Types of used Protocols in RF remote conrtol:

1-Fixed Code

2-Learning Code

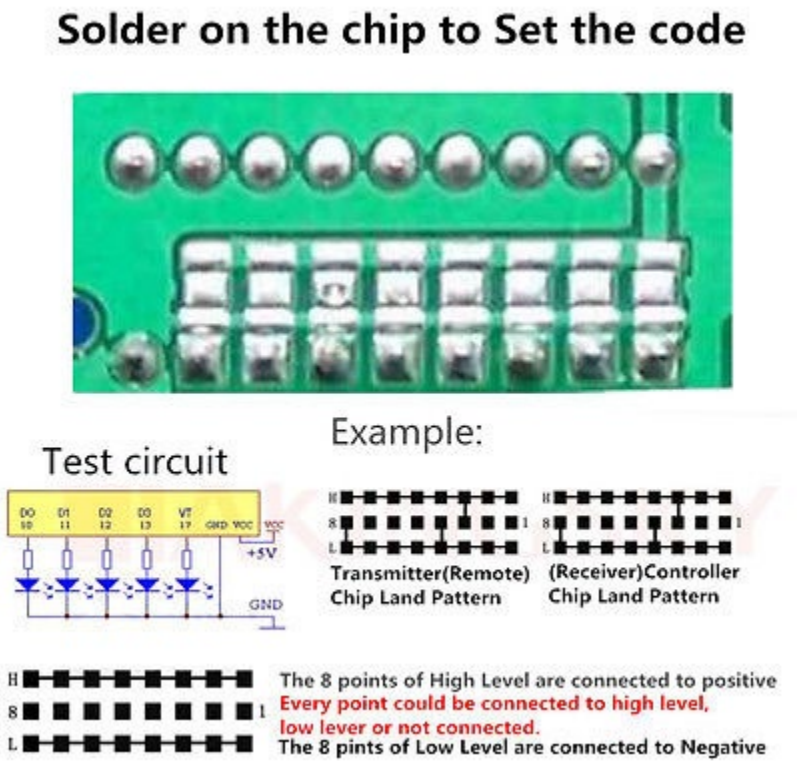
3-Rolling Code

1-Fixed Code:

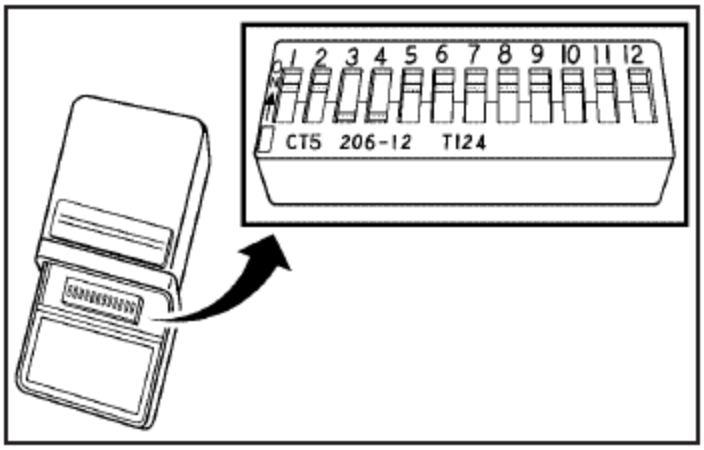
- Principle of operation:

Now it's obvious why it called fixed-code so if we want to use another remote, it's simple: just copy the transmitter's code pattern.

we have two methods to set the fixed code on TX and RX



OR by setting by using DIP switchs:



fixed adress code(38=6561) it could also depends on the used chip

Chips: 2260, 2262, HT12E,pt2262,etc..

Safety factor isn't high since the number of address combinations is limited to the number of dip switchs 2^n which an attacker with appropraite hacking tool which can try all the combinations used which a computer can iterate through a 4000 combination in seconds or record the transmitted code from the victim remote and later unlock the coor or whatever locked device.

2-Learning Code:

But learning-code remotes have more security. It means every remote has a 20 bits written random code in its memory. This code is unique for every remote, so for syncing, receiver has to save every remote’s code and for every request check if it’s a valid remote or not.This method also support encryption .

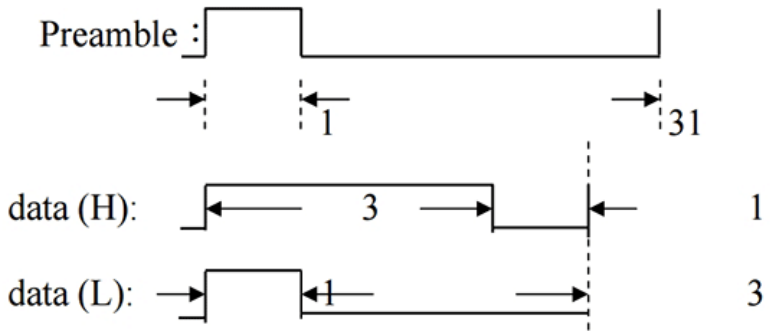
Used chips:

HS1527 EV1527,1527, 2240, HT6P20B

number of combination addresses 2^20=1million codes

Output data frame:

Preambl e	C0 ~ C19 (1 million codes)	D0	D1	D2	D3



In this method we have 24 bits of data consist of 20 bits that is remote's code and 4 bits for the remote's buttons statues. First we have a Preamble situation then wait for 24 bits of data so we need 3 unique situations:

Preamble: in this situation if we have 1 for one µs, must have 0 for 30µ

Logical 1: in this situation if we have 1 for 3 µs, must have 0 for one µ

Logical 0: in this situation if we have 1 for one µs, must have 0 for 3 µ

Note that the times mentioned are for example and depends on inner oscillator; but ratios are true.

safety is minimum we are going to look later on methods how to by pass learning code and even rolling code.