# 全面了解红外遥控

作者：San Bergmans

转载时间：2004年06月10日

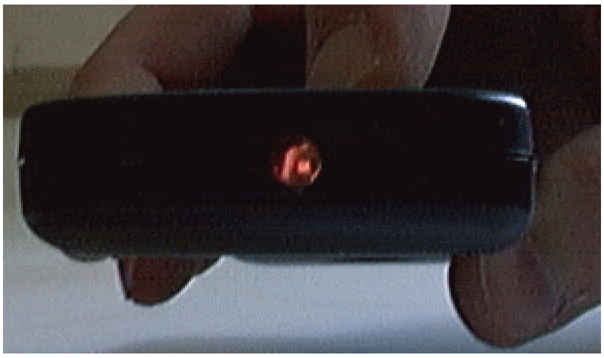
## IR Remote Control Theory

The cheapest way to remotely control a device within a visible range is via Infra-Red light. Almost all audio and video equipment can be controlled this way nowadays. Due to this wide spread use the required components are quite cheap. Thus making it ideal for us hobbyists to use for our own projects.
This part of my knowledge base will explain the theory of operation of IR remote control, and some of the protocols that are in use in consumer electronics.

## Infra-Red Light

Infra-Red actually is normal light with a particular colour. Us humans can't see this colour because it's wave length of 950nm is below the visible spectrum. That's one of the reasons why IR is chosen for remote control purposes, we want to use it but we're not interested in seeing it. Another reason is because IR LEDs are quite easy to make, and therefor can be very cheap.

Although we humans can't see the Infra-Red light emitted from a remote control doesn't mean we can't make it visible.
A video camera or digital camera can "see" the Infra-Red light as you can see in this picture.

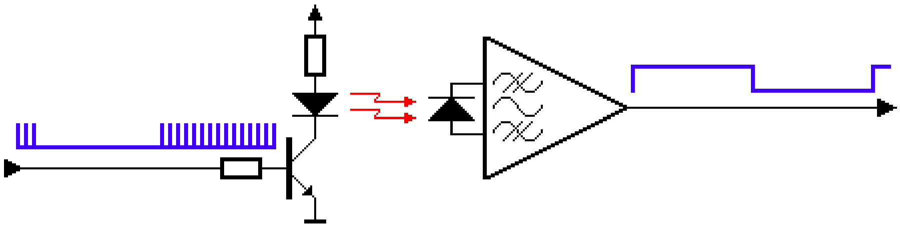


Unfortunately for us there are many more sources of Infra-Red light. The sun is the brightest source of all, but there are many others, like: light bulbs, candles, central heating system, and even our body radiates Infra-Red light. In fact everything that radiates heat, also radiates Infra-Red light.

Therefor we have to take some precautions to guarantee that our IR message gets across to the receiver without errors.

## Modulation

Modulation is the answer to make our signal stand out above the noise. With modulation we make the IR light source blink in a particular frequency. The IR receiver will be tuned to that frequency, so it can ignore everything else.
You can think of this blinking as attracting the receiver's attention. We humans also notice the blinking of yellow lights at construction sites instantly, even in brought daylight.



In the picture above you can see a modulated signal driving the IR LED of the transmitter on the left side. The detected signal is coming out of the receiver at the other side.

In serial communication we usually speak of 'marks' and 'spaces'. The 'space' is the default signal, which is the off state in the transmitter case. No light is emitted during the 'space' state. During the 'mark' state of the signal the IR light is pulsed on and off at a particular frequency. Frequencies between 30kHz and 60kHz are commonly used in consumer electronics.

At the receiver side a 'space' is represented by a high level of the receiver's output. A 'mark' is then automatically represented by a low level.

Please note that the 'marks' and 'spaces' are not the 1-s and 0-s we want to transmit. The real relationship between the 'marks' and 'spaces' and the 1-s and 0-s depends on the protocol that's being used. More information about that can be found on the pages that describe the protocols.

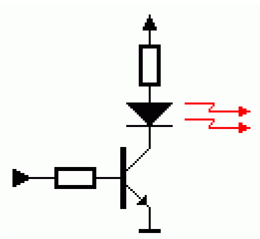
## The Transmitter

The transmitter usually is a battery powered handset. It should consume as little power as possible, and the IR signal should also be as strong as possible to achieve an acceptable control distance. Preferably it should be shock proof as well.

Many chips are designed to be used as IR transmitters. The older chips were dedicated to only one of the many protocols that were invented. Nowadays very low power microcontrollers are used in IR transmitters for the simple reason that they are more flexible in their use. When no button is pressed they are in a very low power sleep mode, in which hardly any current is consumed. The processor wakes up to transmit the appropriate IR command only when a key is pressed.

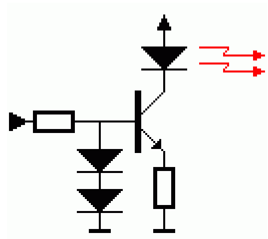
Quartz crystals are seldom used in such handsets. They are very fragile and tend to break easily when the handset is dropped. Ceramic resonators are much more suitable here, because they can withstand larger physical shocks. The fact that they are a little less accurate is not important.

The current through the LED (or LEDs) can vary from 100mA to well over 1A! In order to get an acceptable control distance the LED currents have to be as high as possible. A trade-off should be made between LED parameters, battery lifetime and maximum control distance. LED currents can be that high because the pulses driving the LEDs are very short. Average power dissipation of the LED should not exceed the maximum value though. You should also see to it that the maximum peek current for the LED is not exceeded. All these parameters can be found in the LED's data sheet.



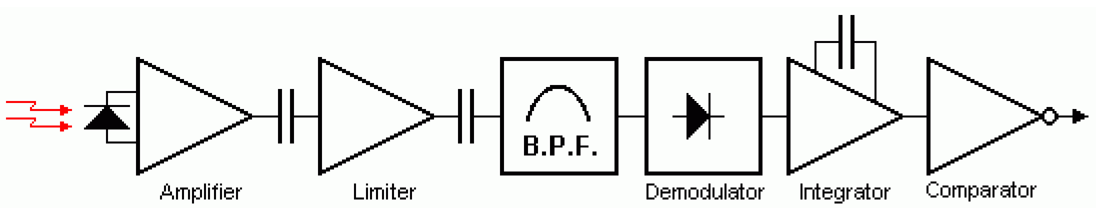
A simple transistor circuit can be used to drive the LED. A transistor with a suitable HFE and switching speed should be selected for this purpose.
The resistor values can simply be calculated using Ohm's law. Remember that the nominal voltage drop over an IR LED is approximately 1.1V.

The normal driver, described above, has one disadvantage. As the battery voltage drops, the current through the LED will decrease as well. This will result in a shorter control distance that can be covered.
An emitter follower circuit can avoid this. The 2 diodes in series will limit the pulses on the base of the transistor to 1.2V. The base-emitter voltage of the transistor subtracts 0.6V from that, resulting in a constant amplitude of 0.6V at the emitter. This constant amplitude across a constant resistor results in current pulses of a constant magnitude. Calculating the current through the LED is simply applying Ohm's law again.



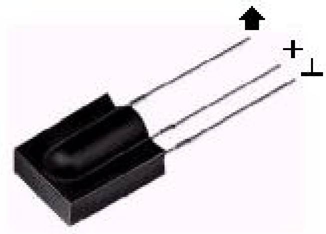
## The Receiver

Many different receiver circuits exist on the market. The most important selection criteria are the modulation frequency used and the availability in you region.




In the picture above you can see a typical block diagram of such an IR receiver. Don't be alarmed if you don't understand this part of the description, for everything is built into one single electronic component.

The received IR signal is picked up by the IR detection diode on the left side of the diagram. This signal is amplified and limited by the first 2 stages. The limiter acts as an AGC circuit to get a constant pulse level, regardless of the distance to the handset.
As you can see only the AC signal is sent to the Band Pass Filter. The Band Pass Filter is tuned to the modulation frequency of the handset unit. Common frequencies range from 30kHz to 60kHz in consumer electronics.
The next stages are a detector, integrator and comparator. The purpose of these three blocks is to detect the presence of the modulation frequency. If this modulation frequency is present the output of the comparator will be low.



As I said before, all these blocks are integrated into a single electronic component. There are many different manufacturers of these components on the market. And most devices are available in several versions each of which are tuned to a particular modulation frequency.

Please note that the amplifier is set to a very high gain. Therefore the system tends to start oscillating very easily. Placing a large capacitor of at least 22µF close to the receiver's power connections is mandatory to decouple the power lines. Some data sheets recommend a resistor of 330 Ohms in series with the power supply to further decouple the power supply from the rest of the circuit.

There are several manufacturers of IR receivers on the market. Siemens, Vishay and Telefunken are the main suppliers here in Europe. Siemens has its SFH506-xx series, where xx denotes the modulation frequency of 30, 33, 36, 38, 40 or 56kHz. Telefunken had its TFMS5xx0 and TK18xx series, where xx again indicates the modulation frequency the device is tuned to. It appears that these parts have now become obsolete. They are replaced by the Vishay TSOP12xx, TSOP48xx and TSOP62xx product series.
Sharp, Xiamen Hualian and Japanese Electric are 3 Asian IR receiver producing companies. Sharp has devices with very cryptic ID names, like: GP1UD26xK, GP1UD27xK and GP1UD28xK, where x is related to the modulation frequency. Hualian has it's HRMxx00 series, like the HRM3700 and HRM3800. Japanese Electric has a series of devices that don't include the modulation frequency in the part's ID. The PIC-12042LM is tuned to 36.7kHz, and the PIC12043LM is tuned to 37.9kHz.

## The End?

This concludes the theory of operation for IR remote control systems intended for use in consumer electronics. I realise that other ways exist to implement IR control, but I will limit myself to the description above. One of the issues not covered here is security. Security is of no importance if I want to control my VCR or TV set. But when it comes to opening doors or cars it literally becomes a 'key' feature! Maybe I will cover this issue later, but not for now.

I also realise that my small list of manufacturers is far from being complete. It is hardly possible to list every manufacturer here.

This page only described the basic theory of operation of IR remote control. It did not describe the protocols that are involved in communication between transmitter and receiver. Many protocols are designed by different manufacturers. You can find the protocols of some manufacturers in the link section at the top of this page.

## ITT Protocol

The ITT IR protocol is a very old one. It differs from other protocols in that it does not use a modulated carrier frequency to send the IR messages. A single command is transmitted by a total of 14 pulses with a width of 10µs each. The command is encoded by the distance between the pulses.
This protocol is very reliable and consumes very little power ensuring long battery life.

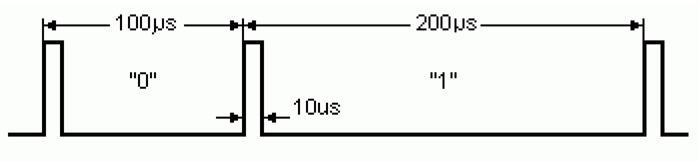
Many consumer electronics brands used this protocol in Europe. Among them were: ITT, Greatz, Schaub-Lorenz, Finlux, Luxor, Salora, Oceanic and later also Nokia, to name but a few.

## Features

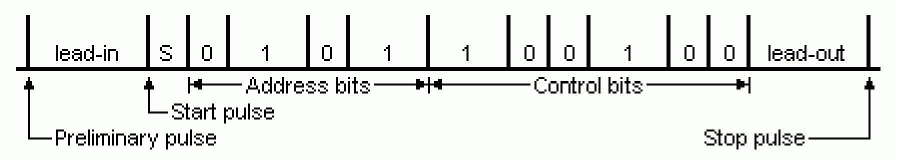
- Only 14 very short IR pulses per message
- Pulse distance encoding
- Long battery life
- 4 bit address, 6 bit command length
- Self calibrating timing, allowing only simple RC oscillator in the transmitter
- Fast communication, a message takes from 1.7ms to 2.7ms to transmit

- Manufacturer Intermetal, now Micronas

# Protocol




An IR message is transmitted by sending 14 pulses. Each pulse is 10µs long. Three different time intervals are used to get the message across: 100µs for a logic 0, 200µs for a logic 1 and 300µs for the lead-in and lead-out.




The preliminary pulse is used by the receiver to set the gain of the amplifier. Then follows a lead-in delay of 300µs, after which the starting pulse is given. The first bit sent is always logic 0, which has a delay duration of 100µs. This start bit can be used to calibrate the timing of the receiver. After the start bit follow 4 bits (MSB first) that represent the address of the message. After that a total of 6 bits (MSB first) for the command are transmitted. A trailing pulse should follow this command. Finally another 300µs delay follows before the very last pulse, functioning as a leadout.

There are a few things the receiving software can check to verify the validity of the received message. The leadout interval should be 3 times longer than the start bit time, which has a duration of 100µs. Bit times should not be off by more than ±20% of the start bit length for logic 0s, or 2x the start bit length for logic 1s.
Don't keep waiting for pulses after 360µs after the last received pulse. It's very likely that the transmission is interrupted or no transmission took place at all if you have to wait longer than that.
The preliminary pulse serves only AGC purposes and may be ignored by the receiving software. Decoding of the message should start at the Start pulse.

# Address and Command

A control message is divided into two groups, an address of 4 bits and a command of 6 bits. Addresses range from 1 to 16, and commands range from 1 to 64. Before the address and command are sent, 1 is subtracted from both values to get them in the range 0 to 15 and 0 to 63.

Addresses are used in pairs. A pair of addresses is a value of 1 to 8 (0 to 7 actually), and it's inverted counter part 16 to 9 (15 to 8 actually).
The lower value address is transmitted the first time a key is pressed. The address value of all subsequent messages will be the inverted value of this first address until the key is released. This enables the receiver to interpret repeat codes properly. Messages are repeated every 130ms as long as the key remains pressed.

# Transmitter

Intermetall has developed a few transmitter ICs for use in handsets. Later microcontrollers were used to facilitate the combination of TV, VCR and SAT remote control in one handset.

The SAA1250 was the first IR controller IC to be released. It can be set to generate 3 different address pairs. A fourth option is transmitting any of the 16 addresses. That option is rarely used, for it requires a manual setup procedure every time the power is lost.

The second generation of IR controller ICs are the IRT1250 and IRT1260. These chips are identical in operation and differ only in the operating voltage. The IRT1250 is intended for 9V operation, whilst the IRT1260 is designed for 3V.
The footprint of the IRT12x0 is the same as that of the SAA1250. The devices differ in addressing capability and current drive capacity for the output stage.

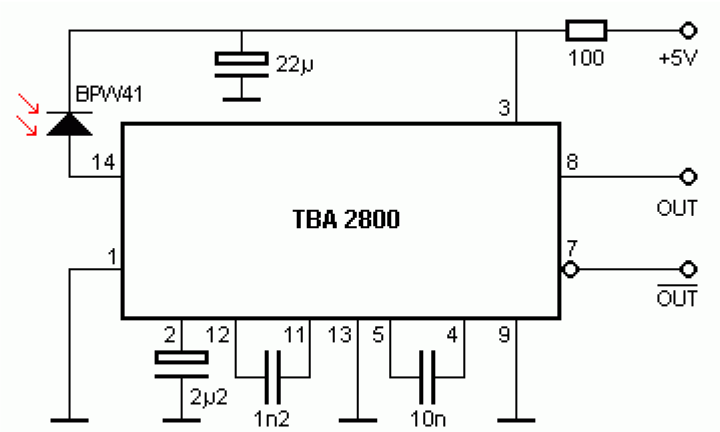
Two address pins are available to set the address pair used.

| A1 | A2 | Adresses |
|---|---|---|
| H | H | 1 & 16 |
| L | H | 3 & 14 |
| H | L | 7 & 10 |
| L | L | 4 & 13 |

Addresses 1 and 16 are always used to control TV sets. Other address pairs are not always uniquely linked to a particular equipment family.

## Receiver

The ITT protocol makes no use of a modulated carier, so the previously mentioned IR receivers won't work for this protocol. Intermetall has created the TBA2800 for use with this protocol. It is a highly senisive IR detection circuit and should be shielded completely inside a metal box that is connected to ground, leaving only a small hole just in front of the IR diode.




There is actually not much more to be told about this IC. Just connect it as shown in the diagram as shown above, and it should work. You can chose between a normal high going output, and an inverted low going output. It depends on the rest of your circuitry which one you should use.
In case of excessive interference you could reduce the sensitivity a little by grounding pin 6 via a 10k resistor.

## Pre-defined Commands

Some of the 64 possible commands are predefined. But unfortunately the definition of the commands is not as clear as with RC-5. You can find most of the pre-defined commands for TV purposes in the following table. TV commands use the address pair 1 and 16.

| Command | Function |
|---|---|
| 1 | |
| 2 | Standby |
| 3 | TV |
| 4 | Ideal |
| 5 | Up |
| 6 | Down |
| 7 | Mute |
| 8 | P+ |
| 9 | P- |
| 10 | Left / Bilingual |
| 11 | Right |
| 12 | |
| 13 | |
| 14 | Last |
| 15 | |
| 16 | |

| Command | Function |
|---|---|
| 33 | |
| 34 | |
| 35 | -/-- |
| 36 | Audio |
| 37 | Video |
| 38 | Clock |
| 39 | |
| 40 | |
| 41 | |
| 42 | |
| 43 | Brightness + |
| 44 | Brightness - |
| 45 | Saturation + |
| 46 | Saturation - |
| 47 | Volume + |
| 48 | Volume - |

| | |
|---|---|
| 17 | 1 |
| 18 | 2 |
| 19 | 3 |
| 20 | 4 |
| 21 | 5 |
| 22 | 6 |
| 23 | 7 |
| 24 | 8 |
| 25 | 9 |
| 26 | 0 |
| 27 | |
| 28 | Zoom |
| 29 | |
| 30 | X |
| 31 | |
| 32 | Info |

| | |
|---|---|
| 49 | |
| 50 | S |
| 51 | Red / Memory |
| 52 | Green |
| 53 | Contrast |
| 54 | Blue / Brightness |
| 55 | Yellow / Saturation |
| 56 | |
| 57 | |
| 58 | |
| 59 | Menu |
| 60 | Auto |
| 61 | Text |
| 62 | OK / Prog |
| 63 | |
| 64 | C |

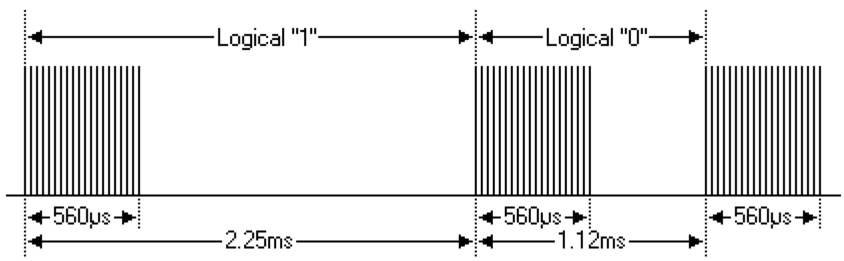
# NEC Protocol

To my knowledge the protocol I describe here was developed by NEC. I've seen very similar protocol descriptions on the internet, and there the protocol is called Japanese Format.
I do admit that I don't know exactly who developed it. What I do know is that it is used in my ancient VCR produced by Sharp and was marketed under the name of Fisher. NEC manufactured the remote control IC.
This description was taken from the VCR's service manual.

Features

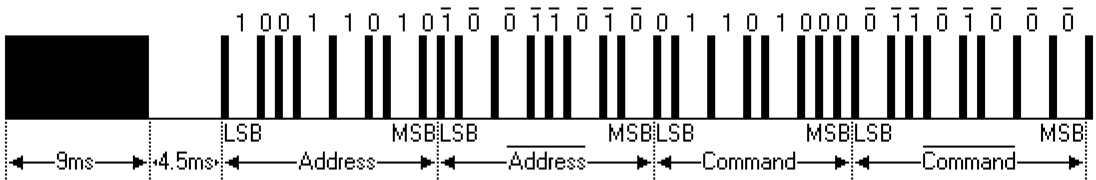
- 8 bit address and 8 bit command length
- Complete address and command are transmitted twice for reliability
- Pulse distance modulation
- Carrier frequency of 38kHz
- Bit time of 1.12ms or 2.25ms

Modulation

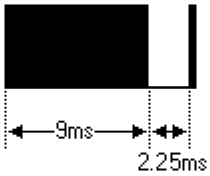



The NEC protocol uses a pulse distance encoding of the bits. Each pulse is a 560µs long 38kHz carrier burst (about 21 cycles). A logical "1" takes 2.25ms to transmit, while a logical "0" is only 1.12ms. The recommended carrier duty-cycle is 1/4 or 1/3.

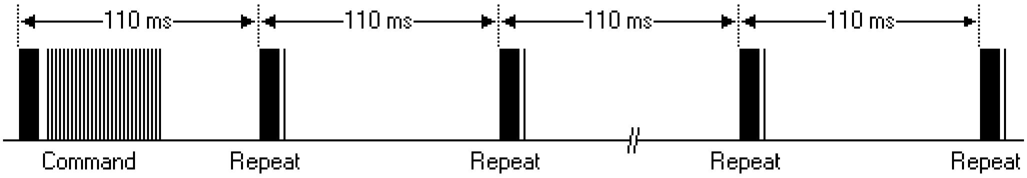
Protocol




The picture above shows a typical pulse train of the NEC protocol. With this protocol the LSB is transmitted first. In this case Address $59 and Command $16 is transmitted. A message is started by a 9ms AGC burst, which was used to set the gain of the earlier IR receivers. This AGC burst is then followed by a 4.5ms space, which is then followed by the Address and Command. Address and Command are transmitted twice. The second time all bits are inverted and can be used for verification of the received message. The total transmission time is constant because every bit is repeated with its inverted length. If you're not interested in this reliability you can ignore the inverted values, or you can expand the Address




and Command to 16 bits each!
A command is transmitted only once, even when the key on the remote control remains pressed. Every 110ms a repeat code is transmitted for as long as the key remains down. This repeat code is simply a 9ms AGC pulse followed by a 2.25ms space and a 560µs burst.




Example Commands

The table below lists the messages sent by the remote control of my senior Fisher 530 VCR.

| NEC Message | Key Function |
|---|---|
| $68-$00 | Play |
| $68-$01 | Rec |
| $68-$02 | Audio Dub |
| $68-$03 | Frame Adv |
| $68-$04 | Slow |
| $68-$05 | Quick |
| $68-$06 | Cue |
| $68-$07 | Review |
| $68-$08 | FF |
| $68-$09 | Rew |
| $68-$0A | Stop |
| $68-$0B | Pause/Still |
| $68-$0C | Up key |
| $68-$1E | Down key |

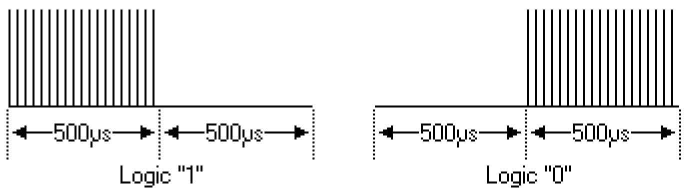
# Nokia NRC17 Protocol

The **N**okia **R**emote **C**ontrol protocol uses **17** bits to transmit the IR commands, which explains the name of this protocol. The protocol was designed for Nokia consumer electronics. It was used during the last few years that Nokia produced TV sets and VCRs. Also the sister brands like Finlux and Salora used this protocol. Nowadays the protocol is mainly used in Nokia satellite receivers.

Features

- 8 bit command, 4 bit address and 4 bit subcode length
- Bi-phase coding
- Carrier frequency of 38kHz
- Bit time of 1ms
- Battery empty indication possible
- Manufacturer Nokia CE

Modulation

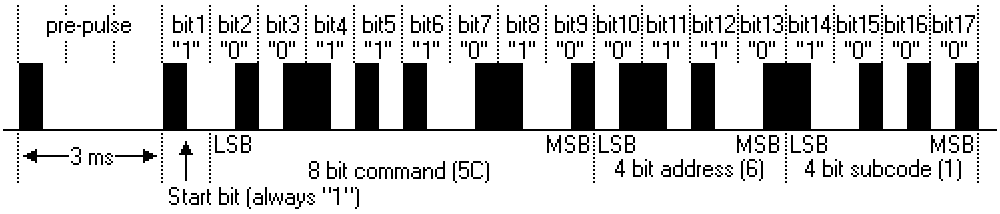



The protocol uses bi-phase (or so-called NRZ - Non Return to Zero) modulation of a 38kHz IR carrier frequency. All bits are of equal length of 1ms in this protocol, with half of the bit time filled with a burst of the 38kHz carrier and the other half being idle. A logical one is represented by a burst in the first half of the bit time. A logical zero is represented by a burst in the second half of the bit time.

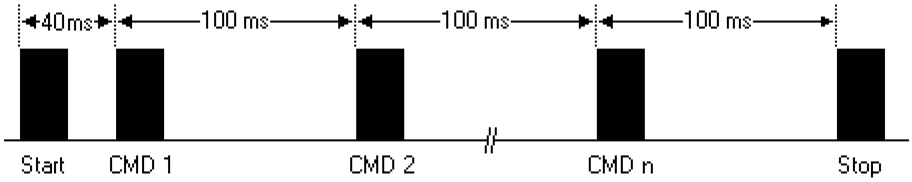
The pulse/pause ratio of the 38kHz carrier frequency is 1/4, to reduce power consumption.

## Protocol

The drawing below shows a typical pulse train of an NRC17 message. This example transmits command $5C to address $6 subcode $1.




The first pulse is called the pre-pulse, and is made up of a 500µs burst followed by a 2.5ms pause, giving a total of 3 bit times.
Then the Start bit is transmitted, which is always a logic "1". This pulse can be used to calibrate the bit time on the receiver side, because the burst time is exactly half a bit time.
The next 8 bits represent the IR command, which is sent with LSB first. The command is followed by a 4 bit device address. Finally a 4 bit subcode is transmitted, which can be seen as an extension to the address bits.
A message consists of a 3ms pre-pulse and 17 bits of 1ms each. This adds up to a total of 20ms per message.




Every time a key is pressed on the remote control a start message is transmitted containing a command of $FE and address/subcode of $FF. The actual message is send 40ms later, and is repeated every 100ms for as long as the key on the remote control remains down. When the key is released a stop message will complete the sequence. The stop message also uses the command $FE and address/subcode $FF.
Every sequence can be treated as one single sequence at the receivers end because of the start and stop messages. Accidental key bounces are effectively eliminated by this procedure.
The receiver may decide to honour the repeated messages or not. E.g. cursor movements may repeat for as long as the key is pressed. Numerical inputs better don't allow auto repeat.

## Low Battery

The NRC17 protocol provides in a way for the remote control to tell the receiver that the battery capacity is getting low. The receiver may display a message on the TV screen informing the user that the remote control's batteries are to be replaced.
The pre-pulse normally is 3ms long. When the battery power is low this pre-pulse will become 4ms long. In practice only the pre-pulse of the start and stop messages are made longer this way.

## Pre-defined Commands

I only have a small list of pre-defined commands. The protocol description that I have doesn't specify more. Please note that the address of the SAT commands applies to Analog receivers only.

| **NRC17** Command | **CTV** Address: $A Subcode: $4 | **SAT** Address: $C Subcode: $0 |
|---|---|---|
| $00 | 0 / Extern | 0 / Extern |
| $01 | 1 | 1 |
| $02 | 2 | 2 |
| $03 | 3 | 3 |
| $04 | 4 | 4 |
| $05 | 5 | 5 |
| $06 | 6 | 6 |
| $07 | 7 | 7 |
| $08 | 8 | 8 |
| $09 | 9 | 9 |

| | | |
|---|---|---|
| $0C | Standby | Standby |
| $0E | Up key | Up key |
| $0F | Down key | Down key |
| $28 | Mute | Mute |
| $29 | Ideal | Reveal |
| $2A | Alternate | Alternate |
| $2D | Index | Index |
| $2E | Right key | Right key |
| $2F | Left key | Left key |
| $33 | Text | Text |
| $35 | Stop | Stop |
| $38 | Size | Size |
| $3C | Red (OK) | Red |
| $3D | Green (Sound) | Green |
| $3E | Yellow (Picture) | Yellow |
| $3F | Blue (Extra) | 1 |
| $70 | TV | TV/SAT |

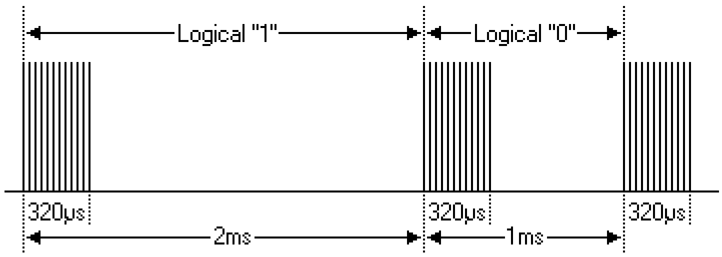
# Sharp Protocol

I have little information on this protocol. It is used in VCRs that are produced by Sharp, that is why I gave it the name Sharp protocol.

Features

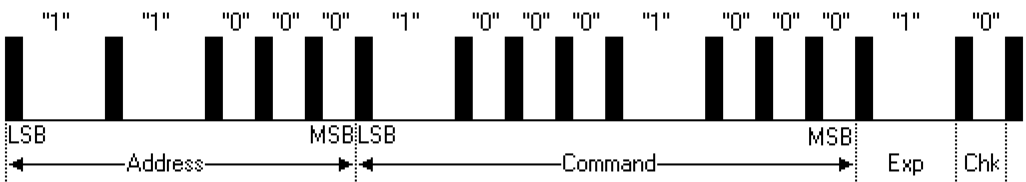
- 8 bit command, 5 bit address length
- Pulse distance modulation
- Carrier frequency of 38kHz
- Bit time of 1ms or 2ms

Modulation

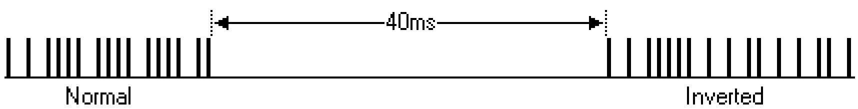



The Sharp protocol uses a pulse distance encoding of the bits. Each pulse is a 320µs long 38kHz carrier burst (about 12 cycles). A logical "1" takes 2ms to transmit, while a logical "0" is only 1ms. The recommended carrier duty-cycle is 1/4 or 1/3.

Protocol




In the picture above you see a typical pulse train sending the command $11 and address $03. The Address is send first and consists of 5 bits. Next comes the 8 bit command. In both cases the LSB of the data is send first.
I don't know the purpose of the Expansion and Check bits that follow the command. Both bits were fixed in the example that I had at hand.



One complete command sequence consist of 2 messages. The first transmission is exactly as described above. The second transmission follows the first one after a delay of 40ms, and basically contains the same information. The only difference is that all bits, except those from the address field, are inverted.
This way the receiver can verify if the received message is reliable or not.

# Sony SIRC Protocol

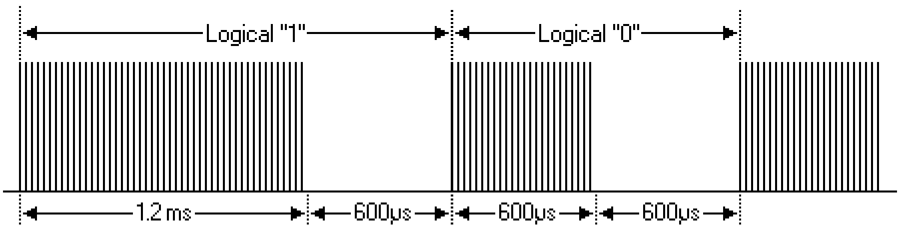
I've collected and combined some information found on the internet about the Sony SIRC protocol. I must admit that I have never worked with this particular protocol, so I could not verify that all information is valid for all situations.
It appears that 3 versions of the protocol exist: 12-bit (described on this page), 15-bit and 20-bit versions. I can only assume that the 15-bit and 20-bit versions differ in the number of transmitted bits per command sequence.

Please note that a lot of confusing documentation about the SIRC protocol exists on the internet. At first I contributed to the confusion by assuming the correctness of the source documents I found myself, until someone with some SIRC experience informed me about my errors. I double checked his story with a universal remote control and a digital storage oscilloscope, and found that the bit and word order I documented were indeed wrong.
The protocol information on this page is according to my own measurements, and should be correct now.

## Features

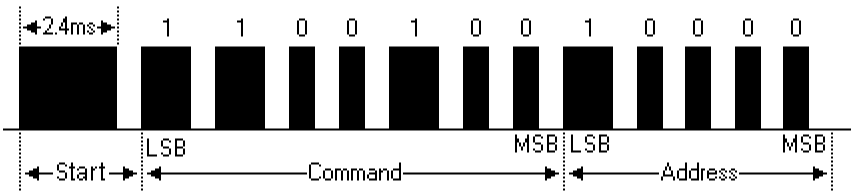
- 12-bit, 15-bit and 20-bit versions of the protocol exist (12-bit described here)
- 5-bit address and 7-bit command length (12-bit protocol)
- Pulse width modulation
- Carrier frequency of 40kHz
- Bit time of 1.2ms or 0.6ms

## Modulation




The SIRC protocol uses a pulse width encoding of the bits. The pulse representing a logical "1" is a 1.2ms long burst of the 40kHz carrier, while the burst width for a logical "0" is 0.6ms long. All bursts are seperated by a 0.6ms long space periode. The recommended carrier duty-cycle is 1/4 or 1/3.

## Protocol




The picture above shows a typical pulse train of the SIRC protocol. With this protocol the LSB is transmitted first. The start burst is always 2.4ms wide, followed by a standard space of 0.6ms. Apart from signalling the start of a SIRC message this start burst is also used to adjust the gain of the IR receiver. Then the 7-bit Command is transmitted, followed by the 5-bit Device address. In this case Address 1 and Command 19 is transmitted.

Commands are repeated every 45ms(measured from start to start) for as long as the key on the remote control is held down.

## Example Commands

The table below lists some messages sent by Sony remote controls in the 12-bit protocol. This list is by no means meant to be complete, as the assignment of functions is probably quite dynamic.

| Address | Device |
|---|---|
| 1 | TV |
| 2 | VCR 1 |
| 3 | VCR 2 |
| 6 | Laser Disc Unit |
| 12 | Surround Sound |
| 16 | Cassette deck / Tuner |
| 17 | CD Player |
| 18 | Equaliser |

| Command | Command |
|---|---|
| 0 | Digit key 1 |
| 1 | Digit key 2 |
| 2 | Digit key 3 |
| 3 | Digit key 4 |
| 4 | Digit key 5 |
| 5 | Digit key 6 |
| 6 | Digit key 7 |
| 7 | Digit key 8 |
| 8 | Digit key 9 |
| 9 | Digit key 0 |
| 16 | Channel + |
| 17 | Channel - |
| 18 | Volume + |
| 19 | Volume - |
| 20 | Mute |
| 21 | Power |
| 22 | Reset |
| 23 | Audio Mode |
| 24 | Contrast + |
| 25 | Contrast - |
| 26 | Colour + |
| 27 | Colour - |
| 30 | Brightness + |
| 31 | Brightness - |
| 38 | Balance Left |
| 39 | Balance Right |
| 47 | Standby |

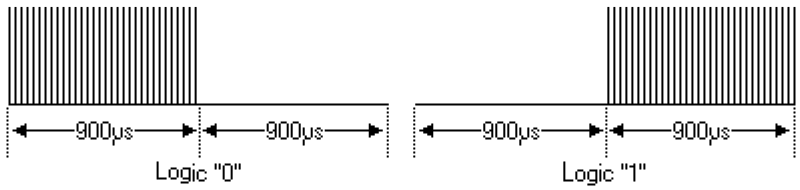
# Philips RC-5 Protocol

The RC5 code from Philips is probably the most used protocol by hobbyists, probably because the wide availability of cheap remote controls.
The protocol is well defined for different device types ensuring compatibility with your whole entertainment system. Lately Philips started using a new protocol called RC6 which has more features.

Features

- 5 bit address and 6 bit command length
- Bi-phase coding (aka Manchester coding)
- Carrier frequency of 36kHz
- Bit time of 1.8ms
- Manufacturer Philips

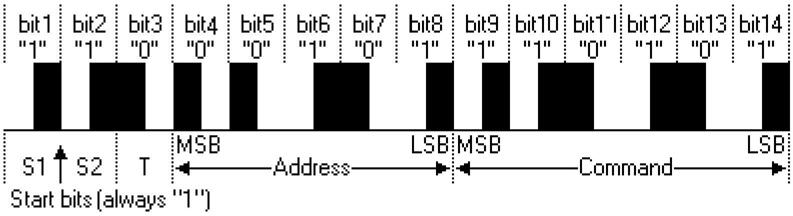
Modulation




The protocol uses bi-phase modulation (or so-called Manchester coding) of a 36kHz IR carrier frequency. All bits are of equal length of 1.8ms in this protocol, with half of the bit time filled with a burst of the 36kHz carrier and the other half being idle. A logical zero is represented by a burst in the first half of the bit time. A logical one is represented by a burst in the second half of the bit time. The pulse/pause ratio of the 36kHz carrier frequency is 1/3 or 1/4, to reduce power consumption.

Protocol

The drawing below shows a typical pulse train of an RC5 message. This example transmits command $35 to address $05.




The first two pulses are the start pulses, and are both logical "1". Please note that half a bit time is elapsed before the receiver will notice the real start of the message.
Extended RC5 uses only one start bit. Bit S2 is transformed to command bit 6, providing for a total of 7 command bits.

The 3d bit is a toggle bit. This bit is inverted every time a key is released and pressed again. This way the receiver can distinguish between a key that remains down, or is pressed repeatedly.
The next 5 bits represent the IR device address, which is sent with MSB first. The address is followed by a 6 bit command, again sent with MSB first.
A message consists of a total of 14 bits, which adds up to a total duration of 25.2ms. Sometimes a message may appear to be shorter because the first half of the start bit S1 remains idle. And if the last bit of the message is a logic "0" the last half bit of the message is idle too.

As long as a key remains down the message will be repeated every 114ms. The toggle bit will retain the same logical level during all of these repeated messages. It is up to the receiver software to interpret this auto repeat feature.

PS: I had rather a big error on this page for quite some time now. For some mysterious reason the LSB and MSB of the address and command were reversed. I can recall correcting this error before, but somehow an old version of the description must have sneeked its way to the internet again.

Pre-defined Commands

Philips has created a beautiful list of "standardised" commands. This ensures the compatibility between devices from the same brand.
A very nice feature, often to be missed with other brands, is the fact that most devices are available twice in the table allowing you to have 2 VCRs stacked on top of each other without having trouble addressing only one of them with your remote control.
I can only show a limited list of standard commands, for this list is about all I know right now.

| RC5 Address | Device |
|---|---|
| $00 - 0 | TV1 |
| $01 - 1 | TV2 |
| $02 - 2 | Teletext |
| $03 - 3 | Video |
| $04 - 4 | LV1 |
| $05 - 5 | VCR1 |
| $06 - 6 | VCR2 |
| $07 - 7 | Experimental |
| $08 - 8 | Sat1 |
| $09 - 9 | Camera |
| $0A - 10 | Sat2 |
| $0B - 11 | |
| $0C - 12 | CDV |
| $0D - 13 | Camcorder |
| $0E - 14 | |
| $0F - 15 | |
| $10 - 16 | Pre-amp |
| $11 - 17 | Tuner |
| $12 - 18 | Recorder1 |
| $13 - 19 | Pre-amp |
| $14 - 20 | CD Player |
| $15 - 21 | Phono |

| RC5 Command | TV Command | VCR Command |
|---|---|---|
| $00 - 0 | 1 | 1 |
| $01 - 1 | 2 | 2 |
| $02 - 2 | 2 | 2 |
| $03 - 3 | 3 | 3 |
| $04 - 4 | 4 | 4 |
| $05 - 5 | 5 | 5 |
| $06 - 6 | 6 | 6 |
| $07 - 7 | 7 | 7 |
| $08 - 8 | 8 | 8 |
| $09 - 9 | 9 | 9 |
| $0C - 12 | Standby | Standby |
| $10 - 16 | Volume + | |
| $11 - 17 | Volume - | |
| $12 - 18 | Brightness + | |
| $13 - 19 | Brightness - | |
| $32 - 50 | | Fast Rewind |
| $34 - 52 | | Fast Forward |
| $35 - 53 | | Play |
| $36 - 54 | | Stop |
| $37 - 55 | | Recording |

| | |
|---|---|
| $16 - 22 | SatA |
| $17 - 23 | Recorder2 |
| $18 - 24 | |
| $19 - 25 | |
| $1A - 26 | CDR |
| $1B - 27 | |
| $1C - 28 | |
| $1D - 29 | Lighting |
| $1E - 30 | Lighting |
| $1F - 31 | Phone |

## Other Protocols

Many other protocols exist on the market. Most often it is impossible to find out who has invented the protocol. If I stumble upon one of these obscure protocols I will try to decipher it as far as I can and show it on this page.

I realise that a few big brands are still missing in my descriptions, like Philips RC6, RCMM and B&O. Lack of information and lack of time are the main reasons why I haven't added these protocols to my knowledge base yet.