

Ricevitore FM 88-108 MHz con decodifica RDS ed interfaccia PC con Arduino

di Enrico Guindani IZ2NXF



Tempo fa, un amico mi mostrò un ricevitore portatile acquistato in Cina. Era dotato di display, tastiera numerica e un generoso altoparlante. La ricezione delle broadcast FM era decisamente buona e l'ascolto gradevole. Dopo averci smanettato un po', decise di aprirlo per mostrarmi il suo interno. Con mia grande sorpresa, vidi un circuito stampato semi-deserto: solo pochi componenti SMD sparsi qua e là. Si distinguevano chiaramente tre minuscoli chip. L'amico, indicandoli, mi disse: "Questo è il ricevitore, questo è il controller per tastiera e display, e questo è l'amplificatore audio". Tutto qua!

A distanza di qualche anno, ho voluto realizzare un ricevitore FM con decodifica RDS ed interfaccia per PC ricalcando la stessa filosofia.

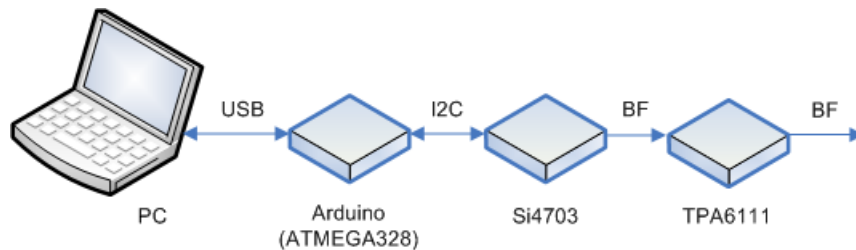
La cosa si è resa possibile grazie alla vasta disponibilità di sofisticati radio-chip che incorporano tutte le funzionalità necessarie in un unico, singolo, piccolissimo componente SMD.

Alcuni di questi integrati si possono reperire sui classici canali di vendita on-line, già montati su pratiche *Evaluation Board* molto comode per chi non vuole cimentarsi in ardite, per non dire impossibili, saldature SMD.

Aspetto interessante è il fatto che questi chip scambiano i loro dati tramite bus I2C.

Inter-Integrated-Circuit (IIC o I2C) è un sistema di comunicazione seriale utilizzato tra circuiti integrati e che necessita di due soli fili chiamati SDA (Serial Data) e SCL (Serial Clock) oltre, naturalmente, a Vcc e GND. Tramite I2C si impostano i parametri della radio (frequenza, volume, ecc...) e si leggono i valori relativi alla ricezione (intensità segnale, RDS, ecc...).

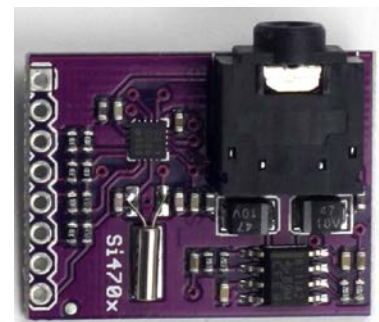
Considerando che **Arduino** è in grado di gestire il protocollo I2C grazie alla libreria WIRE, va da sè che si presta ottimamente per interfacciarsi con questo tipo di componenti.



La “Radio board”

Il chip ricevitore (nella foto, sopra il quarzo) è un **Si4703** prodotto dalla Silicon Labs. E’ montato su una *Board* pronta all’uso, già dotata dei necessari componenti discreti e pure di un piccolo amplificatore audio (**TPA6111** - nella foto, in basso a destra) con tanto di connettore d’uscita jack stereo.

Il **Si4703** riceve la classica banda FM 88 – 108 MHz, ma può essere impostato anche per scendere fino a 76 MHz per sintonizzare la banda FM-OIRT in uso nei paesi dell’est europeo.



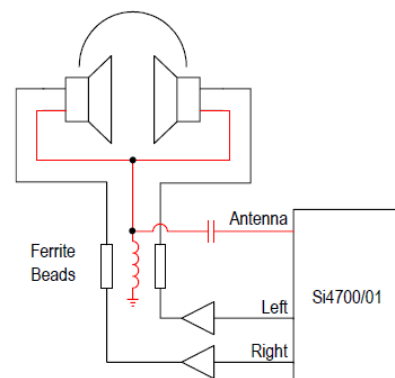
L’uscita audio del **TPA6111** è di 150 mW e si prestano bene per pilotare una cuffia. Per ottenere un ascolto più corposo, in altoparlante, occorre un’ulteriore amplificazione.

La qualità audio, unita ad un’ottima ricezione, è davvero notevole. Per questo motivo, in questo progetto non ho integrato un altoparlante per l’ascolto diretto, ma ho lasciato disponibile un’uscita jack stereo per l’ascolto in cuffia o il collegamento ad un amplificatore “serio” come quello dell’Hi-Fi di casa o una “docking station” audio.

E l’antenna?

L’antenna è realizzata tramite la calza schermante del cavo delle cuffie che si vanno a collegare; modalità, questa, utilizzata anche nei telefoni cellulari dotati di Radio FM.

Visti i fortissimi segnali FM che abbiamo in Italia, tale antenna è più che sufficiente per una buona ricezione, ma, in ogni caso, ho reso disponibile verso l’esterno un connettore BNC a cui ho applicato uno stilo telescopico.



Va ricordato che il **Si4703** è indirizzato al mercato consumer e non a quello professionale o radioamatoriale. Di conseguenza, non consente nessun intervento a livello RF. Quindi, nessun controllo, nessuna filtratura, niente di niente, sul segnale in ingresso.

RDS

Un'importante caratteristica del **Si4703** è la capacità di estrarre dal segnale l'RDS (Radio Data System). La mia attenzione si è quindi concentrata sulla decodifica e sulla visualizzazione di tre importanti dati:



Station Name: composto da otto caratteri, indica il nome della stazione emittente.

RDS Text: 64 caratteri che riportano dati relativi al programma in ascolto (p.es. titolo e autore della musica che si sta ascoltando).

PI-code: codice che identifica la stazione.

Spendiamo due parole riguardo quest'ultimo dato, sconosciuto ai più: il *PI-code*.

Ad ogni emittente FM è assegnato un codice di quattro caratteri esadecimale che identifica univocamente la stazione, indipendentemente dalla frequenza. Per esempio, a RAI-1 è assegnato il codice 5201. Qualunque sia la zona d'Italia e qualunque sia la frequenza, quando riceviamo RAI-1, il *PI-code* codificato nell'RDS sarà 5201. Questo dato è molto utile per identificare con sicurezza la stazione, specie in condizioni di ricezione non ottimali, dato che il *PI-code* risulta più semplice da decodificare rispetto allo *Station Name*.

Va anche detto che il suo primo carattere identifica la nazione trasmittente. Nello specifico, all'Italia è stato assegnato il '5', alla Svizzera il '4', alla Francia la 'F'.

Tastiera e manopola

L'utilizzo del ricevitore avviene tramite una tastiera numerica a membrana e una manopola con annesso encoder rotativo.

Tramite la tastiera si digita direttamente la frequenza da sintonizzare, oppure si seleziona una delle memorie preimpostate nel software.

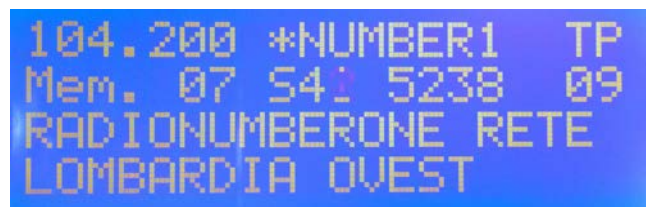
La manopola, invece, in base alla modalità impostata con i tasti '*' e '#', regola il volume, cambia la frequenza, scorre le memorie, o gestisce la ricerca delle stazioni.

Il display

Per la visualizzazione mi sono affidato ad un luminoso LCD a 4 righe per 20 colonne. Il display mostra l'impostazione corrente del ricevitore, più i dati ricevuti dall'RDS.

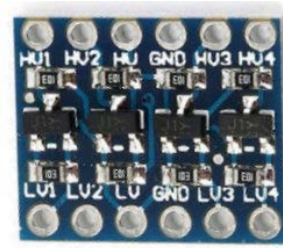
Le prime due righe visualizzano la frequenza corrente, lo *Station Name*, il volume, la memoria, l'intensità del segnale, l'indicazione stereo/mono e il *PI-code*. La terza e la quarta riga mostrano l'*RDS Text*. Siccome tale testo può contenere fino a 64 caratteri, nel caso ci siano da visualizzare più di 40 caratteri, il testo viene visualizzato in due tempi, alternativamente.

Per risparmiare pin su **Arduino**, visto che la tastiera da sola ne occupa sette, ho utilizzato un display con adattatore I2C. In questo modo, i fili che collegano il display sono soltanto quattro invece di dieci o dodici.



I2C LCD Adapter

Però, mentre il display funziona a 5V, il ricevitore **Si4703** funziona a 3.3V. Sono quindi dovuto ricorrere ad un **adattatore di livello** per connettere sullo stesso bus I2C due dispositivi che lavorano a tensioni diverse, la qual cosa ha complicato sensibilmente la realizzazione, come si può vedere osservando lo schema.



I2C Level Converter

Sempre guardando lo schema, salta all'occhio una connessione "strana" o, quantomeno, inusuale: l'alimentazione dell'encoder ottico. Essendo già occupato il relativo pin di **Arduino**, ho prelevato i necessari 5V da uno dei due connettori ICSP.

Il firmware

Internet è davvero zeppo di esempi a riguardo, con schemi e codice C++; basta cercare "Arduino Si4703" e poi scegliere quello che meglio si adatta alle nostre esigenze. Segnalo comunque il sito di *Sparkfun*, ottimo riferimento per attingere a tutte le informazioni hardware e software che servono:

<https://www.sparkfun.com/products/retired/10344>

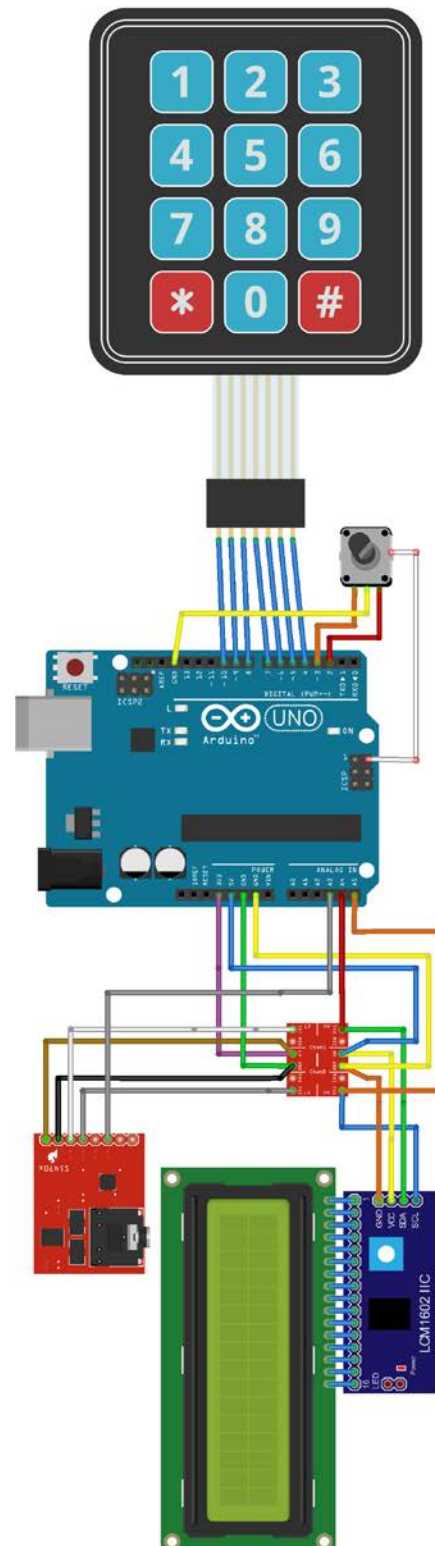
<https://www.sparkfun.com/products/12938>

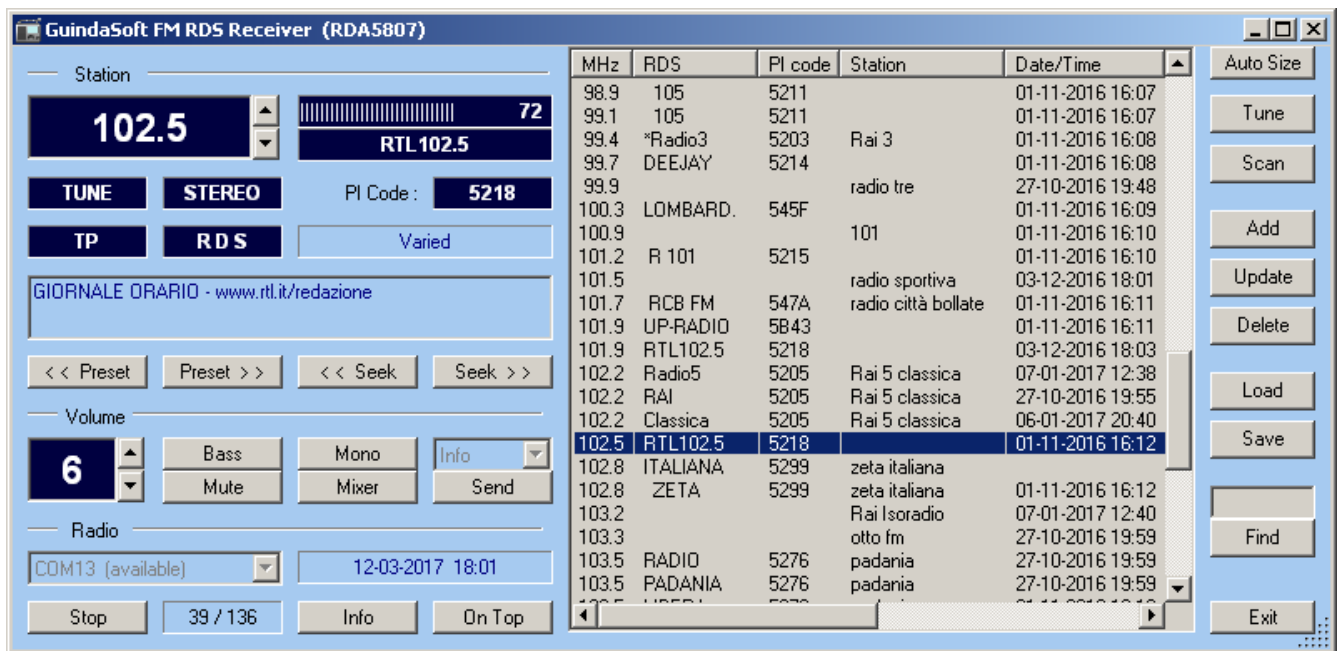
Il software

Di ricevitori FM commerciali ne esistono a bizzeffe. Non tutti, però, visualizzano l'RDS. E di quelli che lo fanno, pochissimi visualizzano anche *RDS Text* e *PI-code*.

A parte questa considerazione, ritengo che il vero valore aggiunto di questo progetto sia il software scritto per PC con cui si può controllare completamente il ricevitore via USB. Tale software, oltre a gestire i parametri della radio, può effettuare autonomamente la scansione della banda FM, leggere i dati RDS delle stazioni e salvare tutto in un file. In questo modo, oltre a poter sintonizzare la stazione desiderata con un semplice click di mouse, si può mantenere un elenco costantemente aggiornato delle emittenti che si ricevono e verificare personalmente quanto sia ancora "animato" il mondo del broadcasting FM.

Per finire, uno speciale ringraziamento va a Giuseppe IK2UTZ per la preziosa collaborazione nella realizzazione del pannello frontale.





Control Software per PC

Quello appena descritto è un po' un punto di arrivo, un'evoluzione di alcuni progetti precedenti, più semplici, che mi hanno consentito di capire come funzionano questi radio-chip, piccoli gioielli di miniaturizzazione integrata.

Prima, per esempio, ne ho realizzato uno senza interfaccia esterna, destinato ad essere utilizzato esclusivamente in abbinamento al PC con il software di cui sopra.

Collegato all'ingresso AUX della scheda audio, si ascolta la radio direttamente dalle casse del computer.

Grazie all'impiego di un **Arduino Nano** (più piccolo del classico **Arduino Uno**), ho potuto utilizzare come contenitore un compatto parallelepipedo di plastica trasparente di dimensioni 100 x 45 x 45 mm.



Buon ascolto a tutti!

Enrico Guindani, IZ2NXF

iz2nxf@gsi.net

www.gsi.net/iz2nxf