

```
/* RDA5807M based FM Radio
```

```
HecaWorld unLTD
```

```
info@hecaworld.com
```

```
March 25th, 2022.
```

```
Digital channel/frequency tuning with Channel storage in EEPROM with EEPROM.write and  
EEPROM.read
```

```
*/
```

```
#include <Wire.h>
```

```
#include <EEPROM.h>
```

```
//Declare channel up and channel down buttons
```

```
int menu = A1; //Pin number 3
```

```
int menuValue = 0;
```

```
int volumeNumber = 4;
```

```
bool toggleMenu = true;
```

```
int Vin = 5;
```

```
float Vout = 0;
```

```
float R1 = 30000;
```

```
float R2 = 0;
```

```
float buffer = 0;
```

```
//int channelDown = 4;
```

```
//Declare variable to hold station/channel number
```

```
int stationNumber;
```

```
int chan_Up = HIGH; // Channel Up
```

```
int chan_Down = HIGH; // Channel Down
```

```
// Variables to store the time is in unsigned longs because the time, measured in
```

```
// milliseconds, will quickly become too big for numbers that can be stored in an int variable.  
unsigned long lastDebounceTime = 0; // The last time the input pullup pin was toggled  
unsigned long debounceDelay = 1500; // The debounce time; Select appropriate number to avoid  
flickers
```

```
void setup() {  
  // make the pushbutton pins inputs:  
  // pinMode(menu, INPUT_PULLUP);  
  // pinMode(channelDown, INPUT_PULLUP);  
  Serial.begin(9600);  
  /*Read the stored station number from the EEPROM.  
  If none stored, make is channel 183 elase,  
  the station number is what has been read from the EEPROM*/  
  stationNumber = EEPROM.read(0);  
  Serial.print("Station Number: ");  
  Serial.println(stationNumber);  
  if (stationNumber == NULL){  
    stationNumber = 183;  
  }else{  
    stationNumber = stationNumber;  
  }  
  // stationNumber = 183;  
  tuneStation ();  
}
```

```
void loop(){
```

```
// read the state of each switch into separate variables:
```

```
menuValue = analogRead(menu);
```

```
if(menuValue > 200){
```

```
Serial.print("menuValue Data: ");
```

```
Serial.println(menuValue);
```

```
buffer = menuValue * Vin;
```

```
Vout = (buffer)/1024.0;
```

```
buffer = (Vin/Vout) - 1;
```

```
R2= R1 * buffer;
```

```
Serial.print("R2 Value is: ");
```

```
Serial.println(R2);
```

```
// delay(1000);
```

```
// }
```

```
// Serial.println(menuValue);
```

```
// delay(2000);
```

```
// int chan_Down = digitalRead(channelDown);
```

```
//=====
```

```
if (R2 > 20000 && R2 < 50000) {
```

```
if (toggleMenu == true){
```

```
stationNumber = stationNumber + 2;
```

```
if (stationNumber > 211){
```

```
stationNumber = 9;
```

```
}
```

```
Serial.println("Channel: ");
```

```
Serial.println(stationNumber);
```

```
tuneStation ();
```

```
} else {  
    volumeNumber ++;  
    if(volumeNumber > 15){  
        volumeNumber = 15;  
    }  
    Serial.print("Volume is: ");  
    Serial.println(volumeNumber);  
}  
}  
//}  
//=====
```

```
if (R2 > 50000 && R2 < 70000) {  
    if (toggleMenu == true){  
        stationNumber = stationNumber - 2;  
        if (stationNumber < 9){  
            stationNumber = 211;  
        }  
        Serial.println("Channel: ");  
        Serial.println(stationNumber);  
        tuneStation ();  
    }  
    else {  
        volumeNumber --;  
        if(volumeNumber < 0){  
            volumeNumber = 0;  
        }  
        Serial.print("Volume: ");  
        Serial.println(volumeNumber);  
    }  
}
```

```

    }
}

if (R2 > 80000) {
    Serial.print("Toggle Menu State: ");
    Serial.println(toggleMenu);
    if(toggleMenu){
        Serial.println("Set Volume");
    }
    else
    {
        Serial.println("Set Channel");
    }
    toggleMenu = !toggleMenu;
}

//=====
delay(2000);
}
}

void tuneStation (){
    uint16_t channel = stationNumber;
    EEPROM.write(0, stationNumber);
    // channel = (<desired freq in MHz> - 87.0) / 0.1
    #define RDA5807M_ADDRESS 0b0010000 // 0x10

```

```

#define BOOT_CONFIG_LEN 12
#define TUNE_CONFIG_LEN 4

uint8_t boot_config[] = {
    0b11000000, 0b00000011, 0b00000000, 0b00000000,
    0b00001010, 0b00000000, 0b10001000, 0b00001111,
    0b00000000, 0b00000000, 0b01000010, 0b00000010,
};

uint8_t tune_config[] = {
    /* register 0x02 */
    0b11000000,
    0b00000001,
    (channel >> 2),
    ((channel & 0b11) << 6) | 0b00010000
};

//=====

Wire.begin(); // join i2c bus (address optional for master)

// Sending boot configuration (device should reset)...;
Wire.beginTransmission(RDA5807M_ADDRESS);
// Write the boot configuration bytes to the RDA5807M
Wire.write(boot_config, BOOT_CONFIG_LEN);
Wire.endTransmission(); // stop transmitting
Wire.beginTransmission(RDA5807M_ADDRESS);
// Write the tuning configuration bytes to the RDA5807M
Wire.write(tune_config, TUNE_CONFIG_LEN);

```

```
Wire.endTransmission(); // stop transmitting
//=====
// Serial.println(stationNumber);
Serial.print("You're tuned to: ");
Serial.print(stationNumber / 10.0 + 87);
Serial.println(" FM Radio Station");
}
```