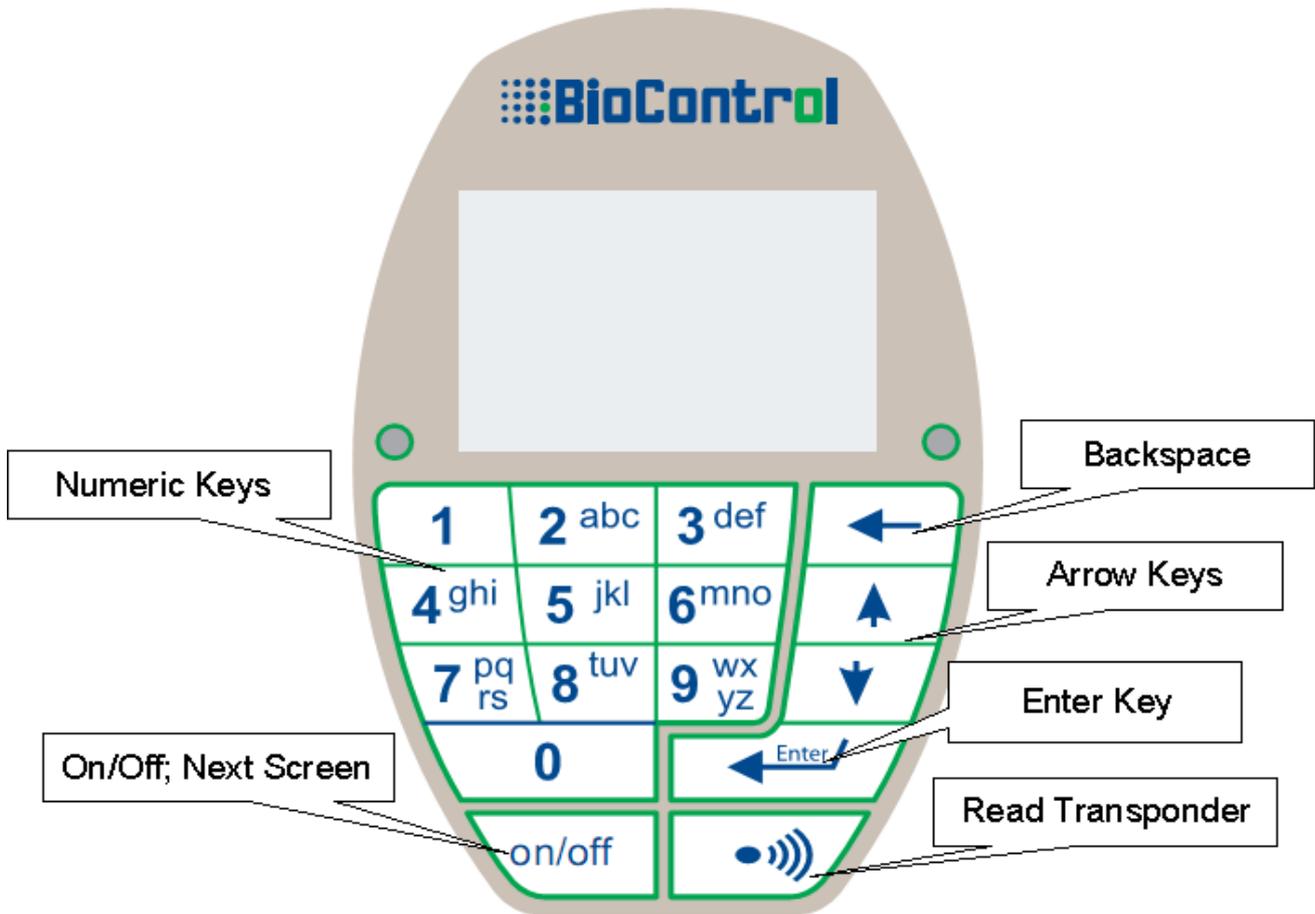

HANDHELD READER HHR 3000 PRO V2

HHR 3000 PRO Getting Started

CONTENTS

INTRODUCTION.....	4
GETTING STARTED.....	5
Menu.....	5
Table.....	6
FREE TO GO.....	8
Example: Macro NEW_REC.....	9
Appendix A: Example application.....	11
Appendix B: Keyboard operation.....	13



Keyboard Rules

The Key	Speed Mode	Normal Mode
Enter Key	Moves cursor to the next operation input box(skip icons except Icon Confirm); finish editing a value in input box; enter to icon function.	
Back Space	Short press: out of macro, menu, delete last character when editing a value in input box. Long press: out of editing a value in input box without changing it.	
0..9	If the cursor is in field(input box) starts editing a value in field(input box)	
Transponder Read	Depending on App Program reads transponder number.	
Power On/Off	Moves to the next screen, if HHR reaches last screen move to the 1 st one.	
Arrow keys	Save current record in memory and moves to the previous record (down arrow), or next record (up arrow).	Short press: move to the next field (input box) and icon (+ ; - ; search ; confirm) Long press: Move to the previous (Down arrow) or next (Up arrow) record. When user is editing a value in field(input box) moves between characters.

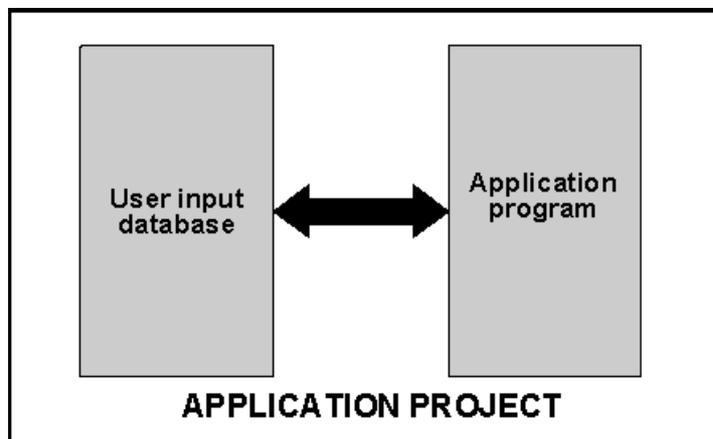
INTRODUCTION

HHR 3000 PRO enables you to customize your HHR especially for your needs, you can define your table, on which you will be working, your own menu and many more.

You can program your HHR with **HHR Manager** which is a PC program enabling to send or receive **User Input Database**; HHR Manager enables you to program your HHR with **Application Program**.

Application Program is file defining work of HHR, in we are going to create the one in this document.

Application Project consist Application Program written in simple HDS (Hhr Development Script) language and User Input Database. Those 2 files combined (compiled) by HHR Manager are Application Project.



The **Application Development System** contains:

- HHR 3000 PRO reader,
- Application Project,
- HHR Manager.

GETTING STARTED

This document will be based on example application “**Example APP 02BT**”

To begin creating your Application Program you must know 2 things:

- Menu that you want to be in your HHR
- Table definition.

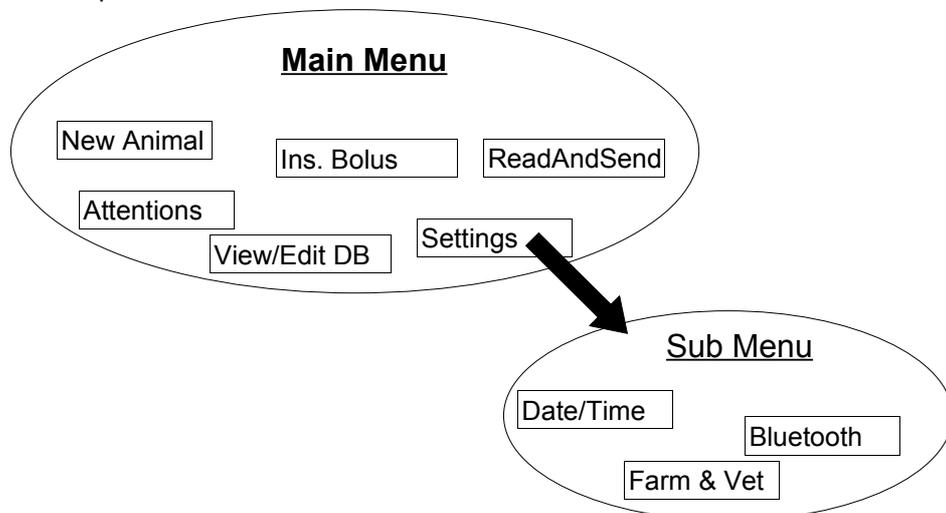
Menu

The design of menu is very simple, you need to specify the main menu(the menu just after HHR turns on), and all the sub menus, its easy to present in flow-chart or tree schema.

New Animal
Attentions
Ins. Bolus
View/Edit DB
ReadAndSend
Settings
|---- Bluetooth
|---- Date/Time
|---- Farm & Vet

On tree schema presented above we see the structure of menu, elements in bold are main menu, there is also one sub menu which will appear when user enter the “Settings” menu item, then “Bluetooth”, “Date/Time” and “Farm & Vet” elements will appear.

We can present it as a flow-chart.



Such project of menu can be defined in Application Program as:

```
MENU
    M0:{New Animal,M1};{Attentions,M2};{Ins.
Bolus,M3};{View/Edit DB,M4};{ReadAndSend,M5};
{Settings,M6}
    M1:NEW_REC
    M2:ATT
    M3:INSBOLUS
    M4:VIEW
    M5:SENDBYBT
    M6:{Bluetooth,M7};{Date/Time,M8};{Farm &
Vet,M9}
    M7:BLUETOOTH
    M8:DATE
    M9:GLOBAL
END
```

The M0 level is main menu, the elements of menu are in curly brackets, are labels which will be displayed on HHR's LCD, second parameter is a identifier to next element of each menu. In this way sub menu may be created (M6 is a sub menu from "Setting" menu element in M0 level). Final step for each menu is macro(M1, M2, M3, M4, M5, M7, M8, M9). If you need more detailed information refer to Menu Section Chapter in HHR 3000 PRO Programmers Manual.

Macro is a script(function, program) which defines sequence of action taken by HHR when user enters a menu item. Macro define screens which will appear when user is working in the macro. If you need more detailed information refer to Macro Section Chapter in HHR 3000 PRO Programmers Manual.

Table

The idea and design of Table is crucial in creating Application Program.

To help you during this process answer those questions:

1. How many columns do you need(how many information do you want to store in HHR for each transponder).
2. What kind of would be stored in each column – you can select a date, time, transponder number, freely definable value(string) or other advanced options like global type, message type, or option type. The global, message and option type allow you to create a list o available values that user can put in to this column, and all values which aren't on the list will be banned. If you need more data please refer to Table Section Chapter in HHR 3000 PRO Programmers Manual.

Basing on those points we may outline the structure of Table. We can easily present the structure of Table like below:

Column name	Column Type	Description
EID	Transponder number - R	The column will contain numbers of reader transponders, type R is appropriate in this case*.
Weight	Freely definable string - S	The column will contain a weight of an animal, user will put this value to HHR by keyboard, freely definable string is appropriate in this case.
Sex	Option type - O	The column will contain two values: "Male" or "Female" to avoid user errors we can use Option type**.
Cr_Date	Current date - D	The column will contain current date, the type D is appropriate in this case.

* - the alternative type for this column is Any Type, for more detailed refer to Table Section Chapter in HHR 3000 PRO Programmers Manual.

** - the alternative type for this column is Global Type, for more detailed refer to Table Section Chapter in HHR 3000 PRO Programmers Manual.

Table definition require also a few other parameters defining usage of column those parameters are:

- index, editable, null;
- mask;
- default value.

Parameters index, editable and null define if values in column are or aren't index, editable or null, for more detailed refer to Table Section Chapter in HHR 3000 PRO Programmers Manual.

Table definition:

```

TABLE
  EID;R;1;1;1;0000 0000 0000;;
  WEIGHT;S;0;1;1;0000;;
  SEX;O;0;1;1;M,F;F;
  CR_DATE;D;0;1;1;0000000000;;
END

```



FREE TO GO

Now You have a general idea of your Application Program, we have designed the Table and Menu the basis in every Application Program. Now You are free to go, but it isn't the end its just a beginning. Every Application Program consist 5-9 sections, there are sections which are required to HHR could work:

- Header
- Table
- Macro
- Menu
- Start Screen

All 9 sections have been shortly described in “Appendix A: Quick Reference” in HHR 3000 PRO Programmers Manual.

Beyond the idea of Table and Menu You must also know what HHR would do when user enters a menu item, the actions are defined by macros, so you have to know what macro should contain.

I advise you to study “Creating a macro and user functionality” and especially “The main advantages of HHR script language” in Macro Section Chapter, HHR 3000 PRO Programmers Manual. By this very short (2 pages only) part of document you will be able to create the best possible Macros.

The Example Application Program on which this document is based on consist 8 macros. First macro in Example APP 02BT Application Program will be analyzed.

Full source of this Application Program is available on included CD, in appendix A the key parts of code are included.

Example: Macro NEW_REC

This example refer Macro NEW_REC in Example application program “Example APP 02BT”, a source code of macro is in Appendix A.

Macro, generally, is set of screens where user can:

- enter/modify records in Table,
- delete/ create records in Table,
- set the parameters of HHR like Blue Tooth, Current Date or Time,
- display selected fields in a record,
- seek for a value in Table
- count values in Table
- Scroll records in Table without getting out of macro

All those possibilities and many others are brought with function, apart from it pay attention to “The main advantages of HHR script language” mentioned above.

In every macro 3 types of areas can be distinguished, action area, screen area and control area.

Action area is responsible for authentication, creating new records etc, In New_Rec Macro action area creates new record and place a transponder number to EID column, global values are placed in FARM and VET column. It is available to copy and paste data, last three functions in action area are responsible for pasting data from last record to newly created record, the columns which will be fulfilled with pasting are: BREEDING, DOB, SEX.

Functions from action area will be performed when user press Transponder Read button(defined by ReadNew), or when Special Screen functions are performed(EditReadNew, EditReadSeek, IconAddRec, EditNewField).

First screen in the macro is responsible for displaying EID and VID column with PrintField function and EditMaskedField.

```
PrintText(0,8,2,"EID:")
PrintField(0,20,4,EID)
PrintText(0,38,2,"VID:")
EditMaskedField(25,37,2,VID)
```

Moreover user can add a record, remove a record, and look for record in Table.

```
IconAddRec(3,51,table,1)
IconDelete(28,51,table,0)
IconFindCol(53,51,VID,0)
```

Rest of screen is used as interface, there are information about amount records in Table and current record number.

Second screen in macro is responsible for displaying and adding/modifying data in columns: DOB, SEX, BREEDING, WEIGHT.

```
PrintText(0,8,2,"Born:")
EditDateField(31,7,2,DOB)
PrintText(0,21,2,"Sex:")
EditChoiceField(25,20,2,SEX)
PrintText(0,34,2,"Breeding:")
EditChoiceField(56,33,2,BREEDING)
PrintText(0,47,2,"Weight:")
EditMaskedField(48,46,2,WEIGHT)
```

This screen include user-interface information like in previous screen: amount of records, number of current record.

Third screen in macro is responsible for displaying data from columns: FARM and VET.

```
PrintText(0,8,1,"Farm:")
PrintField(4,19,2,FARM)
PrintText(0,35,1,"Vet:")
PrintField(4,46,2,VET)
```

This screen include user interface information like in previous screens.

```
PrintText(0,0,0,"REC:")
PrintExp(21,0,0,noofrec())
PrintText(113,0,0,"1.1")
PrintText(53,0,0,"OF:")
PrintExp(69,0,0,count(*))
DrawLine(0,6,127,6)
```

All screens contain a lot of PrintText functions which enable to put any text on the screen, this is the best way to label a frames where user will be able to put a data from keyboard.

The last area in macro is **Copy area** which define functions taken just before action area is performed(by functions mentioned above). There is only one Control area functions – Copy(), which copies current record, it enables pasting data in action area. This functions has been used in control area in NEW_REC macro.

If you area doubtful about anything please refer to “Macro Section” Chapter in HHR 3000 PRO Programmers Manual.

Appendix A: Example application

```
// *****
// * bootloader: Program Loader.exe
// * interpreter: HHR_PRO_84.hex
// * compile: HHRApp_1_1.exe
// *
// *****
// 2007-03-18 11:12 PANAX
//
//
//          EXAMPLE APP 02BT
//
//
// *****

HEADER
EXAMPLE APP 02BT
2007/04/28
10:34:59
dd/mm/yyyy
hh:mm:ss
;
LOG_false
.
BT_true
GL_section
END

TABLE
EID;R;1;1;1;0000 0000 0000;;
VID;S;1;1;1;&& &&&& &&&&;
DOB;D;0;1;1;0000000000;;
SEX;O;0;1;1;M,F;F;
BREEDING;O;0;1;1;Y,N;Y;
WEIGHT;S;0;1;1;0000;;
VET;G;0;1;1;##;1;
FARM;G;0;1;1;##;0;
ATT1;O;0;1;1;DontMilk,Treat,Separate;;
ATT2;O;0;1;1;DontMilk,Treat,Separate;;
ATT3;O;0;1;1;DontMilk,Treat,Separate;;
FLAG1;S;0;1;1;#;;
FLAG2;S;0;1;1;#;;
FLAG3;S;0;1;1;#;;
FLAG4;S;0;1;1;#;;
END

GLOBAL
0:PL492520300030
1:002396303Q
END

MACRO

//*****// macro NEW_REC
begin_macro:NEW_REC
begin_action_area
ReadNew(EID)
```

```
PutGlobal(FARM,0)
PutGlobal(VET,1)
Paste(BREEDING)
Paste(DOB)
Paste(SEX)
end_action_area

begin_screen
PrintText(0,0,0,"REC:")
PrintExp(21,0,0,noofrec())
PrintText(113,0,0,"1.1")
PrintText(53,0,0,"OF:")
PrintExp(69,0,0,count(*))
DrawLine(0,6,127,6)
PrintText(0,8,2,"EID:")
PrintField(0,20,4,EID)
PrintText(0,38,2,"VID:")
EditMaskedField(25,37,2,VID)
IconAddRec(3,51,table,1)
IconDelete(28,51,table,0)
IconFindCol(53,51,VID,0)
end_screen

begin_screen
PrintText(0,0,0,"REC:")
PrintExp(21,0,0,noofrec())
PrintText(113,0,0,"1.2")
PrintText(53,0,0,"OF:")
PrintExp(69,0,0,count(*))
DrawLine(0,6,127,6)
PrintText(0,8,2,"Born:")
EditDateField(31,7,2,DOB)
PrintText(0,21,2,"Sex:")
EditChoiceField(25,20,2,SEX)
PrintText(0,34,2,"Breeding:")
EditChoiceField(56,33,2,BREEDING)
PrintText(0,47,2,"Weight:")
EditMaskedField(48,46,2,WEIGHT)
end_screen

begin_screen
PrintText(0,0,0,"REC:")
PrintExp(21,0,0,noofrec())
PrintText(113,0,0,"1.3")
PrintText(53,0,0,"OF:")
PrintExp(69,0,0,count(*))
DrawLine(0,6,127,6)
PrintText(0,8,1,"Farm:")
PrintField(4,19,2,FARM)
PrintText(0,35,1,"Vet:")
PrintField(4,46,2,VET)
DrawLine(0,63,127,63)
end_screen
begin_control_area
Copy(table)
end_control_area
```

```

end_macro
END

MENU
    M0: {New Animal,M1}; {Attentions,M2}; {Ins.
Bolus,M3}; {View/Edit DB,M4}; {ReadAndSend,M5};
{Settings,M6}
    M1:NEW_REC
    M2:ATT
    M3:INSBOLUS
    M4:VIEW
    M5:SENDBYBT
    M6: {Bluetooth,M7}; {Date/Time,M8}; {Farm &
Vet,M9}
    M7:BLUETOOTH
    M8:DATE
    M9:GLOBAL
END
START_SCREEN
    begin_screen
        PrintText(0,0,0,"EXAMPLE APP 02BT")
        DrawLine(0,6,127,6)
        PrintText(33,20,5,"I D & T")
        DrawLine(0,56,127,56)
        PrintText(0,58,0,"HHR:")
        PrintVersion(21,58,0)
    end_screen
END

WARNING
w_WAIT:Please/wait!
w_UNIQUE:This value/must be unique!
w_NOEDIT:This value/is NOT editable!

```

```

w_NOTNULL:This value/must be not null!
w_LOWBATTERY:Low battery!
w_CHARGEBATTERY:Battery/charging!
w_NOT_FREE_MEMORY:No free memory!
w_NOT_STORED:NOT STORED!
w_STORED:STORED!
w_DATE_GOOD:Date changed.
w_DATE_NO_CHANGE:Date not/changed.
w_DATE_NOT_GOOD:Wrong date!
w_TIME_GOOD:Time changed.
w_TIME_NO_CHANGE:Time not/changed.
w_TIME_NOT_GOOD:Wrong time!
w_NOTAG:No Tag.
w_READING:Reading transp.
w_SHUTOFF:See you!
w_USBCONNECTED:USB/connected!
w_NEW_REC:New record/created!
w_CHOICE:Choice:
w_FIND:Find:
w_TAGNOTFIND:Transponder/no. not found!
w_TAGEXIST:Transponder/already exist!
w_YES:YES
w_NOT:NOT
w_ADD_REC:Do you want/to add/new rec.?
w_DEL_REC:Do you want/to del/cur. rec.?
w_UNKNOWN_MES:Unknown/message!
w_VALEXIST:This value/already exist!
w_BTACTIVE:Bluetooth/is activated.
w_BTNOACTIVE:Bluetooth/is closed.
w_ON:Turn on.
w_OFF:Turn off.
w_MEMORY:MEMORY:
END

```

Appendix B: Keyboard operation

<u>Keyboard Rules</u>		
The Key	Speed Mode	Normal Mode
Enter Key	Moves cursor to the next operation input box; finish editing a value in input box; enter icon.	
Back Space	Short press: out of macro, menu, delete last character when editing a value in input box. Long press: out of editing a value in input box without changing it.	
0..9	If the cursor is in field(input box) starts editing a value in field(input box)	
Transponder Read	Depending on App Program reads transponder number.	
Power On/Off	Moves to the next screen, if HHR reaches last screen move to the 1 st one.	
Arrow keys	Save current record in memory and moves to the previous record (down arrow), or next record (up arrow).	Short press: move to the next field(input box); Long press: Move to the previous (Down arrow) or next (Up arrow) record. When user is editing a value in field(input box) moves between characters.

In both example application programs Normal Mode has been used.