**Infrared Room Control (with Samsung IR protocol)**

[with 29 comments](http://rusticengineering.com/2011/02/09/infrared-room-control-with-samsung-ir-protocol/#comments)

I had ear some time ago that laziness sometimes is the best thing for inspiration. That week I discover that’s true.  Some nights,  I  lay on my bed to watch some thing on TV and doing some electronic research, before sleep. And, I don’t like to put off the lights (effficient ilumination is the best friend for a good reading) . Then, obviously, before sleep, I need to stand up to turn lights off. At my side, on bed, always resides a great remote control TV, with a set of unused buttons, telling “*use us to turn the lights off*“.



My remote

That’s a project that I was retarding because other, but now I decide to initiate and finish it.

**The Basic Idea**

The first step is recognize what my remote control is send to TV. It’s a Samsung remote control, with TV, VCR, DVD and STB specific and shared functions. My basic idea is: use some of this unused buttons to turn the lighst ON and OFF. But, why not dimmer the lamp? Why not doing some other useful thing? (Yes, thats because some projects growing up…)

So, the basic idea now is:

* Make a device that recognize when a command is addressed to it self or not;
* Recognize different commands;
* Save usuful states (dimmer regulation);

**Samsung Protocol**

The Internet has a lot of information about RC5, NEC and Sony IR protocols, but not the same for Samsung. After some search at Google, I found this [usefull page](http://www.techdesign.be/projects/011/011_waves.htm) with some information about Samsung IR protocol. Resuming Samsung protocol:

* 37.9KHz carrier wave (ON state is a burst of carrier with some duration, OFF is absense of it);
* 1 Start bit (4.5ms ON, and 4.5ms OFF);
* 32 data bits stream (data + address?);
* bit “1” (590μs ON, 1690μs OFF) *(thanks to* ***Islam qabel****,  for the more precise bit duration)*;
* bit “0” (590μs ON, 590μs OFF);
* 1 Stop bit (590μs ON, 590μs OFF);

I think that more information only with a test.

**Test Circuit**

The test circuit is very simple, consisting on a IR receiver (with filter, carrier demodulation and output). My only device available at home is a TSOP2236. Well, let’s go see what hapens.

TSOP2236 is a dedicated IR receiver with PIN diode and preamplifier, assembled on lead frame. The epoxy
package is designed as IR filter, to improve sensibility. The demodulated output signal can directly be
decoded by a microprocessor, but it’s logical reversed:

* ON state (carrier presence): TSOP output LOW;
* OFF state (carrier absence): TSOP output HIGH;

To make measures and confirm my suspects, I had use my [Logic Analyzer](http://www.saleae.com/logic/) from Saleae on the follow circuit:



Test Circuit

That’s the results for press button “1”:



Received data after pressing “button 1″ on remote



Signal stream



Start bit identification (9ms total)



Bit “1” representation



Bit “0” representation

The process to gathering all this information is very easy with Logic Analyzer. The measured times differs a little bit, but this isn’t a problem. I count the transitions after the START bit and there’s 32 bits in.

Look again at the sequence of high and down states. Remember that we have a reversed version of the original IR wave from remote, because TSOP OUT signal is inverted. But actually it’s really doesn’t matter. What I need is: pressing “1” differs from pressing “2” or other buttons on remote. I don’t wanna to create a compatible device.

**Decoding the data stream**

To decode the data stream I used a PIC18F2520 MCU. I already has working on 16bits and 32bits MCU, but this project is much simple to require a great MCU. My design goals are:

* Use internal clock (8Mhz, Instructiom Time TCy = 4/8MHz = 500ns);
* Use Timer0 as counter with 1us increment (Timer0 prescaler to 1:2);
* Use External Interrup 0 (INT0) to handle incoming IR waves and get the building IR code (final version);

That’s the circuit:



Decoding Test Circuit

**Coding**

The code (for Microchip C18 compiler) consist in a set of functions to detect when the signal change it’s state from DOWN to UP or UP to DOWN. See the sequence (basic):



Basic Algorithm

Note that for data bit identification, only HIGH time is important. It’s the difference between logic “1” and “0”, so you don’t need to be very precise in that differentiation. The red right program branch, needs to run 32 times, for all 32 bits.

**I get…**

The test was a success! After all, I have a 32bits unique data, that contains (probably) a Address Part and a Data Part. Again, I don’t worry about what is address or data. Only need unique 32 bits codes. If address part repeats (because you are send codes to one device), data should be unique. Ok to me.

The stream is read as it is delivered from TSOP to MCU, from left to right (START BIT to right). I’m considering from LSB to MSB, so when you see the analyzer stream signal, you are viewing the reversed version. See:



IR Data Stream Mapping

After some nights adjusting some things, the code works great!  I’ll publish it here in the next week.

As I think, some codes changes if you are using TV, DVD, VCR or STB buttons. See the codes for VCR function:



Some codes

If you observe the first pictures, about button “1” stream, you can see the sequence of 1110000 (reverse order, 0x07).

**Proof of concept**

To test my concept, I programing the device to waiting VCR signal functions (ended with ….0505) as you can see:

* Buttons 1,2 and 3 turn ON color RED, GREEN and BLUE;
* Buttons 4,5 and 6 turn OFF color RED, GREEN and BLUE;
* Button 9 turn ALL ON;
* Button 0 turn ALL OFF;
* VOLUME+ increase LED power (PWM duty cycle);
* VOLUME- decrease LED power (PWM duty cycle);

See my video:

**Next Step**

Now I need to design a transformerless power supply and the power control circuitry (TRIAC, etc) to control the real lights. Some nice ideas to next version:

* Store dimming states on internal PIC EEPROM, so you can save desired conditions every time;
* TV Remote now can act in others projects too (robots, toys, other home automation, etc);
* With a small LCD, create a small menu, some configuration options, to more advanced projects;

I hope yours enjoy this project. From the first paper to here, I took one week to get it working properly. In a few weeks, I’ll share all the files here, in downloads page. Thank’s!