BioControl

HANDHELD READER HHR 3000 PRO V2

HHR 3000 PRO
Programmers Manual
Functions description

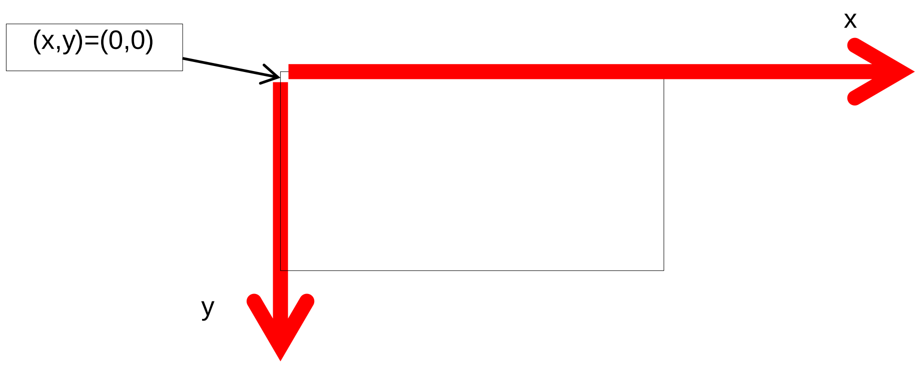
This paper is fully compatible from 5.40 versions of HHR Manager and Operating System

Screen Area functions

PrintText(x,y,font,text)

PrintText functions simply prints text on HHR's LCD, the parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).



HHR's LCD dimensions.

- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- text – the text which you want to appear on HHR's LCD, according to function syntax you have to take the text in quotation marks.

Example:

```
PrintText(0,0,0,"TABLE: ESPANA 01BT")
```

According to example above text will be displayed starting with top-left point of screen; (x,y) represent top-left point of text or object.

PrintField(x,y,font,column_name)

PrintField functions display on HHR's LCD a pointed column at current record; the parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column name – name of column defined in TABLE section, the size of letters is important.

Example:

```
PrintField(4,18,2,FARM)
```

PrintExp(x,y,font,expression)

PrintExp functions display on HHR's LCD expression, the expression is a result of a few functions which will be described below; the parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- The expression – the expression as it was already mentioned is a result of a few functions, those functions are:

count(column_name|*)

Count() expression counts all records according to parameter, this parameter might be a "*" , that means that HHR will count all records in Table. You can put any column name as parameter then HHR will count all record where value exist in indicated column.

incCount(column_name)

incCount() expression counts how many values is in Table in pointed column, the function doesn't care about value, it only check if there is any value. The value which will be displayed will be incremented with 1.

date()

date() expression returns a current date according to data format from HEADER section.

time()

time() expression returns a current time according to time format from HEADER section.

noofrec(db_ident)

noofrec() expression returns a number of all records in Table or Log, noofrec(number of records). db_ident parameter describes to which table in HHR you want to refer a main Table or to Log. If you want to refer to main table use "table", if you want to refer to Log table use "log".

mFull()

mFull() expression returns the amount of unfree memory in HHR's EEPROM.

Empty()

mEmpty() expression returns the amount of free memory in HHR's EEPROM.

amountRecJoined()

amountRecJoined() returns amount of joined record from LOG to current record in TABLE.

numberRecJoined()

numberRecJoined() returns a number of joined record from LOG to current record in TABLE.

dec(column_name,start,len)

dec() returns the number of transponder in decimal format.

tagtype()

tagtype() returns the type of transponder as the 'HEX' or 'FDX'

global()

global() expression returns the value of global. The result of this expression will be placed at current record, at column_name pointed in function.

countequ(column_name, global)

countequ() returns the number of fields with a value equal to the global.

Example:

```
PrintExp(46,0,0,count(*))
PrintExp(12,2,5,count(BREED))
PrintExp(26,0,0,date())
PrintExp(46,0,2,mEmpty())
PrintExp(46,0,2,noofrec(table))
PrintExp(0,0,2,incCount(WEIGHT))
```

There are also a Transponder functions included to HDS, those functions allows you to select an interesting part of transponder number, and display it. To remind you what information are included in a number look at this table(ISO standard).

Transp. digit	***Information***
1	Animal Mark
2	Retagging
3	User Info
4	
5	Reserved
6	
7	Additional Info
8	Country Number
9	
10	
11	
12	Animal Number
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	

The Transponder functions:

PrintAnimalMark(x,y,font,column_name)

PrintRetagging(x,y,font,column_name)

PrintUserInfo(x,y,font,column_name)

PrintReserved(x,y,font,column_name)

PrintAddInfo(x,y,font,column_name)

PrintCountryNumber(x,y,font,column_name)

PrintAnimalNo(x,y,font,column_name)

PrintCountryCode(x,y,font,column_name)

PrintCountryName(x,y,font,column_name)

All those function have the same parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- Column_name – this parameter describes form which column data will be displayed, pointed column has to be Transponder read-type.

Two last functions PrintCountryCode and PrintCountryName hasn't got a clear interpretation at Transponder number, a values displayed by those functions are modification of Country number which is included in Transponder information. Those two functions were included to HDS to make you work easier with HHR.

PrintGlobal(x,y,font,global_no)

PrintGlobal functions returns a value of global defined in Global section. The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- global_no – a number of global which you want to be displayed on HHR's LCD

Example:

```
PrintGlobal(4,46,2,1)
```

DrawLine(x_1,y_1,x_2,y_2)

DrawLine function draws a line on HHR's LCD according to coordinates given in parameters where x_1 and y_1

is first point and x_2, y_2 is the second point.

Example:

```
DrawLine(12,32,12,45)
```

DrawRec(x_1,y_1,x_2,y_2)

DrawRec function draws a rectangle on HHR's LCD according to coordinates given in parameters where x_1 and y_1 is first corner and x_2, y_2 is the opposite corner.

Example:

```
DrawRec(12,32,12,45)
```

There will be presented a functions which helps you to edit your TABLE and LOG at current record, all those functions have the same parameters which will be explained after describing a functions.

EditMaskedField(x,y,font,column_name)

This function allows you edit a value in current record at pointed column, the column must include string values (string type S).

EditReadField(x,y,font,column_name)

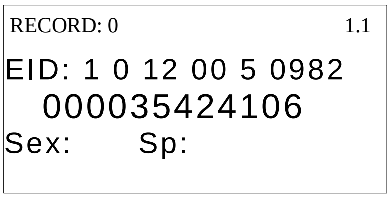
This function allows you to read a transponder number (by pressing Read Transponder button) and place it in current record at pointed column(field), the column must be a Transponder Read type(R)

EditReadFieldExp(x,y,font,column_name)

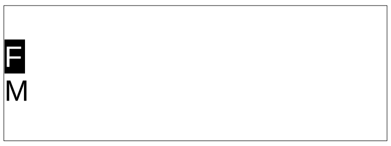
This function allows you to read a transponder number (by pressing Read Transponder button) and place it in current record at pointed column(field), the column must be a Transponder Read type(R) in addition it executes Action area.

EditChoiceField(x,y,font,column_name)

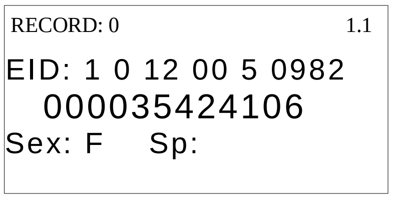
This function allows to edit a Option type column at current record, the column must be a Option type.


```
RECORD: 0                     1.1
EID: 1 0 12 00 5 0982
  000035424106
Sex:      Sp:
```

After you put a frame on a EditChoiceField function and press any digit button a screen appear with all available options for the column pointed in function.


```
F
M
```

After you select an appropriate for you value you will come back to the screen with a fulfilled field in Table and frame on the screen.


```
RECORD: 0                    1.1
EID: 1 0 12 00 5 0982
  000035424106
Sex: F    Sp:
```

EditChoiceFast(x,y,font,column_name)

This function allows to edit values in Option type column with numerical keys, when cursor is on the field and user press '1' then first element from the list is selected and saved in DB. User can select 1-9 values(basing on order from column definition). To erase value in the field user must press '0'.

EditGlobalField(x,y,font,column_name)

This function allows to edit a Global type column at current record, the column naturally has to be a Global type. During editing a Global value you put only a number of Global not a value.

EditDataField(x,y,font,column_name)

This function allows to edit a Data type column at current record, the column naturally has to be a Data type.

EditTimeField(x,y,font,column_name)

This function allows to edit a Time type column at current record, the column naturally has to be a Time type.

The parameters for those functions are similar to prior functions:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column_name – this parameter describes which column will be used, using rules which were described for each function.

Now, when you have quite a bit range of functions we can design some screens. Lets assume that we have a non-empty User input database so that after turning on HHR we will be working on some data because first record won't be empty.

Edit functions allow to edit database field. After selecting the field on LCD screen a frame will appear on it and then you can edit the value defined at edit-function.

```
TABLE

     SPECIE;O;0;1;1;Ovino,Caprino,;;
     BREED;S;0;1;1;00;00;
     SEX;O;0;1;1;H,M,;M;
     BORN_DATE;D;0;1;1;0000000000;;
     FARM;G;0;1;1;##;0;
     ID_DATE;D;0;1;1;0000000000;;
```

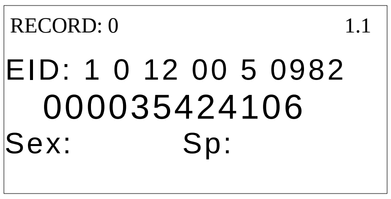
```
  C_ID_ELEC;R;1;1;1;0000 0000 0000;;
  ID_TYPE;S;0;1;1;0;2;
  VET;G;0;1;1;##;1;
  RETAG;S;0;1;1;0;0;
  MESS;M;0;0;0;0;;
END

//--------Macro definition fragments-----------
//A Screen

begin_screen
 PrintText(0,0,0,"RECORDS:")           }
  PrintExp(46,0,0,count(*))            }   A
  PrintText(115,0,0,"1.2")             }

  PrintText(0,7,1,"Born:")             }   B
  EditDateField(46,7,1,BORN_DATE)      }

  PrintText(0,22,1,"Id. date:")        }   C
  EditDateField(46,21,1,ID_DATE)       }

  PrintText(0,37,1,"Id. type:")        }   D
  EditMaskedField(46,36,1,ID_TYPE)     }

  PrintText(63,37,1,"Breed:")          }   E
  EditMaskedField(100,36,1,BREED)      }

  PrintText(47,36,1,"Sp.:")            }   F
  EditChoiceField(69,35,1,SPECIE)      }
end_screen
```

This example of App Program we can share 6 fragments,

“A” fragment is only a information how many records is in the database(`PrintExp(46,0,0,count(*))`) and some more user info (last function at “A”)

The “B” fragment is responsible for editing a BORN_DATE column at current record, the first function is user information and the second one allows user to enter the data. As you see at “B” fragment EditDateField function is used, this function operates only on Data type column and according to TABLE section BORN_DATE is data type.

The “C” fragment is similar to “B” one, during design you must remember that after editing any field (any type) the value appear on the screen so that it take place according to declared font, so you have to pay attention to collision text not happen. A screen below is example of such effect.



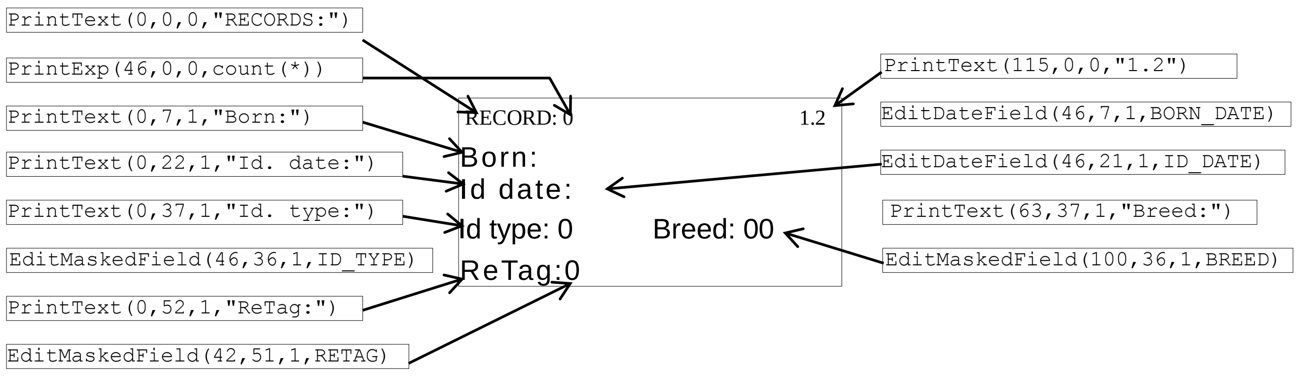
```
RECORD: 0                    1.1
EID: 1 0 12 00 5 0982
  000035424106
Sex:         Sp:
```

According to screen above the “Female” value placed in a frame is too big because it collide with static text “Sp:”, so you have to predict such situation and avoid it. To help you predict it you can use information about mask or value which will be placed there, for example if the value is string type you can count characters according mask, if it is a date or time, date and time settings are in header section co you can predict whenever it will be 6, 8 or 10 characters, if a EditGlobalFiled will be used there are only 2 characters in this filed because Global type column take only two characters.

The “D” and “E” fragments are similar, both of them use EditMaskedField function which allows to edit a String type column, according to mask defined in Table section

The “F” fragment edits a Option type field with EditChoiceField function, the action which will be taken at this function were described before after editing a EditChoiceField function.

During design a screens you must remember about easiness of work with your Application Program and screens directly. Other important thing is “putting a frame” mentioned above, all the Edit function modify or put a value, there is no logic option to modify all the fields simultaneously so HHR has the frame which points what is editing now(which operation element), at the beginning of each screen the frame appears on Edit function which is placed first in code(at first operation element), if you want to move to next operation element and edit another field(a column in current record) you have to press Enter key on keyboard, the keyboard rules were mentioned at the beginning of this chapter. To make HHR's user work without any problems you should make labels next to (or before) the operation elements(frame).



```
PrintText(0,0,0,"RECORDS:")
PrintExp(46,0,0,count(*))
PrintText(0,7,1,"Born:")
PrintText(0,22,1,"Id. date:")
PrintText(0,37,1,"Id. type:")
EditMaskedField(46,36,1,ID_TYPE)
PrintText(0,52,1,"ReTag:")
EditMaskedField(42,51,1,RETAG)
PrintText(115,0,0,"1.2")
EditDateField(46,7,1,BORN_DATE)
EditDateField(46,21,1,ID_DATE)
PrintText(63,37,1,"Breed:")
EditMaskedField(100,36,1,BREED)
```

The reason why `EditDateField(46,21,1,ID_DATE)` arrow point nothing is that ID_DATE has no default value declared in TABLE section while columns like BREED or RETAG have default value.

! Beside Edit function there are also another function which generate operation elements, those functions are IconDelete, IconAddRec which will be described below.

EditGlobal(x,y,font,global_number,global_mask)

This function allows to to change a global value during work with HHR according to global mask, there is no need to reprogram your HHR with Application Manager. The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- global_number – its a number of global which you want to be changed.
- global_mask – its the mask according which user can change a global value, mask was described at Table section.

Example:

```
EditGlobal(4,18,2,0,&& &&&& &&&& &&&&)
```

EditCounter(x,y,font,counter_no)

This function allows user to set a limit for a counter, there are 4 counters available in HHR. This function cooperates tightly with Beep(column_name,counter_no) function which makes HHR beep when amount of values in pointed column is bigger or equal than value of counter_no pointed as parameter.

As a result of EditCounter function an operation element will appear(input box) where user can set the limit of counter pointed as parameter.

If user wants to turn off the counter and beeping of HHR, set value in the field on the screen as 0.

- describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- counter_no – number of editing counter, available values: 0-3.

Example:

```
EditCounter(4,18,2,0)
```

EditSeekField(x,y,font,column_name)

EditSeekField function seeks for placed value in operation element(input box) at pointed column, and moves to the record where such value has been found. If the value wouldn't be found current record number won't be changed.

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column name – name of column defined in TABLE or LOG section, the size of letters is important.

Example:

```
EditSeekField(4,18,2,FARM)
```

EditPartField(x,y,font,column_name*,start, lenght)*

This function allows to edit part of column: S - String, A - Any type, T - Transponder read, Z - Leading Zero, at current record.

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column name – name of column defined in TABLE or LOG section, the size of letters is important.
- start - The edited string will start at the *start*'th position in *string*, counting from zero. For instance, in the string '*ABCDEFG*', the character at position *0* is '*A”*, the character at position *2* is

'*C*', and so forth.

- length - the string returned will contain at most *`length`* characters beginning from *`start`*

Example:

```
10200000999271234567890
EditPartField(4,18,2,EID,8,3)

999 => 234

10200000234271234567890
```

IconDelete(x,y,db_ident,operation_el)

This function allows you to delete a current record, this function isn't realized like others like a frame, using this function you will make HHR display delete icon (-) a minus. After user delete a current record, current record number will take a 1st – first record in database.

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- db_ident – describe whenever user refer to TABLE or LOG. As this parameter you can use “table” or “log”, as identifier to which database you refer to. When you use “log” here without declaring it in HEADER section and LOG sections is pointless. Log section is explained in appendix C
- operation_el – describe to which operation element user moves:
  1. operation_el=0 user stays at current element after the function finish it task.
  2. operation_el=1 user moves to previous operation element(the one which is before IconDelete) declared in App Program after the function finish it task.
  3. operation_el=2 user moves to next operation element(the one which is after IconDelete) declared in App Program after the function finish it task.

Example:

```
IconDelete(3,51,table,0)
```

IconDeleteRel(x,y,db_ident,operation_el)

This function allows you to delete a current record and records related to this one from LOG or TABLE, using this function you will make HHR display delete related icon ($-_R$) a minus and “R” in subscript. After user delete a current record, current record number will take a 1st – first record in database.

This function will delete records related to current record only if tables has a relation defined with Join function. The function will delete current record from TABLE or LOG, and records related(with Join function) to this one from other Table.

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- db_ident – describe whenever user refer to TABLE or LOG. As this parameter you can use “table” or “log”, as identifier to which database you refer to. When you use “log” here without declaring it in HEADER section and LOG sections is pointless. Log section is explained in appendix C. Note: When you set “table” parameter the current record in the table will be deleted. When you use “log” parameter only records from log will be deleted.
- operation_el – describe to which operation element user moves:

1. operation_el=0 user stays at current element after the function finish it task.
2. operation_el=1 user moves to previous operation element(the one which is before IconDelete) declared in App Program after the function finish it task.
3. operation_el=2 user moves to next operation element(the one which is after IconDelete) declared in App Program after the function finish it task.

Example:

```
IconDeleteRel(3,51,table,0)
```

IconFindCol(x,y,column_name,operation_el)

This function allows user to find a record with a required value, the column in which an user will be looking is defined in function parameters. This function is realized on a HHR's screen as a magnifying glass. After user select a value HHR will change current record number to user selected record number.

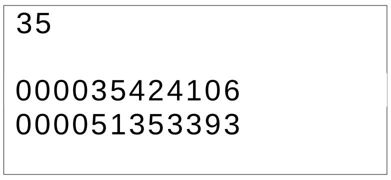
- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- column_name – describe a column in which an user will be looking for data.
- operation_el – describe to which operation element user moves:
  1. operation_el=0 user stays at current element after the function finish it task.
  2. operation_el=1 user moves to previous operation element(the one which is before IconFindCol) declared in App Program after the function finish it task.
  3. operation_el=2 user moves to next operation element(the one which is after IconFindCol) declared in App Program after the function finish it task.

Example:

```
IconFindCol(35,51,C_ID_ELEC,0)
```

The user interface for this function is a little bit complicated. **After pressing any digit on selected icon** a screen appears where user can write a sequence of characters, user doesn't have to write a whole value which he is looking for, he can write part of it and HHR will find all records where such characters are.

HHR is looking for string which exists in column given in function parameters.



```
35

000035424106
000051353393
```

IconFindCol screen

Pay special attention that a string “35” exist both in first and second values, the values where string “35” doesn't exist aren't displayed at HHR's LCD

If you are using **Speed Mode** you must be aware that below icons are available only after long pressing the arrow up or down

- IconDelete
- IconAddRec
- IconFindCol
- IconConfirm
- AddRecConf

IconList(x,y,column_name,operation_el)

This function allows user to find a record. The search results are presented in a list of results and user can scroll down and up to find the record. The column in which an user will be looking is defined in function parameters. This function is realized on a HHR's screen as a magnifying glass. After user select a value HHR will change current record number to user selected record number.

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- column_name – describe a column in which an user will be looking for data.
- operation_el – describe to which operation element user moves:
  1. operation_el=0 user stays at current element after the function finish it task.
  2. operation_el=1 user moves to previous operation element(the one which is before IconList) declared in App Program after the function finish it task.
  3. operation_el=2 user moves to next operation element(the one which is after IconList) declared in App Program after the function finish it task.

Example:

```
IconList(108,51,EID,0)
```

SetDate(x,y,font)

SetTime(x,y,font)

Those functions allows you to edit a date and time in HHR. The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetTime(40,9,1)
SetDate(40,24,1)
```

SetPin(x,y,font)

This function allows you set your Pin code which will be required to turn on your HHR, this enables a minimum of safety in your HHR. The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetPin(55,42,1)
```

SetBTName(x,y,font)

This function allows you to edit a HHR name under which HHR will be visible during BlueTooth connection.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetBTName(43,26,1)
```

SetBTOnOff(x,y,font)

This function allows user to turn on or off BlueTooth module, it can be used to transfer data to PC or other device which manage BlueTooth protocol. To make BlueTooth module work you have to enable BlueTooth module in Header section (BT_true).

If there is a BT_true in Header and user turned on BlueTooth module with SetBTOnOff function a frame with transponder number, date and time is send to connected device.

If there is a BT_true in Header and user turned off BlueTooth module with SetBTOnOff function a frame with transponder number, date and time is send to PC through USB port.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetBTOnOff(67,10,1)
```

SetBTPin(x,y,font)

This function allows user to set a BlueTooth Pin which is necessary to connect HHR with PC through BlueTooth protocol.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetBTPin(67,10,1)
```

SetBTMode(x,y,font)

This function allows user to set a BlueTooth Mode which is option field AUTO or SLAVE. In AUTO mode HHR works as simple master device and can initialize Bluetooth connection. User have to introduce the MAC address of search device e.g printer. See also SetBTMAC() function.

In the second mode HHR works as SLAVE so a host device e.g. PC is needed to establish bluetooth connection.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetBTMode(67,10,1)
```

SetBTMAC(x,y,font)

This function allows user to set a MAC address of search device e.g printer when HHR works as simple host device.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetBTMAC(33,38,1)
```

IconBT(x,y), IconRS(x,y), IconGSM(x,y)

Those functions allows user to put an inform icon on HHR's LCD, the icon will be informing user whenever HHR is connected to other device through BlueTooth; RS232 port is opened or GSM module was initialized(HHR is connected to GSM Network). If HHR is connected a icon is black otherwise icon is white.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).

Example:

```
IconBT(67,10)
```

```
IconRS(27,40)
IconGSM(17,40)
```

ArrowUp(x,y)

This function draws Up Arrow at screen and pointed in parameters x and y. Parameters are explained upper.

Example:

```
ArrowUp(25,12)
```

ArrowDown(x,y)

This function draws Down Arrow at screen and pointed in parameters x and y. Parameters are explained upper.

Example:

```
ArrowDown(25,12)
```

DelLog()

DelLog function deletes LOG, it is accessible from Screen Area so, you can create a special menu that will be deleting Log.

Example:

```
DelLog()
```

DelTable()

DelTable function deletes TABLE, it is accessible from Screen Area so, you can create a special menu that will be deleting Table.

Example:

```
DelTable()
```

Continue(c_param)

This function turns on continue reading of transponder after user enters macro where Continue function has been used. It is possible to store data in memory or send it with USB or BT, RS232 modules.

This function has optional parameter(c_param) which defines target for storing read transponder numbers with Date and Time stamp, it is also possible to not store that information, then HHR would inform user about successful read of tag with LEDS and beeper. Despite the parameter of this function, in all cases HHR will notify user about successful reading of tag with LEDS and beeper.

User can break continuous reading with long press of power key or read transponder button.

If no data(value) is placed as parameter for this function HHR saves all read transponder numbers to LOG, HHR also saves current time and date to LOG if such columns exists in LOG. HHR will put data to first found columns that can take Transponder number(column type:R) Date(Date type column) and Time(Time type column).

The values of c_param:

- NULL – non data will be stored in memory or send with any external interface, HHR will notify user about successful reading of tag with LEDS and beeper;
- TABLE – HHR will store Transponder number, Date and Time to first found columns(of appropriate type) in TABLE;
- LOG – HHR will store Transponder number, Date and Time to first found columns(of appropriate type) in LOG;

- TX – HHR will send Transponder number and Data Time stamp according to frame defined in Header section. HHR will use one of available and turned on interfaces: RS232, BlueTooth or USB. Please remember to turn the module before using Continue function with TX parameter;

If none of interfaces will be turned on(user won't turn on RS232, or BlueTooth) HHR will send data with USB interface

To enable communication with any external device and HHR please use BT_true indicator in Header Section. Possible combinations:

- TABLE|TX – saving data in TABLE and sending it with RS, USB or BT
- LOG|TX – saving data in LOG and sending it with RS, USB or BT
- TX – sending data with one of interfaces: RS232, BlueTooth, USB

Example:

```
Continue()
Continue(TX)
Continue(TABLE)
Continue(LOG|TX)
Continue(TABLE|TX)
Continue(NULL)
```

RSOnOff(x,y,font)

This function allows user to turn on or off RS232 module, it can be used to transfer data to PC.

If you want to connect HHR to PC with USB cable first you have to turn off RS232 module with RSOnOff function.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
RSOnOff(67,10,1)
```

SetRSSpeed(x,y,font)

This function allows user to modify Baud Rate(Speed) off RS232 module. The frame will be displayed on screen so as user will be able to modify the speed. This value will be saved in non-volatile memory until it is changed again with SetRSSpeed once again.

Allowed values for Baud Rate are: 9600, 19200, 38400, 57600, 115200.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetRSSpeed(67,10,1)
```

SetRSDataBits(x,y,font)

This function allows user to modify Data Bits off RS232 module. The frame will be displayed on screen so as user will be able to modify the Data Bits. This value will be saved in non-volatile memory until it is changed again with SetRSDataBits once again. Allowed values for Data Bits are: 5, 6, 7, 8.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetRSDataBits(67,10,1)
```

SetRSParity(x,y,font)

This function allows user to modify Parity off RS232 module. The frame will be displayed on screen so as user will be able to modify the Parity. This value will be saved in non-volatile memory until it is changed again with SetRSParity once again. Allowed values for Parity are: N(for none), O(for Odd), E(for Even).

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetRSParity(67,10,1)
```

SetRSStopBits(x,y,font)

This function allows user to modify Stop Bits off RS232 module. The frame will be displayed on screen so as user will be able to modify the Stop Bits. This value will be saved in non-volatile memory until it is changed again with SetRSStopBits once again. Allowed values for Stop Bits are: 1, 2.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetRSStopBits(67,10,1)
```

TTWeight(x,y,font,column_name)

TTWeigh function generates frame on LCD, it user will enter this frame HHR will send a request to scale to receive stable weight. When weight is received it is displayed it in the field on the screen, user can either accept it and save in memory or delete to make HHR receive weight once again.

Please remember to set mask for column for at least 4 digits, because received weight and decimal point.

The Parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column name – name of column from TABLE or LOG where weight has to be saved, the size of letters is important.

Example:

```
TTWeight(67,10,1,WEIGHT)
```

TTAVG(x,y,font,column_name)

TTAVG function generates frame on LCD, it user will enter this frame HHR will wait for AVG message from scale. When message is received it is displayed it in the field on the screen, and saved in memory in pointed column.

Please remember to set mask for column for at least 4 digits, because of received weight and decimal point.

The Parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column name – name of column from TABLE or LOG where weight has to be saved, the size of letters is important.

Example:

```
TTAVG(67,10,1,WEIGHT)
```

IconixWeight(x,y,font,column_name)

IconixWeight function generates frame on LCD, it user will enter this frame HHR will wait for weight from scale. Weight will be send when user press 'Weight' button on scale.

When weight is received it is displayed it in the field on the screen, and saved in memory in pointed column.

Please remember to set mask for column for at least 6 digits, because of received weight and decimal point.

The Parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5

where 0 is the smallest size and 5 the biggest.

- column name – name of column from TABLE or LOG where weight has to be saved, the size of letters is important.

Example:

```
IconixWeight(67,10,1,WEIGHT)
```

GllWeight(x,y,font,column_name)

GllWeigh function works with Gallagher scales, function works the same as TTWeight.

Please remember to set mask for column for at least 4 digits, because received weight and decimal point.

The Parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column name – name of column from TABLE or LOG where weight has to be saved, the size of letters is important.

Example:

```
GllWeight(67,10,1,WEIGHT)
```

GllRuddWeight(x,y,font,column_name)

GllRuddWeigh function works with Gallagher scales, function works the same as TTWeight.

Please remember to set mask for column for at least 4 digits, because received weight and decimal point.

The Parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column name – name of column from TABLE or LOG where weight has to be saved, the size of letters is important.

Example:

```
GllRuddWeight(67,10,1,WEIGHT)
```

SetGPRSOnOff(x,y,font)

This function allows user to turn on or off GPRS module, GPRS(GSM) module is used to send SMS with IconSendSMS function. Please remember to put SIM Card PIN with SetSIMCardPin function before connecting with GSM network.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetGPRSOnOff(67,10,1)
```

SetSIMCardPIN(x,y,font)

This function allows user to set SIM Card Pin to HHR, if this field will be unfilled then HHR won't be able to connect to GSM Network,

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetSIMCardPIN(67,10,1)
```

IconSendSMS(x,y,global_number,sms_identifier,operational_el)

This function allows user to send SMS using HHR. SMS_identifier has to be placed in parameters of this section, SMS_identifier is label(name) of SMS defined in GSM section, for more information please refer to GSM section in Manual.

Please remember to put SIM Card Pin into the HHR with SetSIMCardPin(), and connect with GSM network with SetGPRSOnOff() function.

The parameters:

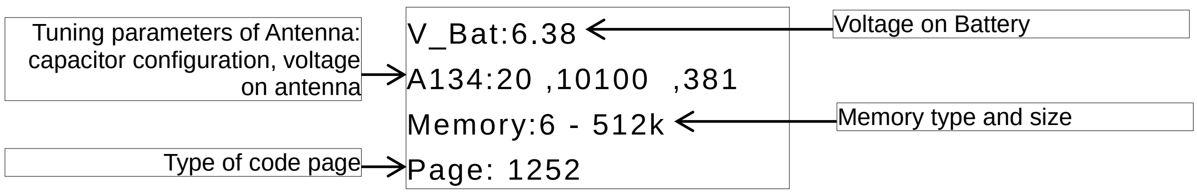
- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- global_number – its a number of global where phone number is stored.
- sms_identifier – it is the SMS identifier declared in GSM section, according to which user wants to send a SMS.
- operation_el – describe to which operation element user moves:
  1. operation_el=0 user stays at current element after the function finish it task.
  2. operation_el=1 user moves to previous operation element after the function finish it task.
  3. operation_el=2 user moves to next operation element after the function finish it task.

Example:

```
IconSendSMS(67,10,1,new_anim,0)
```

TestAntenna()

This function presents special screen, which presents tuning parameters of antenna, and system parameters of HHR.


Tuning parameters of Antenna: capacitor configuration, voltage on antenna
Type of code page
Voltage on Battery
Memory type and size

```
V_Bat:6.38
A134:20 ,10100  ,381
Memory:6 - 512k
Page: 1252
```

Function TestAntenna() should be used as the only one in the macro.

SetSleepTime(x,y,font)

This function allows user to change the time after which HHR turns down if no activities has been performed on HHR.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
SetSleepTime(67,10,1)
```

IconPrint(x,y,printout_name,field,global)

This function allows user to generate a printout using HHR, was introduce a special for printing more records. Using this function you will make HHR display printer icon. For more information please refer to PRINT section in Manual.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- printout_name – is label(name) of printout, defined in PRINT section
- field – this parameter describe to which data should be printed. This parameter has 4 option: table.all, log.all, current or field_name.
- global - this parameter describes which global value will be use as simple filter. HHR will print records where field_name=global only.

Example:

```
IconPrint(3,51,A,table.EID,1)
IconPrint(78,51,B,log.all,0)
IconPrint(108,27,CC,current,0)
```

ChangeBootPIN(x,y,font)

This function allows user to device lock - using a password, this feature prevents unauthorised use of HHR

The parameters:

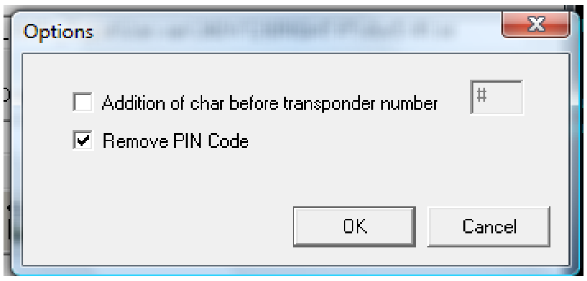
- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
ChangeBootPIN(0,8,2)
```

If you forget your password, you must follow the instructions below to clear the memory before you can access your HHR. All data will be lost!

To disabling PIN lock you have to load application by using HHR Manager with option Remove PIN code.


Options

☐ Addition of char before transponder number #

☑ Remove PIN Code

ChoiceFilter(x,y,font)

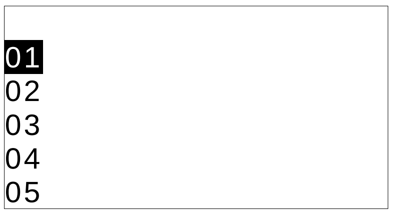
This function allows user to select filter defined in Global section.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.

Example:

```
ChoiceFilter(40,0,2)
```


```
01
02
03
04
05
```

Control Area functions

Control Area as it was already mentioned is executed just after Special functions is executed. Control Area consist only one function Copy().

Copy(db_ident)

This function enables you to copy a current record from indicated table to memory and after that work with it in Action Area, Copy function takes one parameter.

- db_ident – describe whenever user refer to TABLE or LOG. As this parameter you can use “table” or “log”. When you use “log” here without declaring it in HEADER section, LOG, sections is pointless. Log section is described in appendix C.

Action Area functions

FillExp(column_name,expression)

FillExp_p(column_name,expression)

FillExp or FillExp_p functions put an expression to pointed by user column at current record where data has to be pasted is empty. The expressions are results of a few functions, those functions are:

count(column_name|*)

Count expression counts all records according to parameter, this parameter might be a “*” , that means that HHR will count all records database. You can put any column name as parameter then HHR will count all record where value exist in indicated column.

The result of this expression will be placed at current record, at column_name pointed in functions

incCount(column_name)

incCount expression counts how many values is in table in pointed column, the function doesn't care about value, it only check if there is any value. The result of this expression will be placed at current record, at column_name pointed in functions the value which will be placed will be incremented with 1.

date()

date expression returns a current date according to data format from HEADER section. The result of this expression will be placed at current record, at column_name pointed in function.

time()

time expression returns a current time according to time format from HEADER section. The result of this expression will be placed at current record, at column_name pointed in function.

noofrec(db_ident)

noofrec expression returns a number of all records in table, noofrec -number of records. The result of this expression will be placed at current record, at column_name pointed in function. db_ident parameter describes to which Table you want to refer a main Table or to Log. If you want to refer to main Table use “table”, if you want to refer to Log table use “log”.

mFull()

mFull expression returns the amount of unfree memory in HHR's EEPROM. The result of this expression will be placed at current record, at column_name pointed in function.

“”

Using this parameter allows to put any text in to the field, text must be shorter then 23 characters.

mEmpty()

mEmpty expression returns the amount of free memory in HHR's EEPROM. The result of this expression will be placed at current record, at column_name pointed in function.

global()

global expression returns the value of global. The result of this expression will be placed at current record, at column_name pointed in function.

FillExp functions as you see, is rather control functions, by using it you can acquire information about Database size, amount of free or unfree memory. Move over you can fulfill date-type or time-type columns in your Database.

Example:

```
FillExp(NRDEAD,count(WEIGHT))
FillExp(NRUNKOWN,count(*))
FillExp(NRDATE,date())
FillExp(TIME,time())
FillExp(F_MEM,mEmpty())
FillExp(ATT,”LAMB”)
FillExp(PEN,global(1))
FillExp(PEN,global(FILTER))
```

dec(column_name,start,len)

dec() returns the number of transponder in decimal format.

tagtype()

tagtype() returns the type of transponder as the ‘HEX’ or ‘FDX’

countequ(column_name,global)

countequ() returns the number of fields with a value equal to the global.

amountRecJoined()

amountRecJoined() returns amount of joined record from LOG to current record in TABLE.

numberRecJoined()

numberRecJoined() return a number of joined record from LOG to current record in TABLE.

ShowMessage(column_name)

ShowMessage function shows message from current record and pointed by user column on HHR's LCD

Example:

```
ShowMessage(MESS)
```

Paste(column_name)

Paste functions simply pastes the values from memory to current record, before using Paste function a Copy function must be used in Control area, otherwise no data will be pasted.

Paste function take only one parameter a column name.

Paste function paste values from copied record from column_name given in function parameter to new record to column with column name given in parameters.

Example:

```
Paste(BREED)
Paste(BORN_DATE)
```

Paste_p(column_name)

Paste_p functions simply pastes the values from memory to current record only if field in memory where data has to be pasted is empty, before using Paste_p function a Copy function must be used in Control area, otherwise no data will be pasted.

Paste_p function take only one parameter a column name.

Paste_p function paste values from copied record from column_name given in function parameter to new record to column with column name given in parameters.

Example:

```
Paste_p(SOLD)
Paste_p(SEX)
```

PastePart_p(column_name,start,lenght)

PastePart_p functions simply pastes the part of values from memory to current record only if field in memory where data has to be pasted is empty, before using PastePart_p function a Copy function must be used in Control area, otherwise no data will be pasted.

PastePart_p function take only one parameter a column name.

PastePart_p function paste values from copied record from column_name given in function parameter to new record to column with column name given in parameters.

Example:

```
PastePart_p(SOLD,12,5)
```

PutGlobal(column_name, global_number)

PutGlobal_p(column_name, global_number)

These functions allows you to automatically fill some columns in current record with required values, HHR's user don't have any possibility to modify the values.

The parameters:

- column_name – this parameter describe to which column global value will be placed.
- global_number - this parameter describes which global value will be placed in table at pointed column, this parameter take a global number not whole value.

Example:

```
PutGlobal(FARM,0)
```

Sorting(column_name)

This function enables you to define a column after which HHR will be sorting data when user press Up Arrow or Down Arrow Buttons.

When user press an Up arrow HHR will change a current record to record where is smaller value, if user press Down Arrow he will move to next bigger value, the terms bigger and smaller has to be understood according to ASCII table. First character in each record has the highest priority, while the last char has the lowest priority.

Example

```
Sorting(FARM)
```

Lest assume that column FARM consist data listed below:

..
UK01
UK02
DEN1
LIT0
..

Lest assume that current record is the one with gray background, if you press Up Arrow HHR will move to 3th record because in ASCII table 'D' has smaller number then 'L'; If you would press Down Arrow HHR will move to 1st record, HHR will move to 1st and not to 2nd because while first 3 characters are equal in 1st and 2nd records, the last character('1') in 1st record is smaller then the last one('2') in 2nd record.

Sorting function is also used for defining a column for Speed mode, pressing Up/Down Arrow will make HHR skip to record with bigger/smaller value in pointed column.

Join(table_col_name,log_col_name)

This function enables you to connect table and log, if you wish to use Log in your application. 1st parameter defines column from Table and 2nd column from Log, those column will be connected, the idea of connecting will be explained in appendix C.

Example

```
Join(table.FARM,log.FARM)
```

FilterJoinAddLog(log_column_name, global)

FilterJoinAddLog_p(log_column_name, global)

These functions allow you to add new record to the log joined to the table (like AddRec(log) function. The log_column specifies the column connecting the table and log. It will be written by the global value.

Example:

```
FilterJoinAddLog(GROUP,FILTER)
FilterJoinAddLog(GROUP,1)
```

FillAddRec(db_ident)

This function allows you to add new record (like AddRec function) moreover this isn't a special function so you can use simultaneously AddRec and FillAddRec. This will be necessary for log users.

The parameter defines if new record will be added to log or table, if you want to create a new record in table put “table” as db_ident, if you want to create a new record in log put “log” as db_ident.

The idea of using this function will be explained in appendix C.

Example:

```
FillAddRec(log)
```

Beep(column_name,counter_no)

This function allows to make HHR beep when amount of values in pointed column will be bigger or equal then value of counter. There are 4 available counters which value can be modified with EditCounter function.

The parameters:

- column_name – this parameter describe in which column, values will be count
- counter_no - this parameter describes with which counter amount of values will be compared

Example:

```
Beep(FARM,0)
```

Empty()

Empty function prints an empty record when user enters macro.

Example

```
Empty()
```

IncPaste (column_name)

IncPaste functions increments (+1) a value from memory, pointed column and place it to the current record pointed column. Before using IncPaste function a Copy function must be used in Control area, otherwise no data will be pasted. If a value will contain a character other then number '0' will be placed instead incremented value

IncPaste function take only one parameter a column name.

Example

```
IncPaste(BREED)
IncPaste(BORN_DATE)
```

IncPaste_p(column_name)

IncPaste_p functions increments(+1) a value from memory pointed column and place it to the current record pointed column. Data will be placed only if field in memory where data has to be pasted is empty.

Before using IncPaste_p function a Copy function must be used in Control area, otherwise no data will be pasted. If a value will contain a character other then number '0' will be placed instead incremented value

IncPaste_p function take only one parameter a column name.

Example

```
IncPaste_p(SOLD)
IncPaste_p(SEX)
```

AddRecConf(db_ident)

This function creates a new record in Table or Log which need confirmation, only after confirmation a new record will remain in memory otherwise it will be removed.

The parameters:

- db_ident – this parameter define whenever new record will be created in Table or Log

Example:

```
AddRecConf(table)
```

Functions AddRecConf and IconConfirm are working in a team, AddRecConf creates a record which need confirmation to be placed in memory, confirmation can be made by IconConfirm. If user won't confirm in a record won't be saved in memory. IconConfirm function last parameter is an operation_el which define what will happen when function finishes its task (user press the icon).

In case that operation_el=1, function will confirm placing new record in memory, and move to Action Area, where AddRecConf is placed, so again new record will be created in memory which needs confirmation with IconConfirm function.

Print(printout_name,field,global))

This function allows user to generate a printout using HHR, was introduce a special for printing current record. For more information please refer to PRINT section in Manual.

The parameters:

- printout_name – is label(name) of printout, defined in PRINT section
- field – this parameter describe to which data should be printed. This parameter has 4 option: all.table, all.log, current or field_name.
- global - this parameter describes which global value will be use as simple filter. HHR will print records where field_name=global only.

Example:

```
Print(B,current,0)
```

SetFilter(column_name)

This function enable Filter futures in a macro and point the column to check if the data meets certain criteria.

Example:

```
SetFilter(PEN)
```

Index(column_name1, column_name2, column_name3,column_name4)

Indes() creates supporting table to quick searching data. You can create the index for max four columns in current macro. Index function works only on a system with coprocessor.

Example:

```
Index(EID,VID,COUTER,PEN)
Index(EID,VID)
```

Equal(column_name, column_name)

Equal() compares the values of fields in current record.

Example:

```
Equal(PEN,OLDPEN)
```

Special Action Functions

Special Screen functions make HHR move to Control Area and After that to Action Area and come back to the screen.

Special Action Functions are:

ReadNew(column_name)

LoopReadNew(column_name)

This function defines Transponder Read button to work in whole macro according to points below.

- User press Transponder Read button.
- HHR moves to Control Area and perform function attached to it.
- HHR moves to Action Area and reads Transponder
- HHR searches for read Transponder number in indicated column at parameters.
- If HHR finds a number in database a record number is set to the record in which was found.
- If HHR doesn't find a column with read Transponder number HHR creates a new record and set record number to the new record.
- HHR put a transponder number to column indicated in parameters.
- HHR perform all functions below ReadNew function in Action Area.
- HHR moves to the screen where user was before pressing a Transponder Read button.

The parameter which function consist describe where HHR has to look for a transponder number, and where data has to be written.

It is essential you to know that after pressing a Transponder Read button HHR performs all the action described above, next HHR come back to the screen where user was, the actions invisible so user won't see any changes on the screen connected with moving to Control Area and Action Area, the only difference that he will see is that data in operation elements will be updated(is is connected with change of current record number).

Example:

```
ReadNew(MOTHEREID)
```

ReadNew_p(column_name)

This function works similar to ReadNew(column_name) but data will be placed only for a new record.

LoopReadNew_p(column_name)

This function combine ReadNew_p() and Continue().

LoopReadNew_p execute Action area on continue reading of transponder after user enters macro where function has been used. It is possible to store data in memory only for a new records.

HHR would inform user about successful read of tag with LEDS and beeper.

User can break continuous reading with long press of power key or read transponder button.

Example:

```
LoopReadNew_p(EID)
```

ReadSeek(column_name)

LoopReadSeek (column_name)

This function defines Transponder Read button to work in whole macro according to points below.

- User press Transponder Read button.
- HHR moves to Control Area and perform function attached to it.
- HHR moves to Action Area and reads Transponder
- HHR searches for read Transponder number in indicated column at parameters.
- If HHR finds a number in database a record number is set to the record in which was found.
  - HHR put a transponder number to column indicated in parameters.
  - Perform action situated below ReadSeek function.
- If HHR doesn't find a column with read Transponder number
  - HHR put a waring message on screen “Transponder not found”.
  - HHR do not perform functions included to Action Area.
- HHR moves to the screen where user was before pressing a Transponder Read button.

The parameter which function consist describe where HHR has to look for a transponder number, and where data has to be written.

It is essential you to know that after pressing a Transponder Read button HHR performs all the action described above, next HHR come back to the screen where user was, the actions invisible so user won't see any changes on the screen connected with moving to Control Area and Action Area, the only difference that he will see is that data in operation elements will be updated(is is connected with change of current record number).

Example:

```
ReadSeek(MOTHEREID)
```

LoopReadSeek (column_name)

This function combine ReadSeek() and Continue().

LoopReadSeek execute Action area on continue reading of transponder after user enters macro where function has been used. It is possible to store data in memory only for a new records.

HHR would inform user about successful read of tag with LEDS and beeper.

User can break continuous reading with long press of power key or read transponder button.

Example:

```
LoopReadSeek(EID)
```

AddRec(db_ident)

This functions adds a new record at the end of indicated database and sets record number to just created record.

The Parameters:

- db_ident – describe whenever user refer to TABLE or LOG. As this parameter you can use “table” or “log”, as identifier to which database you refer to. Using “log” here without declaring it in LOG section and in HEADER section, is pointless. Log section is described in Appendix C.

Read()

This function read a transponder and place it in buffer, you can read and display content of this buffer with Transponder functions described at page 15. Read function perform all functions which ale below it in Action Area.

The data from the buffer can be send according to defined in HEADER section frame definition. HHR can use one of 3 interfaces USB, RS232 or Bluetooth. HHR will send data with the active one. If both RS and BT are disabled then data will be send with USB. Please keep in mind that to make HHR send data you have to put BT_true in HEADER section.

Bluetooth essentials are explained in Appendix D in HHR 3000 PRO Programmers Manual.

Special Screen Functions

Those function, as it was already mentioned, make HHR move from current screen (for better organization let's call a current screen a screen “A”) to Control Area and Action Area and perform functions included in those sections and come back to screen “A”.

Special Screen Functions are:

EditReadNew(x,y,font,column_name)

EditReadNew function defines Transponder Read button to work only when its function operation element is selected with frame, when user moves frame to other operation element Transponder Read button is undefined to read transponder.

When user put a frame on EditReadNew operation element HHR and press Transponder Read button HHR will behave according to points below.

- HHR moves to Control Area and perform function attached to it.
- HHR moves to Action Area and reads Transponder
- HHR searches for read Transponder number in indicated column at parameters.
- If HHR finds a number in database a record number is set to the record in which was found.
- If HHR doesn't find a column with read Transponder number HHR creates a new record and set record number to the new record.
- HHR put a transponder number to column indicated in parameters.
- Perform Action Area function situated below Special Action Function.
- Move to Screen A and put a frame on operation element created by EditReadNew.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).

- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column_name – this parameter describes form which column will be used.

Example:

```
EditReadNew(MOTHEREID)
```

EditReadSeek(x,y,font,column_name)

This function is similar to EditReadNew(x,y,font,column_name), it defines Transponder Read button to work if user put a frame on operation element created by this function. When user put a frame to its operation element and press Transponder Read button following action are performed:

  - HHR moves to Control Area and perform function attached to it.
  - HHR moves to Action Area and reads Transponder
  - HHR searches for read Transponder number in indicated column at parameters.
  - If HHR finds a number in database a record number is set to the record in which was found.
- If HHR finds a number in database a record number is set to the record in which was found.
    - HHR put a transponder number to column indicated in parameters.
    - Perform action situated below Special Screen function.
- If HHR doesn't find a column with read Transponder number
    - HHR put a waring message on screen “Transponder not found”.
    - HHR do not perform functions included to Action Area.
- Move to Screen A and put a frame on operation element created by EditReadSeek.

The parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column_name – this parameter describes form which column will be used.

Example:

```
EditReadSeek(MOTHEREID)
```

IconAddRec(x,y,db_ident,operation_el)

This function puts a icon on screen which enables user to add new record, after creating the new record HHR moves to Action Area and perform function situated below Special Action Function.

The parameters:

  - x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
  - y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
  - db_ident – describe whenever user refer to TABLE or LOG. As this parameter you can use “table” or “log”, as identifier to which database you refer to. When you use “log” here without declaring it in HEADER section, LOG, JOIN sections is pointless. Log section is described in

appendix C.

- operation_el – describe to which operation element user moves after coming back from Action Area:
    1. operation_el=0 user stays at current element after the function finish it task.
    2. operation_el=1 user moves to previous operation element(the one which is before IconAddRec) declared in App Program after the function finish it task.
    3. operation_el=2 user moves to next operation element(the one which is after IconAddRec) declared in App Program after the function finish it task.

Example:

```
IconAddRec(35,51,table,0)
```

If you are using **Speed Mode** you must be aware that none icons are available to put in Screens, the functions which aren't allowed to use are:

- IconDelete
- IconAddRec
- IconFindCol
- IconConfirm
- AddRecConf

EditPartNewField(x,y,font,column_name,start, lenght))

This function enables you to edit any part of value in whole database.

- User put a frame on operation element and put a value and finish editing a value at this frame.
- HHR moves to Control Area and perform actions included there.
- HHR search for the value in indicated column
- If HHR finds a value in any record in database it change record number
    - HHR moves to Action Area and perform function situated below Special Action Function.
- If HHR does not find a value in database
    - HHR creates a new record in database and change record number to just created record.
    - HHR moves to Action Area and perform function situated below Special Action Function.
- HHR move to Screen A and put a frame on operation element created by EditNewField function.

Parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column_name – this parameter describes form which column will be used, using rules were described to each function.
- start - The edited string will start at the *start*'th position in *string*, counting from zero. For instance, in the string '*ABCDEFG*', the character at position *0* is '*A”*, the character at position *2* is '*C*', and so

forth.

- length - the string returned will contain at most `length` characters beginning from `start`

Example:

```
EditPartNewField(35,35,1,EID,3,5)
```

EditNewField(x,y,font,column_name)

This function enables you to edit any value in whole database.

- User put a frame on operation element and put a value and finish editing a value at this frame.
- HHR moves to Control Area and perform actions included there.
- HHR search for the value in indicated column
- If HHR finds a value in any record in database it change record number
  - HHR moves to Action Area and perform function situated below Special Action Function.
- If HHR does not find a value in database
  - HHR creates a new record in database and change record number to just created record.
  - HHR moves to Action Area and perform function situated below Special Action Function.
- HHR move to Screen A and put a frame on operation element created by EditNewField function.

Parameters:

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- font – describe font size, there are 6 font sizes, you can select a one by choosing a digit from 0 to 5 where 0 is the smallest size and 5 the biggest.
- column_name – this parameter describes form which column will be used, using rules were described to each function.

Example:

```
EditNewField(35,35,1,MOTHERVID)
```

As you already know, Special Action Functions like ReadNew, ReadSeek and Read; and Special Screen Functions like EditReadSeek, EditReadNew define transponder read button to work. Of course there is a clear difference between those functions but if you would declare both of them in one macro collision is possible, because user declare transponder read button by ReadNew function to work in whole macro, and from the other hand user use a EditReadNew function, so after pressing a transponder read button it is impossible to know which function user want to perform ReadNew or EditReadNew.

You can't use EditReadNew or EditReadSeek if you use ReadNew, ReadSeek or Read in the same macro.

IconConfirm(x,y,operation_el)

IconConfirm function place an icon on HHR LCD, when user press this icon a newly created record will be confirmed and placed in memory.

- x – describe horizontal position of text. X can't take more then 127 (LCD dimensions).
- y – describe vertical position of text. Y can't take more then 63 (LCD dimensions).
- oper_el – defines what will happen when user press the icon:

- 0 – confirm creating record and exit macro.
- 1 – confirm creating record and move to Action Area.

Example:

```
IconConfirm(0,0,0)
```

! ***Read function rule.***

If any of Read function (ReadNew, ReadSeek) has been used in Action Area, any of Read function(EditReadNew, EditReadSeek) can't be used in Screens in a Macro.

It is extremely important you to follow this rule, otherwise Application Program won't be compiled by HHR Manager.

Macro Example

```
TABLE
  SPECIE;O;0;1;1;Ovino,Caprino,;;
  BREED;S;0;1;1;00;00;
  SEX;O;0;1;1;H,M,;M;
  BORN_DATE;D;0;1;1;0000000000;;
  FARM;G;0;1;1;##;0;
  ID_DATE;D;0;1;1;0000000000;;
  C_ID_ELEC;R;1;1;1;0000 0000 0000;;
  ID_TYPE;S;0;1;1;0;2;
  VET;G;0;1;1;##;1;
  RETAG;S;0;1;1;0;0;
END

MACRO
 begin_macro:NEW_REC
      begin_action_area
            ReadNew(C_ID_ELEC)
            PutGlobal(FARM,0)
            PutGlobal(VET,1)
            Paste(BREED)
            Paste(BORN_DATE)
            Paste(SPECIE)
            Paste(ID_TYPE)
            FillExp(ID_DATE,date())
      end_action_area

      begin_screen
            PrintText(0,0,0,"RECORDS:")
            PrintExp(46,0,0,count(*))
            PrintText(115,0,0,"1.1")

            PrintText(0,8,1,"EID:")
            PrintAnimalMark(24,6,2,C_ID_ELEC)
            PrintRetagging(34,6,2,C_ID_ELEC)
            PrintUserInfo(44,6,2,C_ID_ELEC)
            PrintReserved(62,6,2,C_ID_ELEC)
            PrintAddInfo(80,6,2,C_ID_ELEC)
            PrintCountryNumber(91,6,2,C_ID_ELEC)
            PrintAnimalNo(5,21,4,C_ID_ELEC)

            PrintText(0,38,1,"Sex:")
            EditChoiceField(25,37,1,SEX)

            PrintText(47,38,1,"Sp.:")
            EditChoiceField(69,37,1,SPECIE)

            IconDelete(3,51,0,table)
            IconFindCol(35,51,C_ID_ELEC,0)
      end_screen

      begin_screen
            PrintText(0,0,0,"RECORDS:")
            PrintExp(46,0,0,count(*))
            PrintText(115,0,0,"1.2")

            PrintText(0,7,1,"Born:")
            EditDateField(46,7,1,BORN_DATE)
            PrintText(0,22,1,"Id. date:")
            EditDateField(46,21,1,ID_DATE)

            PrintText(0,37,1,"Id. type:")
            EditMaskedField(46,36,1,ID_TYPE)

            PrintText(63,37,1,"Breed:")
            EditMaskedField(100,36,1,BREED)

            PrintText(0,52,1,"ReTag:")
            EditMaskedField(42,51,1,RETAG)
      end_screen

      begin_screen
            PrintText(0,0,0,"RECORDS:")
            PrintExp(46,0,0,count(*))
            PrintText(115,0,0,"1.3")

            PrintText(0,7,1,"Farm:")
            PrintField(4,18,2,FARM)

            PrintText(0,35,1,"Vet:")
            PrintField(4,46,2,VET)
      end_screen

      begin_control_area
            Copy(table)
      end_control_area
 end_macro
END
```