

A “Physical Notification” System

Rohit Kishore

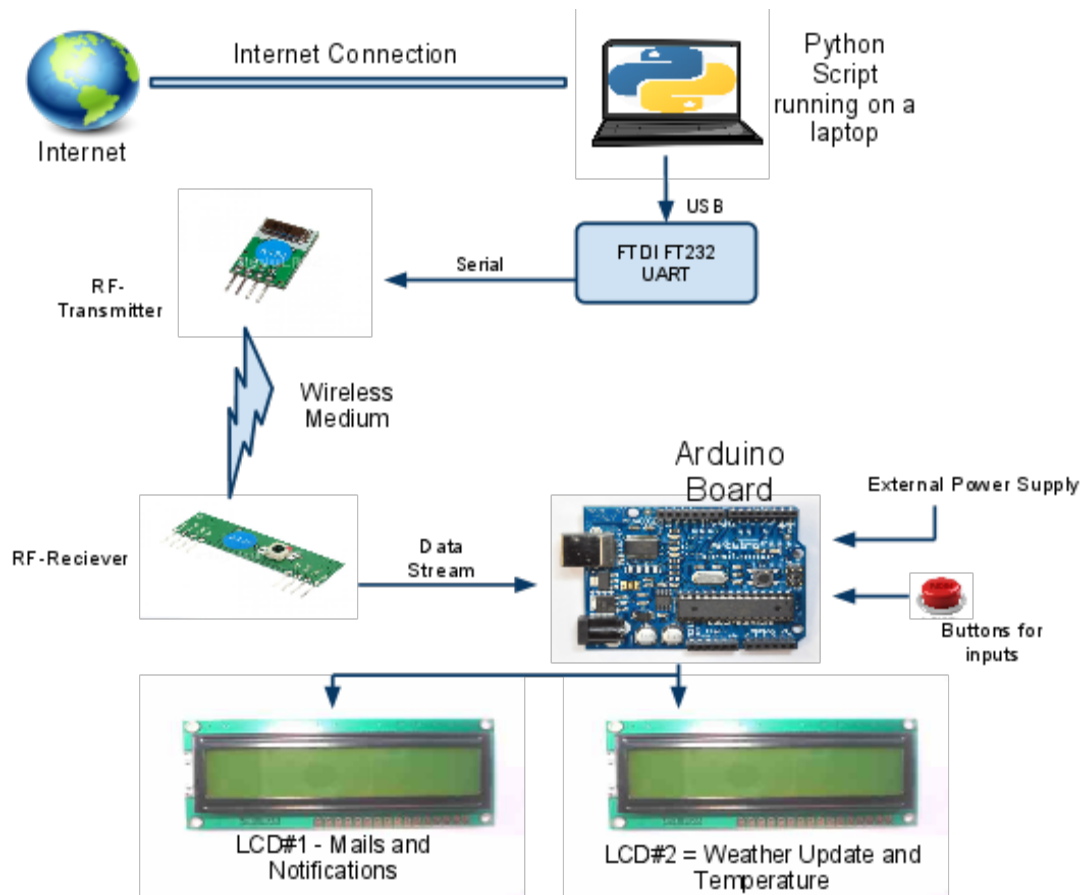
2008045

Synopsis :-

The physical notification system acts as an extension to our personal computers for a variety of things, all working at once place. Be it new mails, any new notification on the facebook, weather update or be it room temperature. Using two external LCDs, the system displays all the above mentioned information at one place and thus saves us from manually checking for the same. The information is extracted using a python script which runs over a computer and parses atom and RSS feeds to obtain the required data using the feedparser library in python.

Architecture cum Hardware Involvements :-

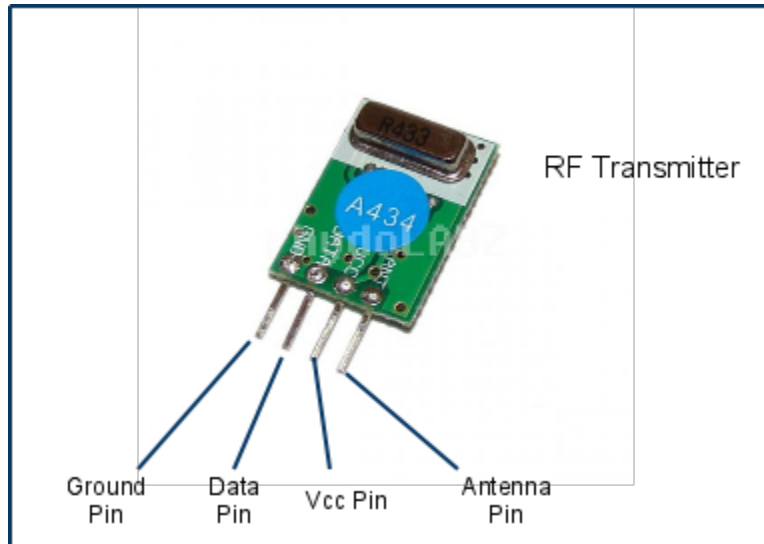
Overall Architecture :-



Subsystems/Hardwares :-

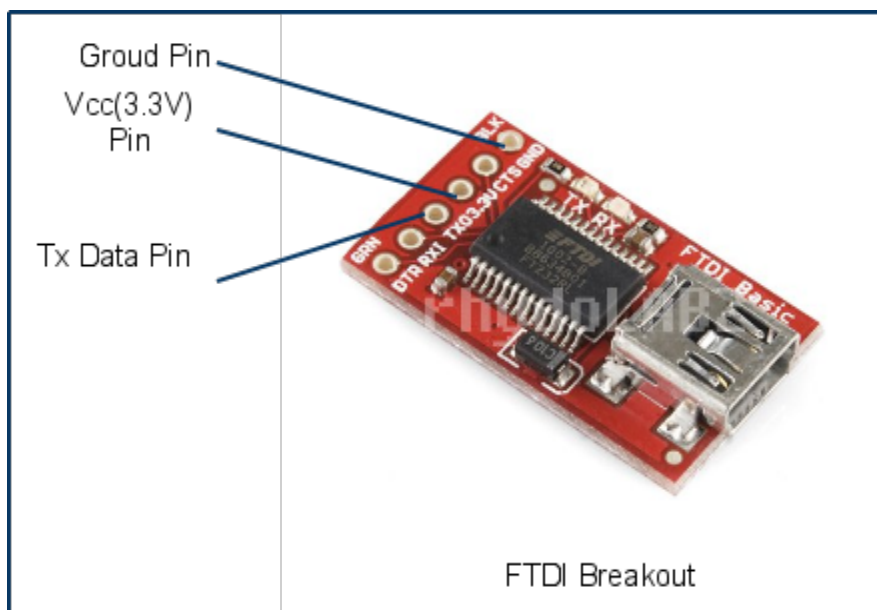
1. RF-Transmitter

RF-transmitters work at 434Mhz frequency(the ones brought in by me) with a respectable range of about 30 feet, when a proper length antenna is used. RF-Transmitter was connected with the PC using a FT232RL UART. This RF-transmitter along with the RF-receiver was purchased from Lajpat Rai Market for Rs. 350 a pair.



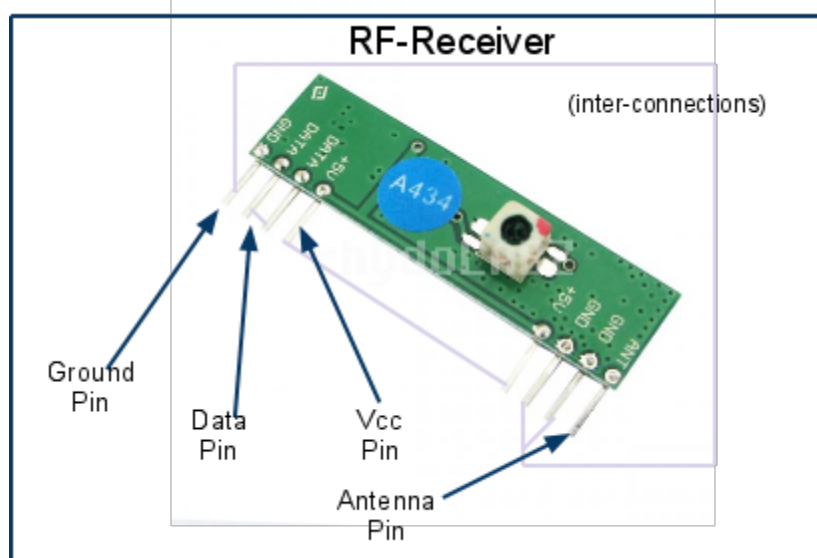
2. USB to Serial Communication for using RF-transmitter directly from the computer, using an RF232RL UART

To get a working UART, i had to order a FTDI breakout(3.3 V) from rhydolabz.com. This piece of hardware costs about Rs. 1200 inclusive of postal charges and other taxes. This breakout board connects with the PC using a USB-A to mini B connector, which was already available as cell phone's data cable.



3. **RF-Receiver**

Information broadcasted from the RF-Transmitter is read using the RF-Receiver at the Microcontroller side using Serial ports at the board.



4. **LCD and Buttons**

LCD and Buttons are connected the usual way, just some attention is required when connecting the buttons to the pins 2 and 3 for functioning as interrupt creators, so that correct wires are attached to the designated pin. To connect two LCDs together, one just needs to get a different wire for pin6, i.e. the enable signal, rest all connections stay the same in both LCDs.

Software :-

In terms of external softwares, only Arduino IDE is used to do the programming of the microcontroller board. Version used was 0018 stable for linux. Python support to use the script was present initially. Only feedparser library was required to be installed, which can be found [here](#). Linux had inbuilt support to run the FTDI breakout board, so no new installation was required. But, if using Windows environment, support patches can be obtained from [here](#).

Learnings :-

There were a lot of learning from this project, namely :-

1. Using RF tx/rx devices,
2. Understanding the working of USB to Serial convertors(UART in this case),
3. Using feedparser libraries in python,
4. Understanding how RSS and ATOM feeds work,
5. Understanding the real application of capacitors(used in button console).
6. Relationship between baud rate and data efficiency.

Challenges Faced :-

Getting the Wireless Subsystem working was the most critical challenge that was faced by me. Problem was that the data reception was very inaccurate and thus after looking into the properties of the RF-transmitters/receivers, it was found that the radios required to be first be in synch with each others' capacities. This was done by repeatedly sending high-low signal for about hundred times before doing actual data transmissions. Also, these cheap radios had a lot of interference from surrounding objects like Screens and speakers and thus i had to turn my speakers off before using it.

References :-

- To connect multiple LCDs :
<http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1265969050>
- To read multiple buttons using one pin :
<https://tronixstuff.wordpress.com/2011/01/11/tutorial-using-analog-input-for-multiple-buttons/>
- To get RF radios working, :
http://www.youtube.com/watch?v=TMebT1OKp_Y
- To parse feeds to get information from the internet :
<http://www.python-forum.org/pythonforum/viewtopic.php?f=2&t=22398>