verified.

GitHub repository: https://github.com/Hridhay08/fm-receiver