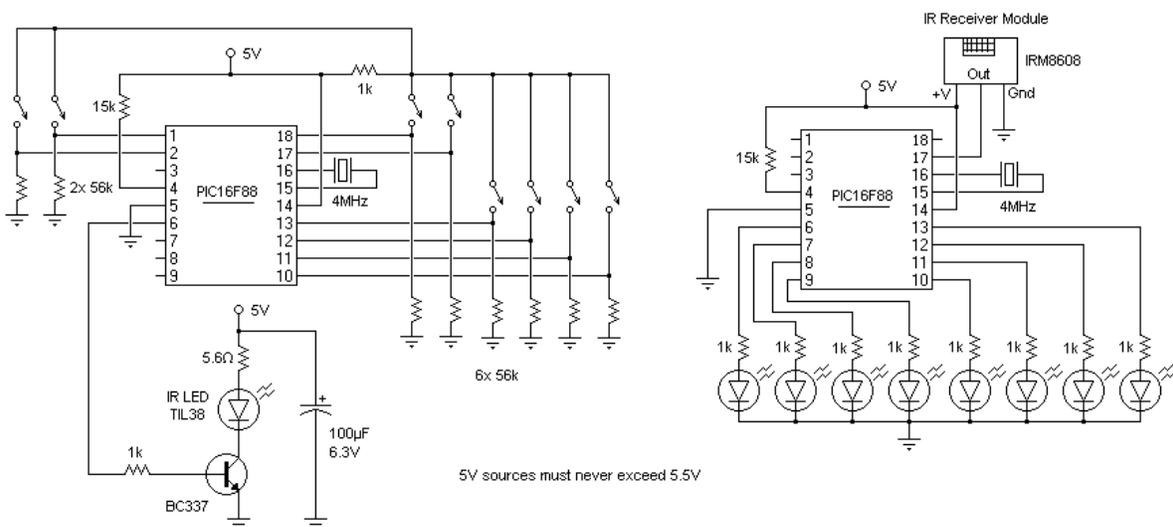


8 CHANNEL INFRARED REMOTE CONTROL



To do something different I thought I would try making a remote control system. It forms the basis for an RF based system I will be making soon. Despite 8 channel remotes with multiple simultaneous output control (mashing lots of buttons won't freeze it up) seem like high-tech equipment, they are simple once broken down into components.



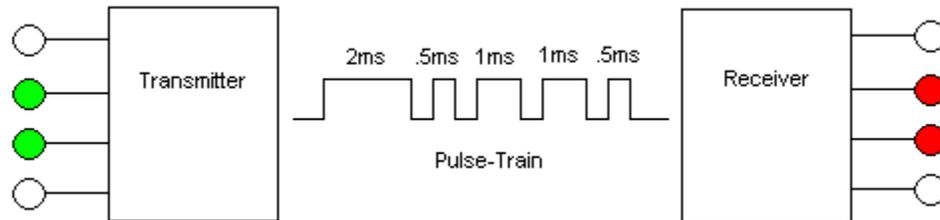
Here is the source, firmware and schematic.

How it works:

When using IR modules, signals can be sent remotely by just plugging some wires together. The receiver module has an internal bandpass filter which filters out anything but IR pulsed at a given frequency. This makes it much more immune to ambient IR, but it will now only respond to IR at its own carrier frequency. The transmitting IR LED has to pulse at this predetermined frequency, which is a job suitable for a 555 timer. Once we've filled these two criteria we can transmit signals without wires, and pulsing the LED will cause the receiver to respond.

With that taken care of we need some way of encoding information from the transmitter so it can be sent through the LED and decoded by the receiver. The transmitter software will sweep through the inputs and set an array with bits according to how the input is. This entire array is then sent via IR. By using an array which changes dynamically any combination of buttons can be pressed at the same time. The IR protocol I made uses a 2ms start pulse, followed by either 1ms pulse for logic 1 or 0.5ms for logic 0. There is 0.5ms between each pulse. The receiving PIC reads this from the receiver module and puts the information into an array. This array is then used to set the output pins. To save power the program will only send a new IR pulse train if the inputs have changed.

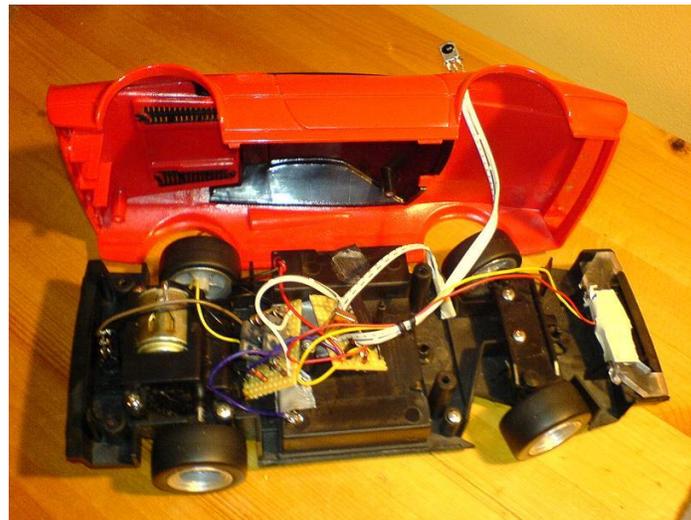
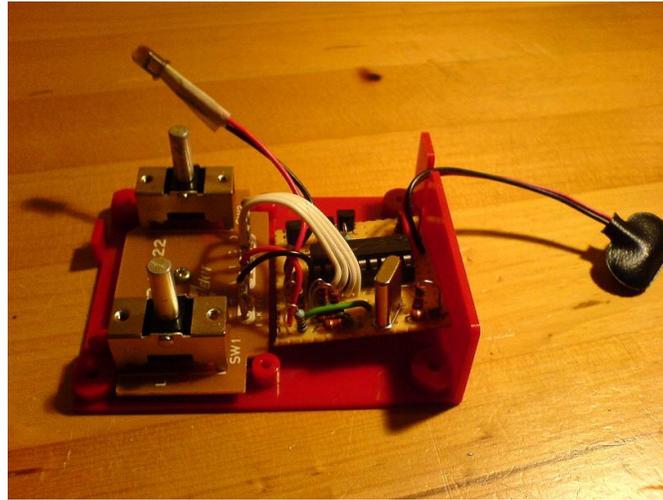
The only disadvantage I have found so far is once the receiver loses reception it will continue with what it was doing. Hardly ideal for RC planes or helicopters!



I think I commented the code well enough for anyone to follow my logic, 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 the PIC16F628 could be used instead. For the receiver even a PIC16F84 can be used!

Of course I had to test the setup, so I found an old RC car with no remote and rebuilt the electronics from scratch. Range is poor with the current IR system, only a few meters but it proves that the concept works. The receiver has a small reception cone which is fine for TVs and stationary objects, but no good for a moving car. Since the receiver is mounted on the roof it almost never gets a clear line of sight and must rely on reflection. To drive the motor I needed an H-bridge, and this one designed by Bob Blick works like a charm.

I used BD139 and 40's for the main drive transistors since the drive current of the motor was so low.



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