

Remote3.0b.bg
a general remote control program
ver 3.0b
1993-01-25
Made by:
Bjorn Gahm
d89-bga@nada.kth.se

INDEX

- 1.1 Disclaimer
- 1.2 Update info
- 1.3 General info
- 1.4 Contents of the directorys

- 2.1 How to use the transmit program
- 2.2 Read a code from a remote control
- 2.3 Contents of an info string
- 2.4 Edit an infostring
- 2.5 Learning a new remote control (how to create a new infostring)

- 3.1 The general remote control program
- 3.2 How to customize the general remote control program

- 4.1 Tips on saving memory
- 4.2 Using alarms

1.1 Disclaimer

WARNING!!!!

Even though this program works fine with my hp48-E I cannot be responsible for memory loss or even hardware damage on your HP48.

I also apologize for any errors in the language since english is not my native language.

1.2 UPDATE INFO

Differences from 2.0:

- * It can now learn new remotes by itself. (Only if the remote uses pulse or space modulation)
- * The length of code to transmit has been increased from 48 to 63 bits
- * The length of pulses and spaces has been increased (the previous versions could not cope with canon codes)
- * The infostring has been changed so old infostrings must be converted.
- * All routines involving displaying info has been changed to display the info in microseconds instead of loops.
- * The TRANSMIT program now cleans the stack, so there is no need for the DROP2 after TRANSMIT that were needed in previous versions
- * There is a general remoteprogram included, to make it easier to customize your hp to be a remote control.

1.3 GENERAL INFO

The only problem is that all the remotes I have tried have different codings. (And I think that there are many more codings that I haven't seen yet).

The types of codings that the program can generate is:

- * Pulse modulated (the length of the pulse is the coding
1 is longer pulse, 0 is shorter)
- * Space modulated (the space between the pulses is the coding)
- * Shift modulated (1 is space+pulse and 0 pulse+space or vice versa).

The code can be reversed.

The codelength can vary between 1 and 63 bits.

An inverted copy can be added to the code.

The number of repetitions can be set between 0 and 15

(it is useful when you want the hp48 transmit a signal as a batch job, since 1 transmission of the code is most of the time not enough for the receiver to recognize)

I have added a read function to read codes from known remotes.

1.4 DIRECTORY CONTENTS

Main directory contains:

TRANSMIT - the transmission program
MAKE - subdirectory for learning remotes (see below)
RC5REMOTE - Calculates the code and transmits (Philips)
GROUP - Holds the group number (used by RC5REMOTE)
OLD - used by RC5REMOTE
RC5 - Philips remote info
MATSU - Panasonic and Technics info
OLDMATSU - Panasonic and Technics info (Older equipment)
UOLDMATSU - Older Pana. and Technics US timing
PION - Pioneer info
JVC - JVC info
SONY - Sony info
SONYCAM - Sony cam info (15bit instead of 12bit datalength)
FUNAI - Funai video info

MAKE directory contains: (MAKE must be a subdirectory of REMOTE)

READ - subdirectory for reading remote codes (see below)
Show - shows all the info from an infostring
HEADER - shows or sets the header data
ONE - shows or sets the 1 data
ZERO - shows or sets the 0 data
LEN - shows or sets the length
SPACE - shows or sets the space between transmissions
REV - toggles if the data is to be reversed
Inv - toggles if an inverted copy is to be added

Mod - toggels of a pulse is to be added after the code
REPS - shows or sets the number of repetitions
MCNV - converts between time and loops for a sampled code
LCNV - converts between time and loops for header one and zero lists
SCNV - converts between time and loops for space
PCNV - converts between time and loops for pulse
S->R
R->S
REVERSE

READ directory contains: (READ must be a subdirectory to MAKE)

LEARN - Reads the code and produces a list with the info needed
the list is stored in L
READIR - Reads a code for the infostring in INFO
RRC5 - read rc5 (Philips)
MKSTRING - Makes a info string of a list (in the L var, produced by LEARN)
Draw - Draws the looks of a sampled signal
INFO - The infostring that are created or used by READIR
L - The list created by LEARN
GETLEN
T
IRSAMP - Mlcode used by READIR
IRLEARN - Mlcode used by LEARN
SAMPLE - the string used by IRSAMP and IRLEARN
PPAR

2.1 HOW TO USE THE TRANSMIT PROGRAM

To transmit a code just put the appropriate infostring on level 2
and the code on level 1 and press TRANSMIT.
(TRANSMIT is located in the main directory, the REMOTE directory)

The infostrings that I have managed to get so far are in the main
directory.

They are Sony, JVC, Panasonic, Technics, Pioneer and Philips.
See the separate listings for details on the different remotes.

Eg to play a sony CD:

Press SONY
Enter #8B2h
Press TRANSMIT

or in a program

<< SONY #8B2h TRANSMIT >>

WARNING WARNING WARNING!!!!

The TRANSMIT program just checks that the data is there but not if it makes
sense. If an invalid string is used the hp48 might transmit for a very long
time. The only way to stop the program then is to press the reset button, since
the program turns off the display and the interrupts.
However it can not destroy any memory because it does not write to the memory

except to the irled.
 OLDER STRINGS CAN MAKE THE HP TO LOCKUP.
 If that is the case then you have to press the reset button under the hp.

2.2 READ A CODE FROM A REMOTE CONTROL

To read a remote control, recall the right infosting and go to the READ directory in the MAKE directory.

Save the infostring in INFO.

Then press READIR (the screen will go blank).

Then hold the remote control close to the irport on the hp (about 2-6cm) and press the right button on the remote.

Now should the screen turn on again and after a couple of seconds you will have a binary on level 1 that is the code.

If the code does not work the try with an another distance between the hp and the remote control.

You might have to try several times.

It might also help to insert new batteries in the remote control.

If you get two binaries on the stack the code is pulsemodulated and only one of the binaries is the right one. (This is the case for sony codes). The reason for this is that the program can't read the length of the last pulse.

(I am trying to solve this problem).

2.3 CONTENTS OF AN INFOSTRING

To check or edit an infostring recall it to the stack and go to MAKE directory.

Press Show to view all the information of the string.

Or press HEADER to see the header info

press ONE to see the 1 info and so on.

(Contents of an infostring see below)

Infostring: (24 chars)

| | | | | | | | | | | | | |
|----|--------|----|-------|----|-------|------|-------|-------|-------|-----|--------|--|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | | |
| | | | | | | | | | | | | |
| | | XX | | XX | | | XX | | XX | | | |
| | LSB | | MSB | | LSB | | MSB | | LSB | | MSB | |
| | Pulse | | Space | | Reps | | Pulse | | Space | | Order | |
| | Header | | | | 1 | | | | | | | |
| | | | | | | | | | | | | |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | |
| | | | | | | | | | | | | |
| | | XX | | XX | | | XX | | | | | |
| | LSB | | MSB | | LSB | | MSB | | LSB | | MSB | |
| | Pulse | | Space | | Order | Data | Trans | Rev | Inv | Add | Repeat | |
| | 0 | | Len | | Space | Data | Data | Pulse | | | | |
| | | | | | | | | | | | | |
| 23 | 24 | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | XX | | | | | | | | | | |
| | LSB | | MSB | | | | | | | | | |

| Space |
| in sample |
| |

Head: Pulse: length of pulse, if length = 0 then there are no header
Space: time to next bit
Reps: 0 = send header always 1-9 send header that number of times

1,0: Pulse: length of pulse
Space: time to next bit
Order: 48 "0" => space pulse
: 49 "1" => pulse space

Data len : length of data in bits

Trans space: specifies the time between the repetitions of the data

Rev Data : 82 "R" => reversed
32 " " => not reversed

Inv Data : 73 "I" => the data is reversed and added to the data -> data+inv(data).
32 " " => ignored

Add Pulse : 65 "A" => a pulse is transmitted after the last bit.
32 " " => ignored

Repeat : "0-?"=> the transmission is repeated (0-15) times if no key is pressed
(if the program is run as a batch job and not manually)

2.4 EDIT AN INFOSTRING

In order to view an infostring or edit it, the string must be on level 1 on the stack.
To view, recall the string to the stack, go to the MAKE directory and press Show.

All the commands that are used to make an infostring work in two ways.
The commands are HEADER, ONE, ZERO, LEN, SPACE, REV, Inv, Mod, REPS.

If the infostring is on level 1 on the stack then the commands will recall the data.

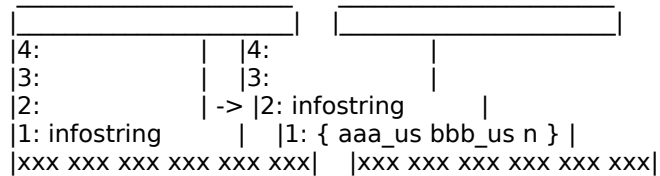
If the infostring is on level 2 and the data is on level 1 then the commands will alter the infostring and leave it on level 1.

To alter a infostring:
Put on stack:
Level 2: infostring
Level 1: data to change to
Press HEADER or ONE or ZERO or LEN

Read infostring:
HEADER ONE ZERO programs:

If the infostring is on level 1 on the stack then HEADER or ONE or ZERO will recall the info for the header one or zero respectively.

Eg.
HEADER Will produce:



Where aaa is the time for the pulse, bbb is the time for the space and n is the number of repetitions of the header.

For sony string the list will be: { 2192_us 592_us 0 }

For ONE and ZERO n will be the order of transmission.

n=1 -> transmits pulse space

n=0 -> transmits space pulse

LEN:

Sets the length of the code.

2: infostring
1: real number

Just put the number of bits as a real and press LEN.

Eg.
2: infostring
1: 12
LEN

Will set the code length to 12 bits.

SPACE:

Sets the space between the transmissions.

2: infostring
1: real number_us

Just put the length of the space as a real and press SPACE.

Eg.
2: infostring
1: 23000_us
SPACE

Will set the space between the transmissions to approx 23ms.

Rev:

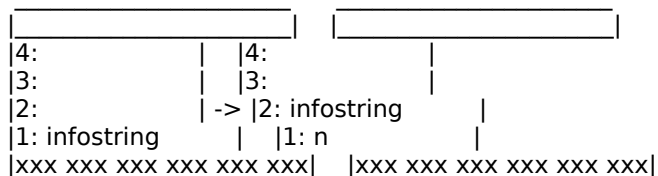
Sets if the code is to be reversed before transmission.
(Sets the transmission order, MSB first or LSB first)

Normally you don't have to change this parameter, but

it could make the codes more understandable if you change the transmission order.

Eg to recall data.

Press Rev :



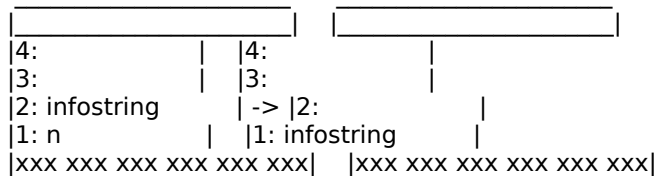
Where n is 1 or 0.

0 means transmit LSB first. (LSB = Least Significant Bit)

1 means transmit MSB first. (MSB = Most Significant Bit)

To alter put the number on level 1 and infostring on level 2:

Press Rev :



Inv:

Works as Rev, but

0 means don't add an inverted copy of the code

1 means add an inverted copy of the code after the code (used by old panasonic and technics remotes)

Mod:

Same as above, but

0 means don't add a pulse after the code (used if the code is pulse modulated)

1 means add a pulse after the code (used if the code is space modulated)

2.5 LEARNING A NEW REMOTE

This part of the code is the newest code in this posting.

It is not functioning good at all times.

But with patience you should manage to learn new remotes.

To learn a new remote, go to the READ directory in the MAKE directory.

There you will find LEARN.

LEARN is the program to generate data to make a infostring.

Press LEARN and when the screen goes blank hold the remote control as close as possible to the HP's irport (it might help to take the cover off).

Press any button on the remote until the screen goes back on again.

Then wait a while (about a half minute).

Now you will have a list on level 1.

The list will consist of:
{ n (h1,h2) (p1,p2) (s1,s2) (sp1,sp2) m }

Where:

n is the length of the code in bits.

(h1,h2) is the length of the header (of there are no header then this will be missing).

(p1,p2) is the length of one codeing

(s1,s2) is the length of zero coding

(sp1,sp2) is the length between transmissions

m is the coding type 1 means space modulated, 0 means pulse modulated

If there are more than 4 complex numbers in the list then the sample was no good. Then you have to try again.

I know that Grundig and Bang&Olufsen will not work. And I have not got Yamaha to work either (though it should work).

I have tried it on JVC, Sony, Funai and Canon with successful results.

When you have a list that seems right then press MKSTR.
now you will have a string that ought to work.

The list will be stored in L and the infostring will be stored in INFO.
Recall INFO and store it under a new name in the REMOTE directory
(Use the name of the brand).

Now you can do as in chapter 2.1 to learn codes.

If the infostring seems right but the codes doesnot work then the length of the pulses might be too short.

Try to make the pulses a bit longer by altering them mauully. (using HEADER, ONE and ZERO in the MAKE directory)

I am trying to fix this timing problem.

3.1 The general remote control program

If you press CST in the REMOTE directory, the custom menu will be the selector between different remotes.

Pressing one of the menu keys will customize the HP to be a remote.

The remotes in the program right now is the remote I use but the program can be altered to use other remores instead. (See next chapter)

The remotes I use is: Sony tapedeck, Sony CD, JVC Video, Philips and Sony videocamera & TV and Panasonic video.

(I work weekends in a Radio&TV shop).

Eg. Press VIDEO.

Now the display will show 8 meny rows on the screen, and only one of them is normal (all the others is inverted).

```
##REW## #PLAY## ##FF### #STOP## #PAUSE# ##REC##  
OPERA CH- CH+ SLO- SLO+  
PRGM CANC GOTO DISP CRES SP/LP  
AMON EJECT
```


By pressing Arrow up or Arrow down you can switch to the other menu rows.
The row that is normal is using the keys A-F to transmit the code.

To play the video press B.
To turn on the video press Arrow Down and then A.

To end the remote program just press ON.
Blue shift right and ON will switch off the HP (as normal) without leaving
the remote program.

Don't press any keys with the shift keys (orange shift left, blue shift right).
If you do the the program will let you alter the keys. (Read more about that
in the next chapter.

3.2 How to customize the program

To add a new remote to this program, make a copy of one of the directories
(take the VIDEO directory or the PVIDEO directory if the code that you are
going to use is longer than 16 bits)
Then edit the CST var in REMOTE directory to add the new remote.
Just look at how the other choices are made in the CST var.

In every remote directory there must be:
The CODE var that holds the name of the infostring that is to be used.
The MD var that holds the list to construct the menus (to begin from scratch
store a empty list in MD).
The DATA var that contains the codes (you don't have to edit this var).

If you are to use codes longer than 16 bits then use the REM program in PVIDEO
directory. You have to alter the prefix code in the REM program.
Make shure that you store the REM progam in the right directory and NOT in
the REMOTE directory, because there is a default REM program there.

Or alter the directories that exists.

To enter new commands just select the right menurow and press orange shift left
and the right key.

If it is one of the menukeys you will be able to alter the command name first
and then alter the code.
If you don't want to alter the command name just press ENTER, otherwise
alter the name and then press ENTER.

If it is the ENTER key row and down you will only be able to alter the keys
code.

If you press blue shift right the HP will sample the new code.
The program can only cope with codes with 16 bits so if you use this
alternative you may have to trim the code (the same applies to codes that
uses pulse modulation).
Don't use this option unless you really know now the codes works.

4.1 Tips on saving memory

If you don't have to learn new codes or new remotes, then you can purge the MAKE directory. (MAKE takes approx 8kbytes of memory).

Save the MAKE directory separate for easier downloading in the HP later. Save it in binary mode (it has some mlcode in it).

Just make shure to store the MAKE directory in the REMOTE directory when loaded back in the HP, otherwise it will not work properly.

Every remote will take up approx 400-500 bytes.
The REMOTE directory will take up approx 5500 - 6000bytes (depending on how many remotes that is in the system).

4.2 Using alarms

I use the HP to wake me up in the morning.

It switches on the video on MTV.

To do this there is a small program in the REMOTE directory that is called ALARM.

Use this a a executable when creating a alarm.

Just recall the program to the stack (by pressing blue shift right and ALARM).

And set the alarm in the TIME menu.

To use the alarm with other codes just alter the ALARM program before recalling it to the stack and setting the alarm.

Have fun.

If you successfully learns new remotes please mail me the new code so that I can keep a database of remotes so that others don't have to sample new remotes more than they have to.

Bjorn Gahm d89-bga@nada.kth.se
Forskarbacken 10/315
104 05 STOCKHOLM
SWEDEN