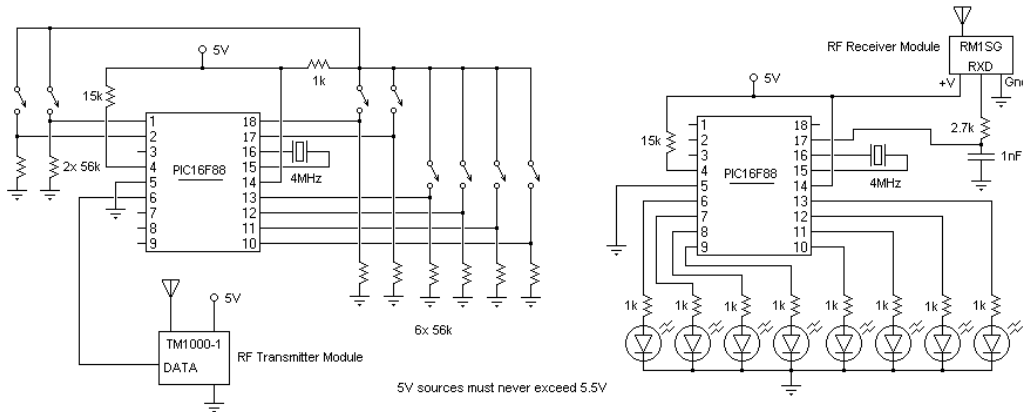


# 8 CHANNEL RF REMOTE CONTROL



After much work I finally got the noisy RF link to work reliably. I had originally thought to make my own RF modules from scratch, but I later scrapped the idea when I found out how cheap real RF modules can be had on ebay. So I bought a TM1000-1 transmitter and RM1SG receiver for a few dollars. The modules weren't as easy to use as I had thought, and unlike a IR modules they send out lots of noise without a steady input signal. Also the RF module I used had long start-up and power down period after receiving a high pulse. To counter all of this I kept the receiver in constant standby mode, but sending a information all the time. That way the noise is flooded out, and the receiver will always respond. I had been trying all sorts of error detection methods and different ways of encoding the bytes, when I just gave up. Since the link is so noisy I decided to cut out all of the error detection methods and just make it accept anything it receives, and see what happened from there. But what do you know, it worked!



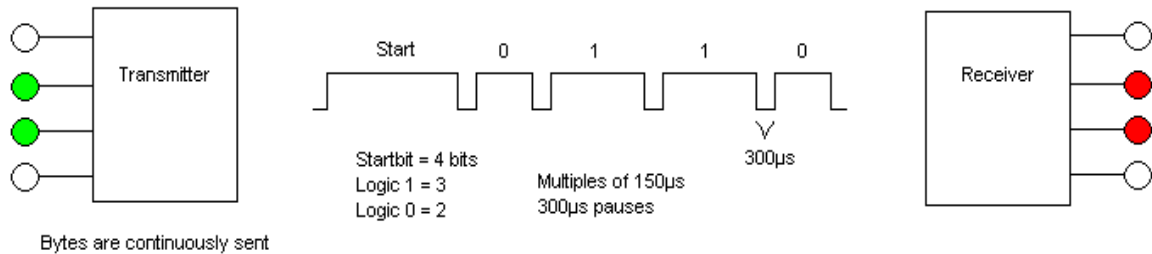
Here are the source files, firmware and schematic.

### How it works:

Unlike the IR transmitter I made, the RF transmitter is designed to constantly send the status of its inputs. 1) because the link is noisy, and there is a good chance the byte will be corrupted. 2) the receiver passes out if no input is received within a few milliseconds of a previous input, and doesn't accept new input until a few hundred milliseconds afterwards. So the receiver is made to recognize a startbit so it knows where the transmission began, it then measures the pulse length, and determines whether it's a 1 or 0. Just like the IR controller. Unlike the IR controller, if a false bit length is sensed the RF receiver ignores it and leaves the value of the array as it was. Meaning if the RC car is moving forward, and the bit which corresponds to forward is corrupted the car will keep moving forward. The benefit of this is that with bytes being sent constantly corrupt bits will be replaced by correct ones quickly.

So even if an entire transmission fails, a new transmission comes right afterwards.

Sort of a brute-force method, but it gets the job done quickly and simply.

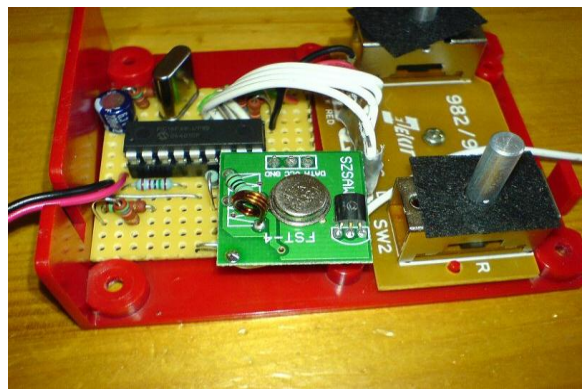


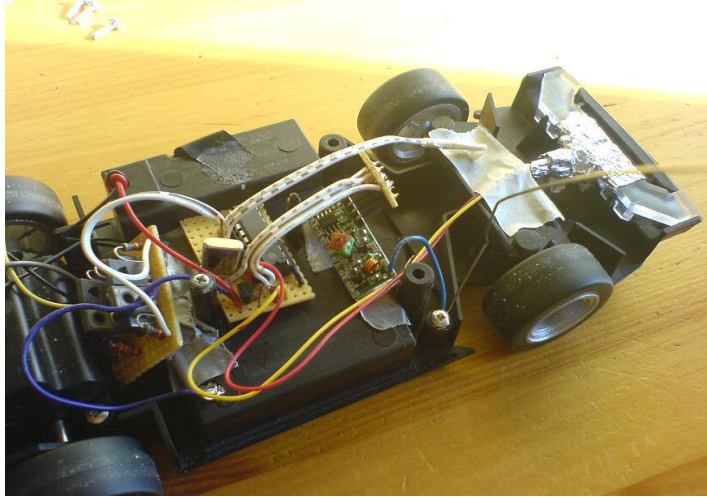
I think I commented the code even better than the IR controller, so just check the source for a more in depth explanation. (It's in basic, written with Mikrobasic) The firmware is designed for two PIC16F88's, but with very few software changes any PIC can be used.

To test the setup I modded the RC car I used with the IR remote control system.

The range seems to be good, although I haven't tested it at more than 15 meters yet.

The transmitter boasts 1km of range under ideal conditions, so I can count on this working anywhere in the house at least. Plus I don't have to run after the car and point at the IR module anymore.





Source: <http://uzzors2k.4hv.org/index.php?page=rfcontrol>