SUNROM TECHNOLOGIES
Your Source for Embedded Systems

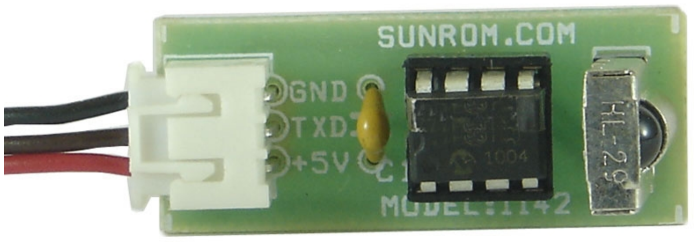
Email: info@sunrom.com

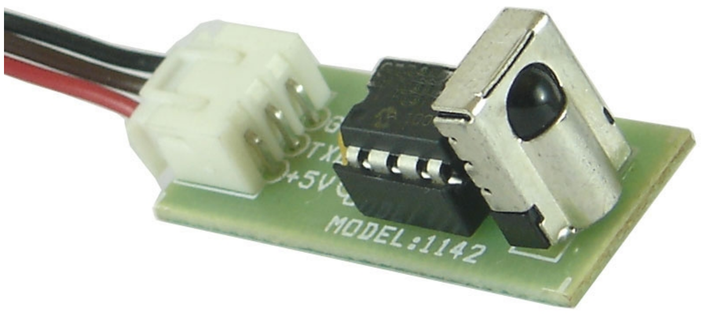
Visit us at http://www.sunrom.com

Document: Datasheet Date: 22-Mar-11 Model #: 1142-NEC Product's Page: www.sunrom.com/p-1045.html

NEC Remote Decoder Serial Out

The NEC remote decoder board is based on the IC ST3679 which decodes the received remote control data and output 16 bit serial data output. The decoded data contains various information like Repeat Command, Address of Remote and Command Key Pressed. This decoded information from transmitter can be used in various ways to make any remote controlled application. The transmitter should be a NEC protocol type normally found in house hold remote controls. Most TV, DVD, AC remotes are either RC5 type or NEC type. Remote of these labels NEC, APEX, HITACHI, PIONEER uses NEC protocol.





Features

- Simple serial data output
- TTL level output compatible with any microcontrollers pins directly
- Decodes any NEC type remote control

Specification

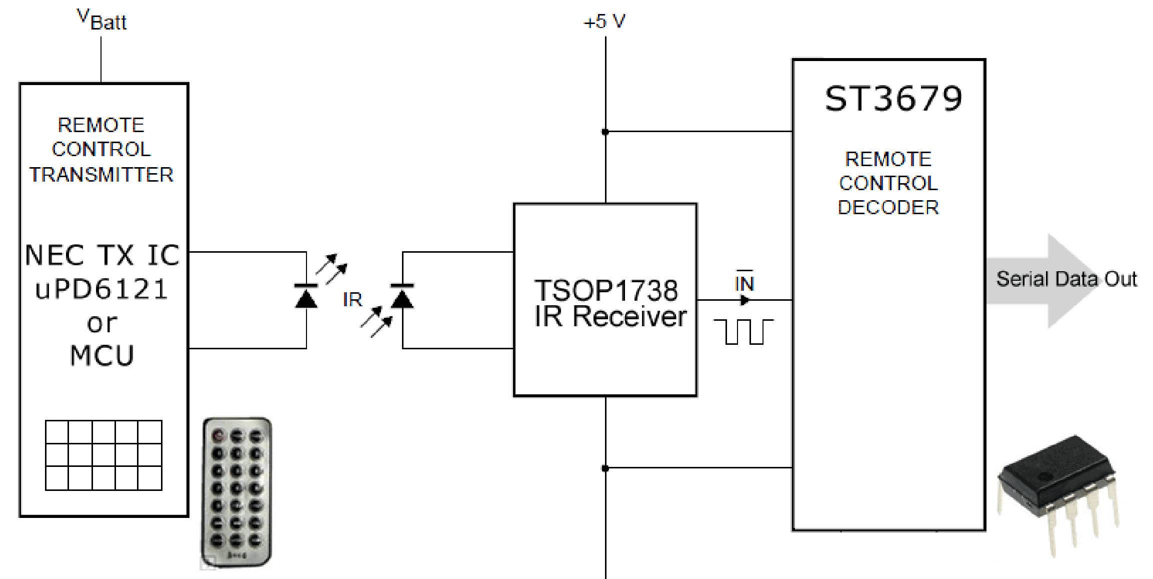
Parameter	Value
Working Voltage	5V DC
Current	10mA
Serial UART Interface	9600 bps, 8 bit data, no parity, 1 stop bit at 5V (For 3V systems insert a 1K resistor in series to TXD to drop the voltage at 3V)
Board Dimensions	36 mm x 15 mm

Data Interface for NEC Decoder

For each output the data is two bytes long containing total 16 bits NEC data. Output of serial data at 9600 bps is particular useful when you have a dedicated serial input pin available on your application microcontroller to get the 2 bytes of data. You can also use the serial data to interface to PC using MAX232 level convertor for serial port or use USB-TTL chip to get a virtual serial port on PC to which many software like Hyperterminal can be connected. Custom software can also be made to monitor the incoming data and develop applications further.

Block Diagram

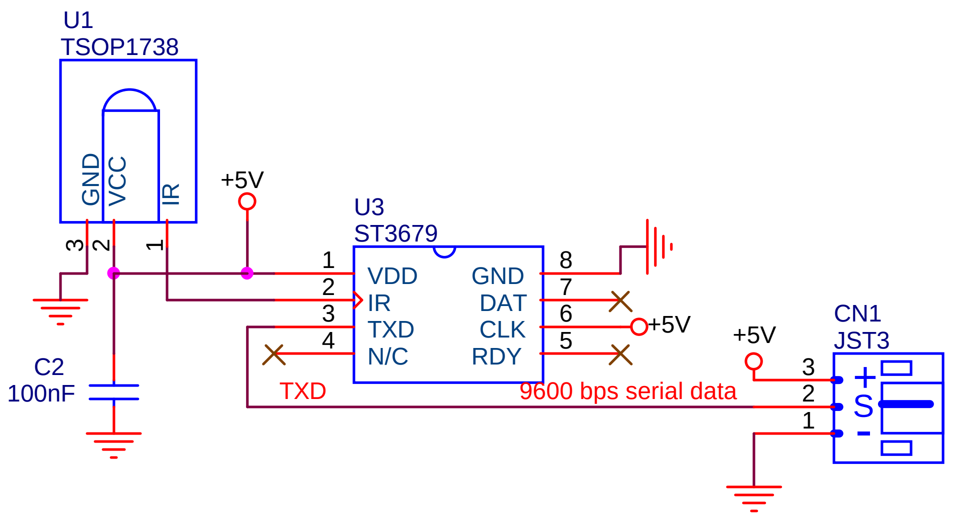
ST3679 needs only one component to work that is IR receiver like TSOP1738 or similar. The data is output as simple 2 bytes of serial data consisting of 16 bits of information for each key press on the remote.



NEC format Remote Control Transmitter
http://www.sunrom.com/p-1022.html

ST3679 - NEC Format Decoder
http://www.sunrom.com/p-1044.html

Board Schematic



Sunrom Technologies	http://www.sunrom.com		
Title	Remote Code Receiver		
Code	1142-NEC	Rev	1
Date:	Wednesday, December 29, 2010	Sheet 1	of 1

Output Data format

Output from ST3679 is in two bytes, thus making total 16 bits of data, let us see meaning of each bit

High Byte - First

Bit Position->	15	14	13	12	11	10	9	8
Value	A7	A6	A5	A4	A3	A2	A1	A0

Low Byte - Second

Bit ->	7	6	5	4	3	2	1	0
Value	D7	D6	D5	D4	D3	D2	D1	D0

A7-A0 = NEC address of remote control. This is also called custom code or device address.
D7-D0 = NEC command for each keypress at remote control.
If you get output as 0x0000 means it's a Repeat command since user has kept key pressed.

Example
For example pressing Key 1 on remote control can output 0x0301 where 0x03 is high byte and 0x01 is low byte. Here 0x03 is address or custom code for the transmitter which remains same for all key press. The data command will change on each key press. Special case of data is repeat command if a key is kept press. It will be output as 0x0000.

If we interpret, in terms of NEC data we get below

High: 0x03 in binary is

Bit ->	15	14	13	12	11	10	9	8
Data->	0	0	1	1	0	0	1	1
	A7	A6	A5	A4	A3	A2	A1	A0

Low: 0x01 in binary is

Bit ->	7	6	5	4	3	2	1	0
Data->	0	0	0	0	0	0	0	1
Value	D7	D6	D5	D4	D3	D2	D1	D0

Important: If you keep any key pressed, you will get 0x0000 which means repeat code. This is part of NEC format when user keeps key pressed, it send repeat command and not the whole data packet. Only on first key press you get actual data, then if key is kept press you get repeat code which is 0x0000. If you leave Key1 and then press again, full value of data/address will be visible. Therefore this data of repeat packet tells you if user is keeping the key press or left the key once and pressed again. This is particular useful if you are implementing Toggle output like Relay ON and OFF logic.

Serial Data Output Format

When you see data output from chip in serial at 9600 baud rate, you will get total six bytes output as each key press in ASCII format so you can view it on screen.

Let us see what data output you will get in serial mode. The last two bytes in serial mode are new line characters so that when you see this data in terminal you can see each new data in new line. If you press Key1 on remote, you will get following output

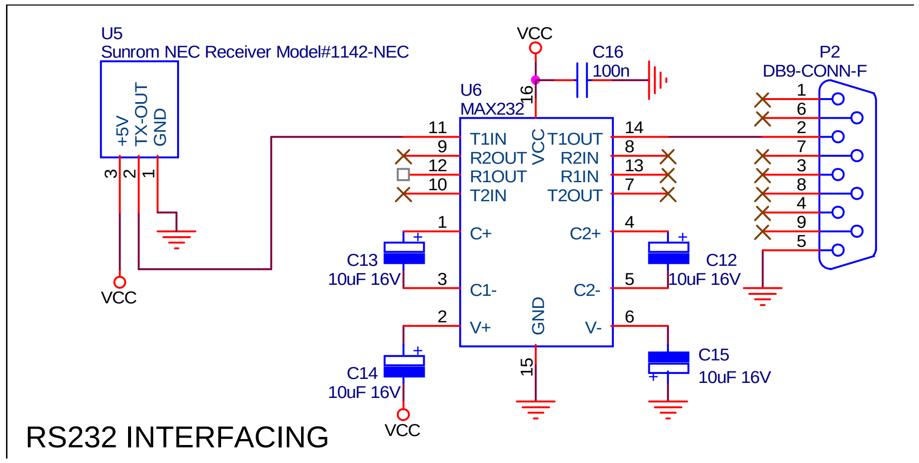
Example Output of Serial in terminal software for Key1:
0301<CR><LF>
Interpreting above result in below table

BYTE COUNT	HEX	DECIMAL	CHARACTER DISPLAYED	Details
1	0x30	4	'0'	Data High Byte
2	0x33	51	'3'	Data High Byte
3	0x30	48	'0'	Data Low Byte
4	0x31	49	'1'	Data Low Byte
5	0x0D	13	'\r' = CR	New Line Character
6	0x0A	10	'\n' = LF	New Line Character

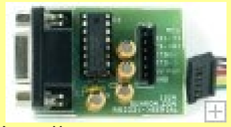
The above values in serial data are ASCII characters. You can convert the value to binary to use in your program by deducting 0x30 from ASCII value. Our sample code given on next page uses this technique to convert this ASCII buffer of four digit to single integer of NEC data variable containing 16 bits.

Interfacing with RS232

If you wish to interface the module with RS232 level like a PC serial port or any other device you need a level convertor such as MAX232 as shown below.

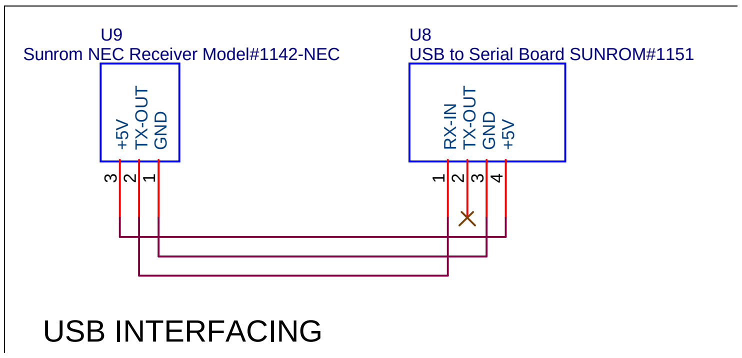


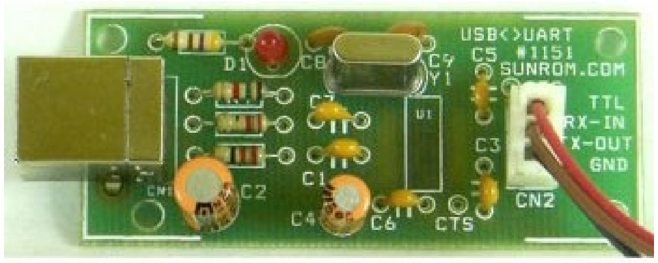
You can also use our Max232 Board Model 1104



http://www.sunrom.com/p-245.html

Interfacing to USB Port and Powering from USB Port

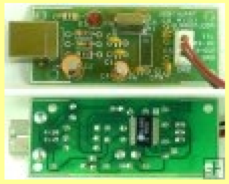

U9
Sunrom NEC Receiver Model#1142-NEC
+5V
TX-OUT
GND
U8
USB to Serial Board SUNROM#1151
RX-IN
TX-OUT
GND
+5V
USB INTERFACING



It will appear as virtual serial port on PC to which you can communicate through any software which can transmit receive by this serial port like hyperterminal or custom made software.

To get +5V power for Decoder from USB port, solder +5V wire of RF module to +ve pin of this capacitor.

You can use our USB to Serial Board Model 1151

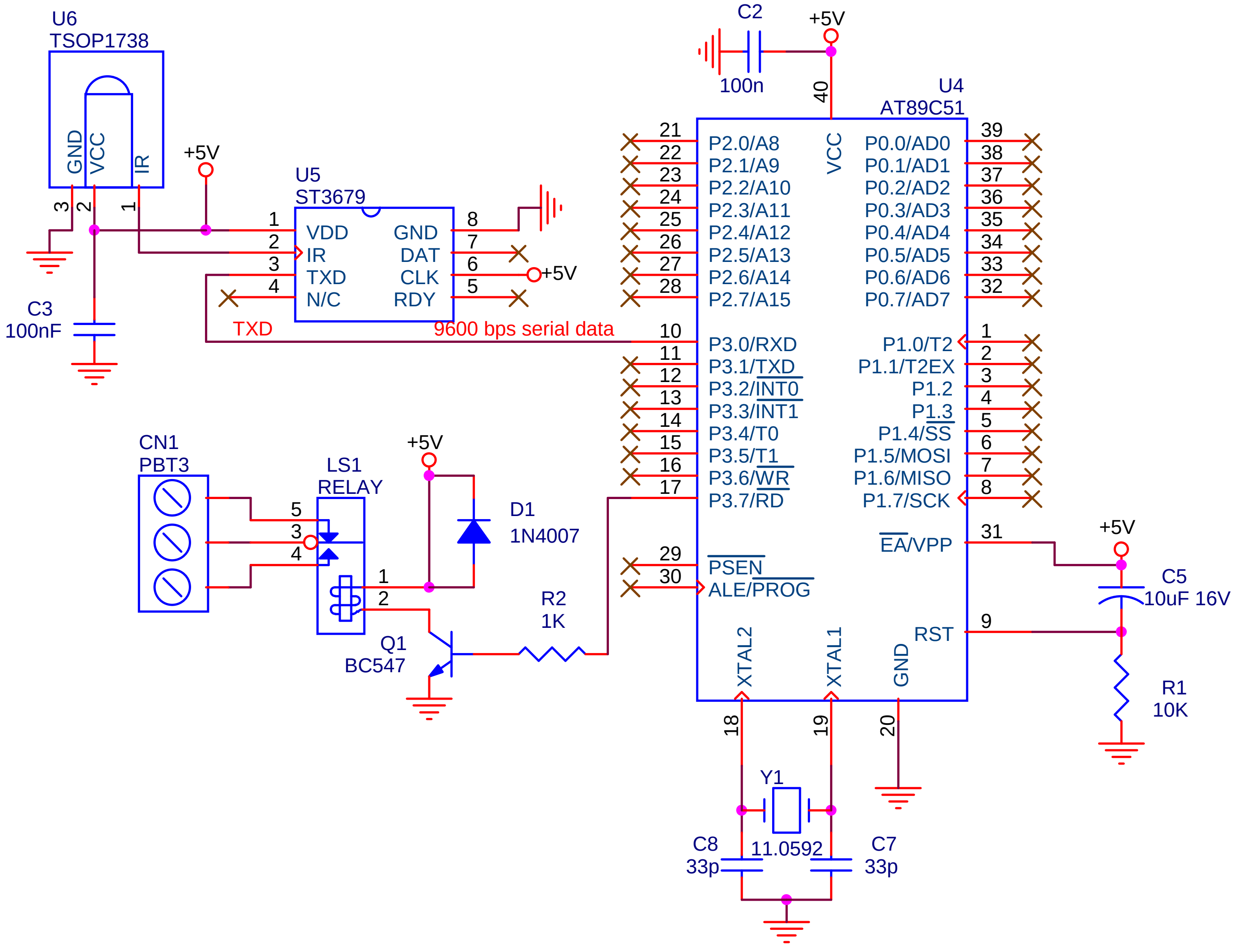


http://www.sunrom.com/p-244.html

Application example using serial data

We have used AT89C51's RXD pin to receive serial data from ST3679 and we switch on Relay ON or OFF for Key1 pressed on Remote. Key1 when first pressed will turn relay ON and Key1 pressed pressed again will make Relay OFF. Relay used is 5V type. But can be any voltage if you have higher voltage available on your application. You can use any microcontroller to interface using this interface. We have chosen AT89C51 to show since it is more widely used. The sample code we have given can be adapted to any C compiler or any microcontrollers like AVR or PIC since with minor changes.

Source code can be downloaded from
http://www.sunrom.com/files/3679-samplecode.zip
Code is compiled using keil compiler



NEC Infrared Transmission Protocol

The NEC IR transmission protocol uses pulse distance encoding of the message bits. Each pulse burst (mark – RC transmitter ON) is 562.5µs in length, at a carrier frequency of 38kHz (26.3µs).

Logical bits are transmitted as follows:

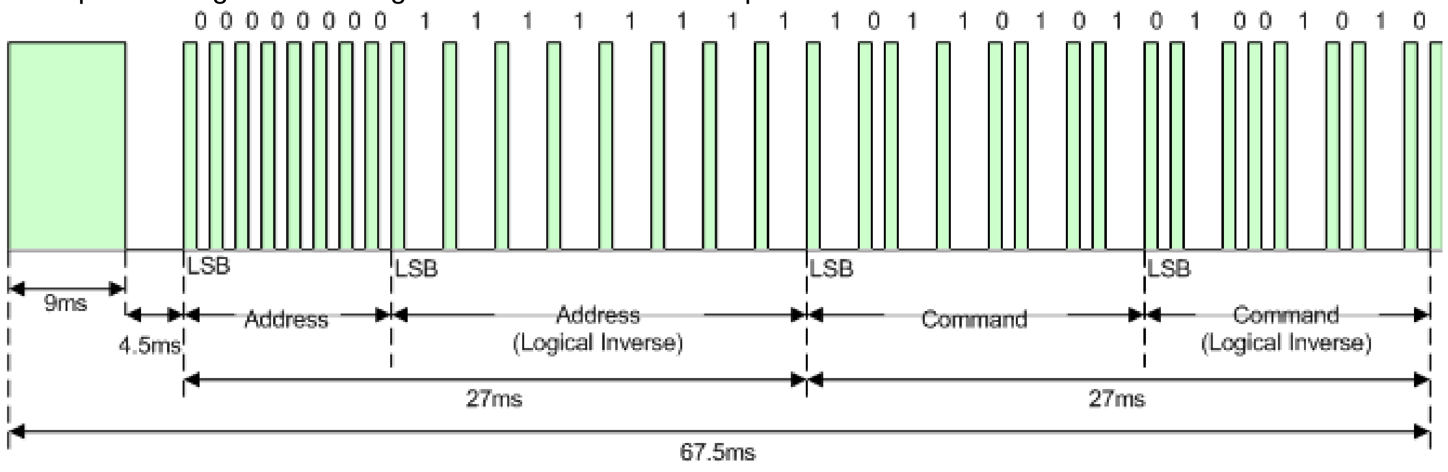
- Logical '0' – a 562.5µs pulse burst followed by a 562.5µs space, with a total transmit time of 1.125ms
- Logical '1' – a 562.5µs pulse burst followed by a 1.6875ms space, with a total transmit time of 2.25ms

When a key is pressed on the remote controller, the message transmitted consists of the following, in order:

- a 9ms leading pulse burst (16 times the pulse burst length used for a logical data bit)
- a 4.5ms space
- the 8-bit address for the receiving device
- the 8-bit logical inverse of the address
- the 8-bit command
- the 8-bit logical inverse of the command
- a final 562.5µs pulse burst to signify the end of message transmission.

The four bytes of data bits are each sent least significant bit first. Figure 1 illustrates the format of an NEC IR transmission frame, for an address of 00h(00000000b) and a command of ADh (10101101b).

Example message frame using the NEC IR transmission protocol.



Notice from Figure 1 that it takes:

- 27ms to transmit both the 16 bits for the address (address + inverse) and the 16 bits for the command (command + inverse). This comes from each of the 16 bit blocks ultimately containing eight '0's and eight '1's - giving (8 * 1.125ms) + (8 * 2.25ms).
- 67.5ms to fully transmit the message frame (discounting the final 562.5µs pulse burst that signifies the end of message).

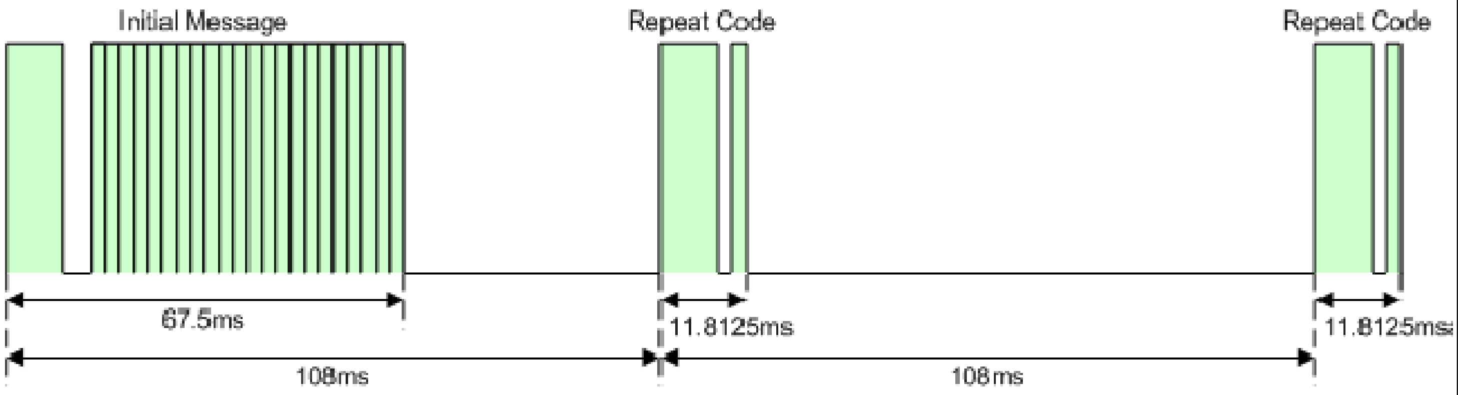
Repeat Codes

If the key on the remote controller is kept depressed, a repeat code will be issued, typically around 40ms after the pulse burst that signified the end of the message.

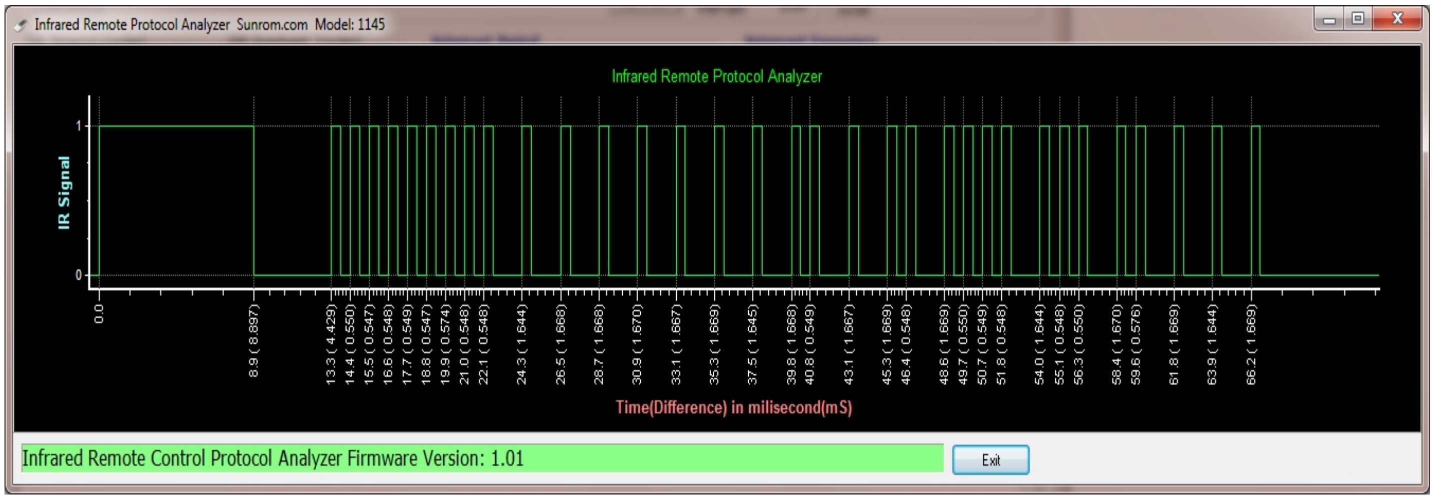
A repeat code will continue to be sent out at 108ms intervals, until the key is finally released. The repeat code consists of the following, in order:

- a 9ms leading pulse burst
- a 2.25ms space
- a 562.5µs pulse burst to mark the end of the space (and hence end of the transmitted repeat code).

Figure illustrates the transmission of two repeat codes after an initial message frame is sent.



Screenshot of actual capture from NEC transmitted data, using Sunrom's IR Protocol Analyzer



Infrared Remote Control Protocol Analyzer Firmware Version: 1.01