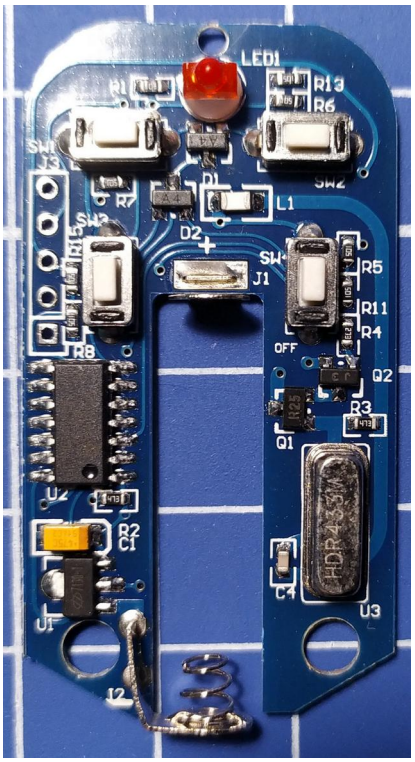
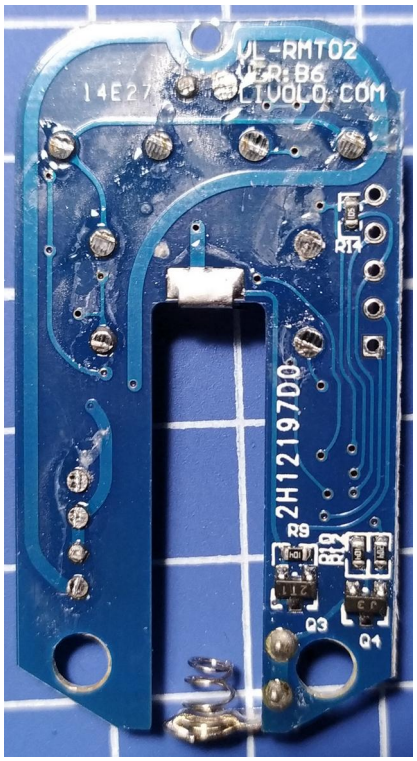
How to reverse engineer a simple RF chinese electronics

by Wojciech Cybowski (github.com/wcyb)

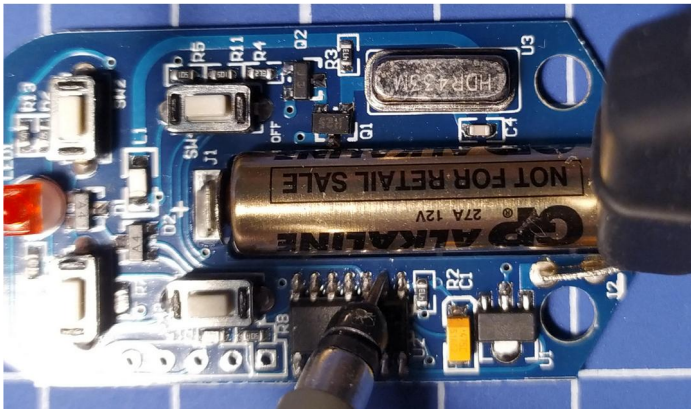
In this short tutorial it will be shown how one can extract data from RF remote used to control light switch. After doing this, it will be possible to create own solution for controlling such light switch, instead of using said remote. It may be useful in smart home scenarios, where we would like to give control of lights to our central hub.

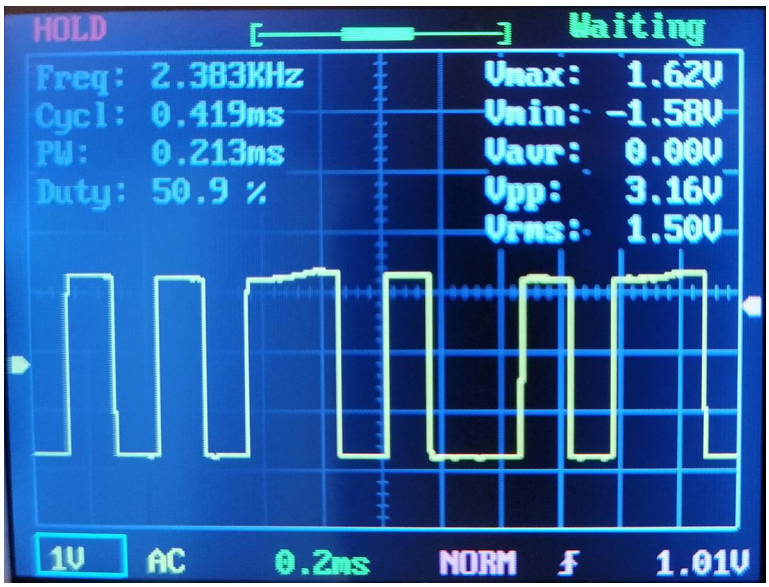




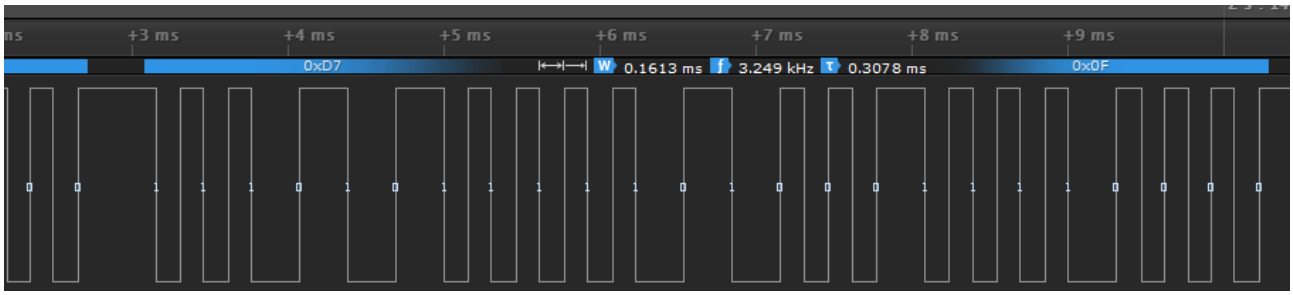
After visual inspection, one can see that this is a very standard and simple RF solution. The 433MHz crystal (U3) is a hint that tells us frequency on which this remote works. If someone would need an exact value of transmit frequency, a SDR could be used to determine this very fast. Also seeing how it is constructed, it looks like there will be no frequency hopping or even strict control of transmitting frequency, so one can use the simplest and cheapest RF transmitter module from AliExpress, that is build to use 433 MHz frequency. It will work.

Before we start doing anything, make sure that battery is removed. Then we can start checking to which pin the antenna is connected. Using continuity mode on a multi meter, we can determine that a resistor (R4) controlling the switching transistor (Q2) is connected to the second pin of microcontroller. To confirm that this pin is used for sending data, we can connect an oscilloscope or logic analyzer to check that.

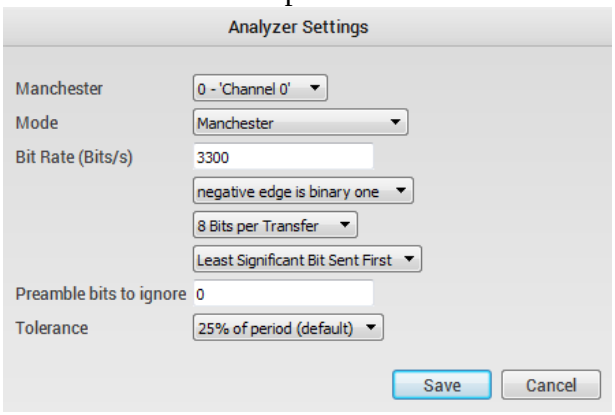




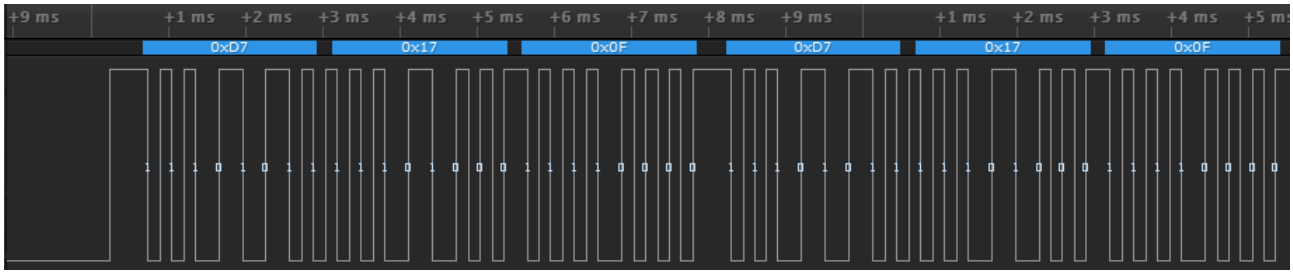
After we are sure that we found right pin, we can start collecting data that is sent by this remote. To do this, we will use Saleae Logic 8. You need to connect ground of the remote and Logic 8 together, and use one of the inputs in Logic 8 to capture data. When both connections will be made you can start capturing data. Capturing rate can be set to about 2 MS/s, and time of capture to about 30 s. It will be enough. Then press some button on the remote, wait for LED on it to stop blinking and stop the capture in Saleae. Now we have the data, but we need to decode it. Before we do this, it is necessary to check bit rate of data we just captured. This can be done by moving the mouse cursor between two shortest pulses on a captured trace, and reading value “f” that will be shown above said trace. You don’t have to check the whole trace to find two shortest pulses, just do it by eye, it will be enough. In my case value of “f” was about 3,25 kHz, so we can round it up to 3,3 kHz.



Because it is a simple remote, we can be pretty sure that Manchester encoding was used, so we clicking on “+” next to “Analyzers”, then select “Show more analyzers” and we selecting “Manchester”. All settings can be left at default values, except bit rate that we determined to be 3300, and also channel on which we have our captured data.



Now we have decoded data, and as one can see, this is just 4 bytes repeated over 1 second. Only thing that stands out is longer pulse on the beginning of captured data. It may be used to wake up the receiver, or maybe it’s just a bug in software of the transmitter.



Having everything that is needed to create our own solution, we can do it very easily using Arduino and cheap RF transmitter module – each press of a button on the remote sends 4 bytes of data encoded using Manchester encoding on 433 MHz frequency. Tested and ready to use code can be found here: https://github.com/wcyb/livolo_vl-rmt02