

Handjet EBS-260 Bluetooth® Scanner Communication Set-Up

Instructions for Scanning a Bar Code and Printing its Contents *Scanner Mode 1*

SCANNER OPTION INSTALLED IN PRINTER

- The EBS-260 Printer has the Bluetooth function installed and is ready to operate
- Typically, the procedure for pairing the printer and scanner is done one time only
- **DISABLE ALL OTHER BLUETOOTH DEVICES, INCLUDING YOUR CELL PHONE**

RECOMMENDED SCANNER MODEL (FOR USE WITH EBS HANDJET PRINTERS)

- Datalogic
 - Gryphon™ | GBT4400

START PRINTER AND ENTER A MESSAGE

- Start the Handjet EBS-260
- SEE PAGE 3 FOR SCREEN SHOTS OF THE FOLLOWING STEPS
 - Create a Project
 - Enter a Project Name
 - Select the + sign to add a field
 - Select “Communications Port”
 - Check that the “Bluetooth” is selected as the default Port on the Communications Tab
 - Change other parameters (font, etc.) as necessary for your message

CREATE THE BARCODE TO ESTABLISH THE CONNECTION BETWEEN SCANNER AND PRINTER (PAIRING PROCESS)

- Select the message just created on the Handjet EBS-260
- Open the Bluetooth Menu from the printer home screen
 - Tap the Bluetooth symbol
- This will show you the Bluetooth address of the printer



- Here is a free on-line barcode generator website

- <http://www.terryburton.co.uk/barcodewriter/generator/>

Required parameters

Symbology: Code 128

Contents: ^FNC3LnkB0080980E241E

Options: includetext parsefnc

Additional parameters

Scale: 2 2

Rotate: 0

Make Barcode (it will appear below)

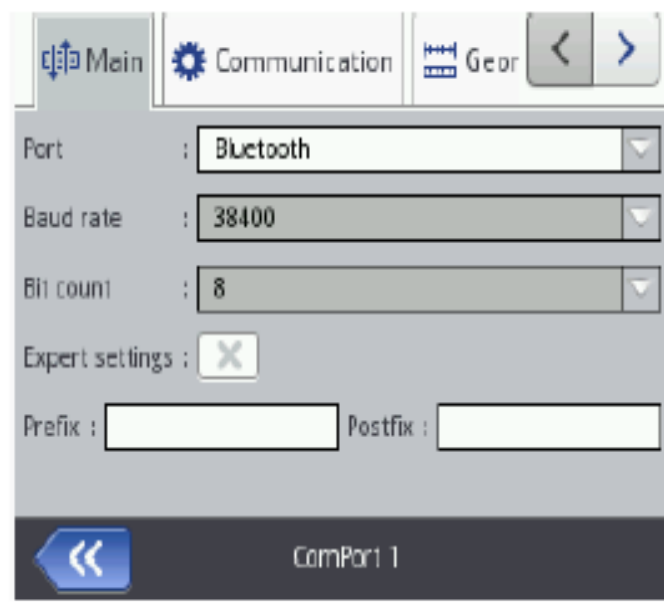
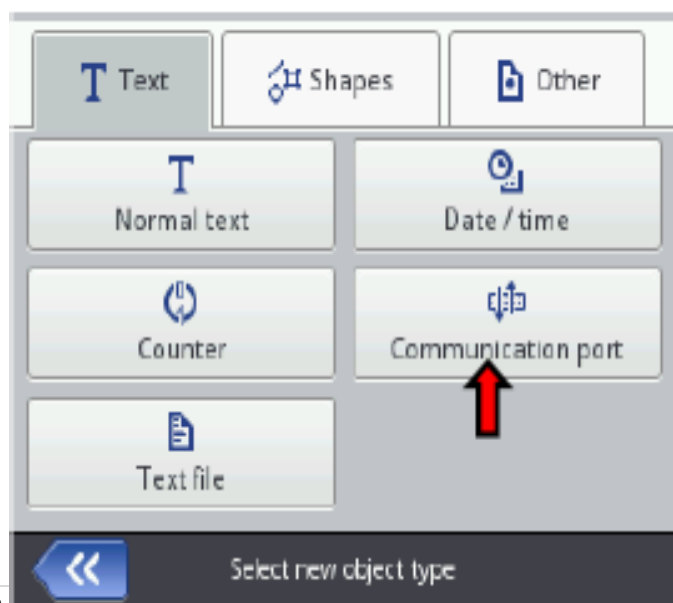
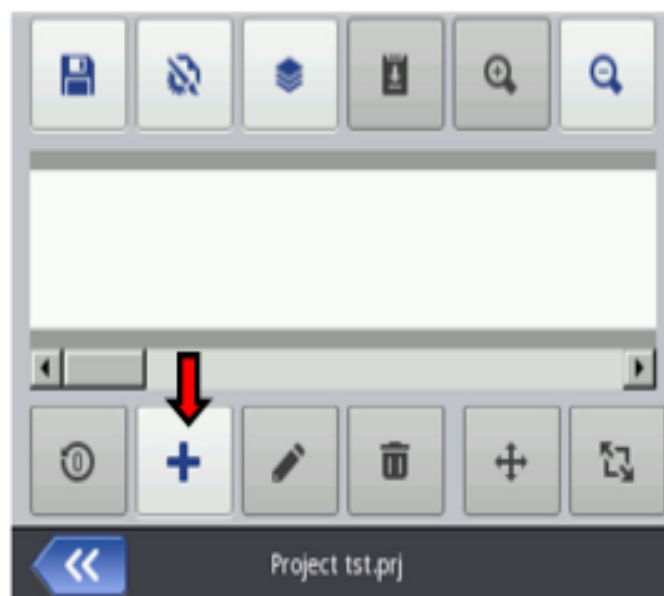
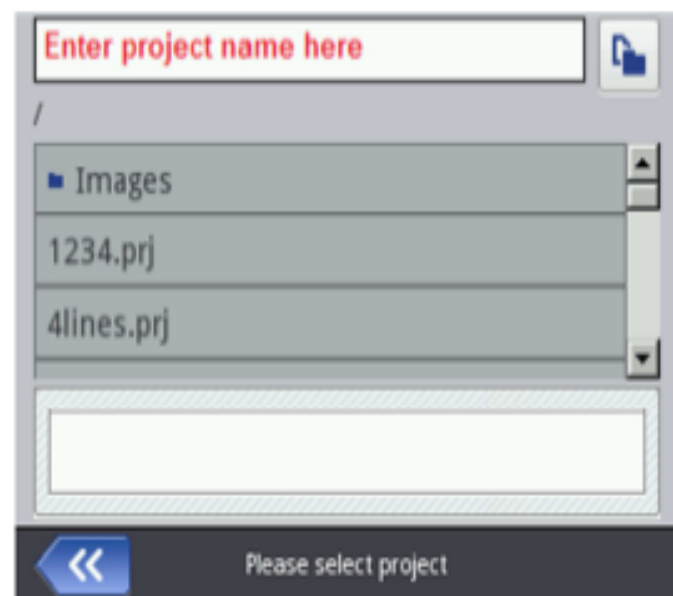
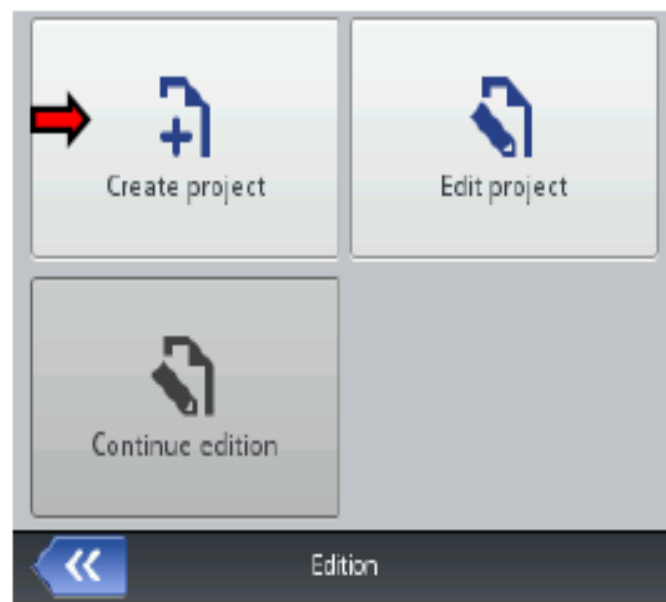
- In the “Symbology” parameters, choose Code 128
- In the “Options” parameters, enter the following:
 - Includetext parsefnc
- In the “Contents” parameters, enter the following EXACTLY as shown:
 - ^FNC3LnkBxxxxxxxxxxxx
 - Where xxxxxxxxxxxx is replaced with the Bluetooth® address of the printer found in the previous step
- Click “Make Barcode”
 - The barcode will be created and appear lower on the web page
- Print this barcode for future reference



- Scan the bar code you’ve just printed (can also scan directly on the monitor screen)
- The scanner will give an audible beep
- The Bluetooth Connection symbol should now light in green

SCAN THE BAR CODE AND PRINT ITS CONTENTS

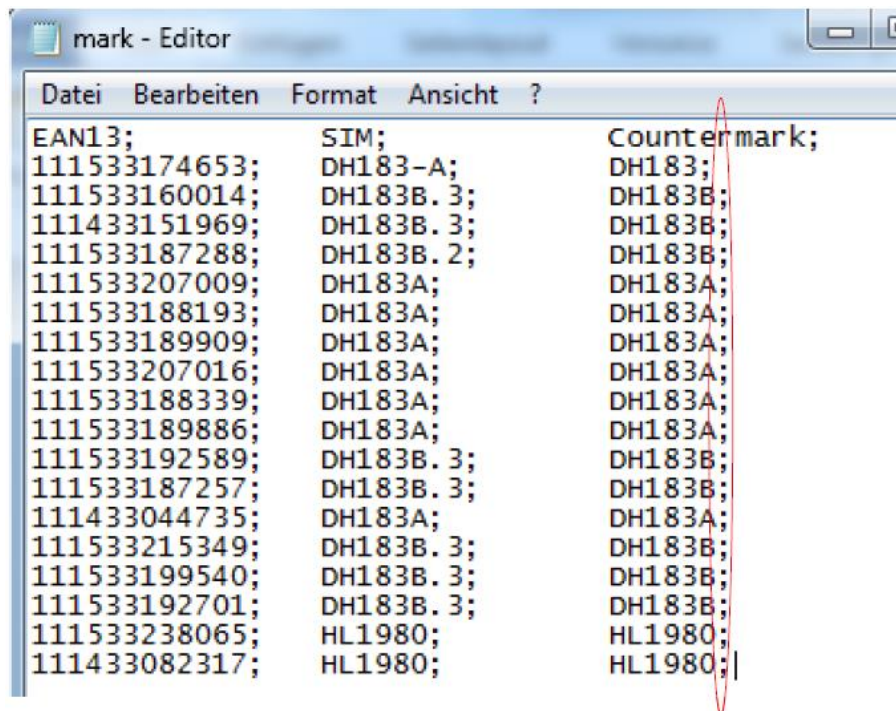
- Start the Print Mode
- Read the first bar code for printing its contents: Scan successive bar codes to print their contents;
- Pairing process is done only once; They will automatically connect when both units are started



Temporary Instructions for Scanning a Bar Code And Printing from a Database *Scanner Mode 2*

GENERATE A .CSV FILE

- Create or open the .csv data file with a *Standard Editor* like Microsoft Editor as follows:
- Put the lines and columns in the position shown in the example
- Insert a semicolon (;) at the end of every entry, *especially the last column*
- First column is the Bar Code Identifier (the Bar Code Number)
- Insert as many columns as necessary



SAVE THE .CSV FILE

- Enter the [name of your database].csv
- Choose UTF-8 Encryption
- Save the File
- Change the Program Properties to Open, Change and Save the file only with the editor

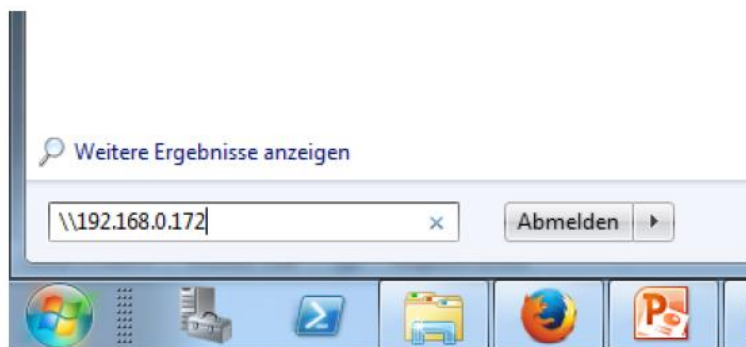


EMBED THE DATABASE (.CSV)

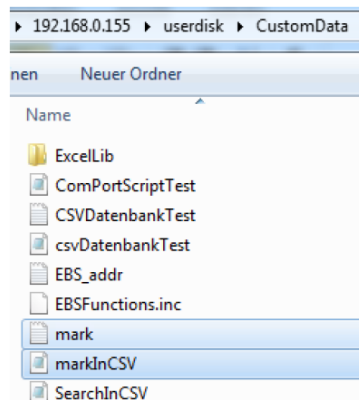
- Connect the EBS-260 Handjet Printer with your Wi-Fi Network



- Copy the Network Address and search the printer on your PC (e.g. [\\192.168.0.174](http://192.168.0.174))



- Select the folder "UserDisk"
- Select "Custom Data"
- Copy the .csv file to this folder



EDIT THE SCRIPT (.PHP)

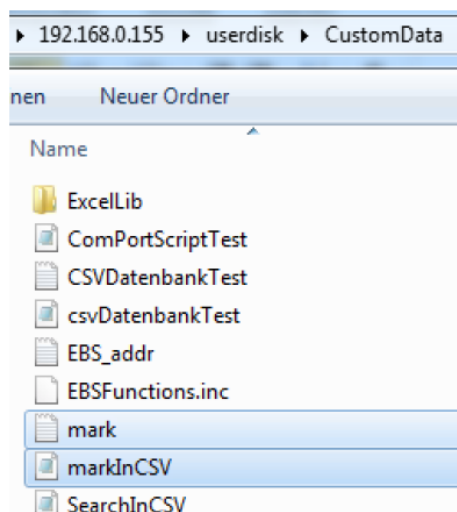
- Connect the EBS-260 Handjet Printer with your Wi-Fi Network
- Use the same procedure above, navigate to “UserDisk” and “Custom Data”
- Open the PHP Script File

```
<?php
ini_set('display_errors',1);
require_once('EBSFunctions.inc');
$result = $EBSFunc->CSV_Search('mark.csv',; '$argv[1]', '1', '$2#$3');
$EBSFunc->writeResult($result);
?>
```

- Insert the actual .csv file name in line 4
- Insert the number of the column(s) you wish to be printed in line 4
- The PHP Script uses the entries from the .csv database as follows:
 - Column 1 in the table is the search column
 - Columns which will be printed are \$2 to \$n
 - \$2#\$3#\$4#\$5 Etc.
 - After scanning a bar code, the bar code content will be searched in Column 1, and then printed from the associated row
-

EMBED THE SCRIPT (.PHP)

- Connect the EBS-260 Handjet Printer with your Wi-Fi Network
- Copy the Network Address and search the printer on your PC (e.g. [\\192.168.0.174](http://192.168.0.174))
- Select the folder “UserDisk”
- Select “Custom Data”
- Save or Copy the .php file to this folder



SET THE OBJECT PARAMETERS

- Object 1 (Text 1) will be the Comport 1 which is getting data from the Comport (in this example from the Bluetooth Scanner)
- Create as many other fields as necessary to print from the database
- Change the object parameter to the desired text fonts, size, etc.
- Open the *Expert Settings* (Drop-down)
 - Select the correct PHP script name
 - Set the formatting string to %ls
 - Enter the Separator Data Count (number of fields you wish to print from the database)
 - Object 2 (3, 4, etc.) relates to the text fields you created in the message
 - View or change each object name if desired
 - Select “Print last message” for No-Data Action to the print the last text (until a new bar code is scanned)
 - Connect the EBS-260 Handjet Printer with your Wi-Fi Network
- Save the object parameters
- Save the entire project

The screenshot shows the 'Object parameters' dialog box with the following settings:

- Port:** Bluetooth
- Baud rate:** 38400
- Bit count:** 8
- Sending messages format:** Normal
- Ready to print after:** End character
- Data buffering:** Enabled
- No-data action:** Print last message
- Max repeat count:** 0
- ASCII code:** ☒ (checked)
- Show experts settings:** (circled in red)
- On starting printing:** (empty text box)
- On processing data:** (empty text box)
- On finishing printing:** (empty text box)
- Script name:** markInCSV.php
- Formatting string:** %ls
- Separator data count:** 2
- Separator character:** #
- ASCII code:** ☐ (unchecked)
- Object 2:** Text 1

START PRINTING

- **Connect the bar code scanner with the EBS-260**
- **Select the project file on the printer (.prj file)**
- **Press the “Print” button on the Handjet to set the printer to print mode**
 - **There will be a green printing symbol on the LED screen**
- **Scan a bar code**
 - **The printer confirms that it received data with a short audible signal**
- **Pull the trigger and print**
- **After printing the last field, the printing symbol on the LED screen turns from yellow back to green indicating you may scan a new bar code or print the same data again**
- **The print job may be set up in different ways**
 - **Print each field in successive lines by using the text divider**
 - **Print 4 fields at once**
 - **Dependent on how the message is set up**

User Scripts (PHP) and Different Functions

Object (ComPort) allows the user to process data from the Bluetooth Module (and incoming from an external device). This allows the user to perform operations on data sent (e.g. from a Bar Code Scanner connected via Bluetooth to the printer).

The purpose of the script is to download the sent data, process the data, and then transfer the data to the Object (ComPort). Scripts can be created by the user while maintaining a few rules:

- Scripts should be created as .php files
- When using a script there is no access to system commands
- Scripts can only be used by inserting them into the “Custom Data” file

After saving the script file in the appropriate directory (Custom Data), it can be chosen for the project as a service script object (ComPort).

For each separator in the object (ComPort), the data that has been transferred to the separator will be passed to the script, so the script will be executed as many times as separators have been filled during data transfer.

Following are sample scripts, along with an explanation of their individual parts.

Basic Assumptions for PHP script processing data received from the Communications Port:

INPUT:

\$argv [1] – Data received from the communications port are forwarded to the PHP script as a command line argument, which is shown inside the script as a variable \$argv [1].

OUTPUT:

All output data to be passed to the printer should be passed as a **WriteResult** function argument as part of the EBSFunctions Library.

EBS Functions Library:

The printer directory CustomData contains the EBSFunctions Library.

To load the library use:

require_once ('EBSFunctions.inc')

This function will be shown as **\$EBSFunc** in the script.

The library includes the following functions:

- **WriteResult (\$data)**
 - \$data argument passed to this function will be passed back to the object of ComPort for the purpose of further processing (according to the objects settings) and for printing, for example:

\$EBSFunc->WriteResult ('EBS-260 HANDJET')

will cause the string 'EBS-260 HANDJET' to be printed

- **Excel_Search (\$XLSFile, \$SearchValue, \$SearchInCol, \$ReturnPattern)**
 - This function is used to search the data in MS Excel Files
 - **\$XLSFile** – this is the name of the MS Excel File where the search will take place; For Example:
 - **BarcodeTest.xls**
 - **\$SearchValue** – this parameter specifies how the phrase is to be searched; For Example:
 - **'EBS-260, \$argv[1], etc.**
 - **\$SearchInCol** – when using this parameter, you must specify which column or columns are to be searched for a particular phrase. The columns are numbered starting from number 1. If searching several columns, the numbers should be separated by a comma; For example:
 - **'1', '1,2', etc.**
 - **\$ReturnPattern** – this parameter defines which cells found in the file are to be returned. Without this pattern, the entire line will be returned. Numbers of columns to be returned must be preceded by a "\$" sign; For example: '\$2' will return the contents of the second column; '\$2 - \$3' will return the contents of the second and third columns with their separation by '-'.
- Data that is found and read from the MS Excel File will be output to the printer with the Excel_Search function.
- Sample script searches for data in an Excel File might look like this:

```
<? php  
  
ini_set('display_errors',1);  
  
require_once ('EBSFunctions.inc');  
  
$result = $EBSFunc->Excel_Search ('BarcodeTest.xls', $argv [1], '1', '$2');  
  
$EBSFunction->WriteResult ($result);  
  
?>
```

- **CSV_Search (\$CSVFile, \$DataSeparator, \$SearchValue, \$SearchInCol, \$ReturnPattern)**
 - This function is used to search the data in a CSV File
 - **\$DataSeparator** – this parameter is used to specify the data separator for the CSV File
 - Other parameters are the same as listed in the Excel_Search function
- Data that is found and read from the CSV File will be output to the printer with the CSV_Search function.
- Sample script searches for data in a CSV File might look like this:

```
<? php

ini_set('display_errors',1);

require_once ('EBSFunctions.inc');

$result = $EBSFunc->CSV_Search ('ExampleData.csv', $argv [1], '1', '$2');

$EBSFunction->WriteResult ($result);

?>
```

- **Retrieve data from a database**
 - **DB_init (\$db_type, \$db_host, \$db_user, \$db_passwd, \$db_name)**
 - This function is used to initiate a connection to a database
 - **\$db_type** – database type which will be used to make a connection
 - Supported types include: MySQL, MSSQL, PostgreSQL
 - **\$db_host** – the address of the database server
 - **\$db_user** – the name of the database user
 - **\$db_passwd** – the database password for the user
 - **\$db_name** – the name of the database to be connected
 - If there is an connection problem, a 'Connection Error" will occur and the script will end its action
 - **DB_Close ()** – this function is used to close the database connection
 - **DB_Search (\$table, \$where, \$resultPattern)** – function performs a SELECT query for the database to which you are connected.
 - **\$table** – table in the database from which the data is retrieved
 - **\$where** – WHERE clause to the SQL query; For Example: 'Printer' = " EBS-260 "
 - **\$resultPattern** – pattern for the return value, the column names from the database to be returned should be in "; e.g., " 'Name' " will return the contents of the Name column; " Hello World 'Name' " will return the string "Hello World" with the contents of the Name column appended to it.
 - Data that is found and read from the database will be output to the printer with the DB_Search function.

- Sample script searches for data in a database might look like this:

```
<? php  
  
ini_set('display_errors',1);  
  
require_once ('EBSFunctions.inc');  
  
$EBSFunc->DB_Init('MySQL','localhost','exampleUser','examplePasswd','exampleData';  
  
$result = $EBSFunc->DB_Search ('exampleTable', 'Printer' =" EBS-260 ", " Hello World  
'Printer'");  
  
$EBSFunction->DB_Close ();  
  
$EBSFunction->WriteResult ($result);  
  
?>
```