



Application Note

AN_306

FT800 Jackpot Application

Version 1.0

Issue Date: 2014-03-25

This document is to introduce the Jackpot Demo Application running on both MSVC and Arduino. The objective of the Demo Application is to enable users to become familiar with the usage of the FT800, the design flow, and display list used to design the desired user interface or visual effect.

Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold FTDI harmless from any and all damages, claims, suits or expense resulting from such use.

Future Technology Devices International Limited (FTDI)

Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom

Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758

Web Site: <http://ftdichip.com>

Copyright © 2014 Future Technology Devices International Limited

Table of Contents

1	Introduction	2
1.1	Overview	2
1.2	Scope.....	2
2	Application Flow.....	3
3	Description.....	4
3.1	Initialization.....	4
3.1.1	Global Arrays.....	4
3.1.2	Load Bitmap	4
3.2	Functionality	4
3.2.1	Fonts.....	10
3.2.2	Audio	10
3.2.3	Buttons.....	10
4	Contact Information.....	11
Appendix A – References		12
Document References.....		12
Acronyms and Abbreviations		12
Appendix B – List of Tables & Figures		13
List of Tables		13
No table of figures entries found.....		13
List of Figures		13
Appendix C – Revision History		14

1 Introduction

This application is a custom version of a Slot Machine game and it demonstrates the usage of built-in FT800 widgets and primitives. Steps to construct the User Interface (UI) and the algorithm used for some components are also discussed in this document.



Figure 1-1 Slot Machine

1.1 Overview

This document presents a basic understanding of the FT800 built-in feature sets.

For Arduino platforms a FAT formatted SD card is required and all files in the “Test” folder must be copied to the root directory of the SD card.

1.2 Scope

This document will be used by software programmers to develop GUI applications by using the FT800 with any MCU via SPI.

2 Application Flow

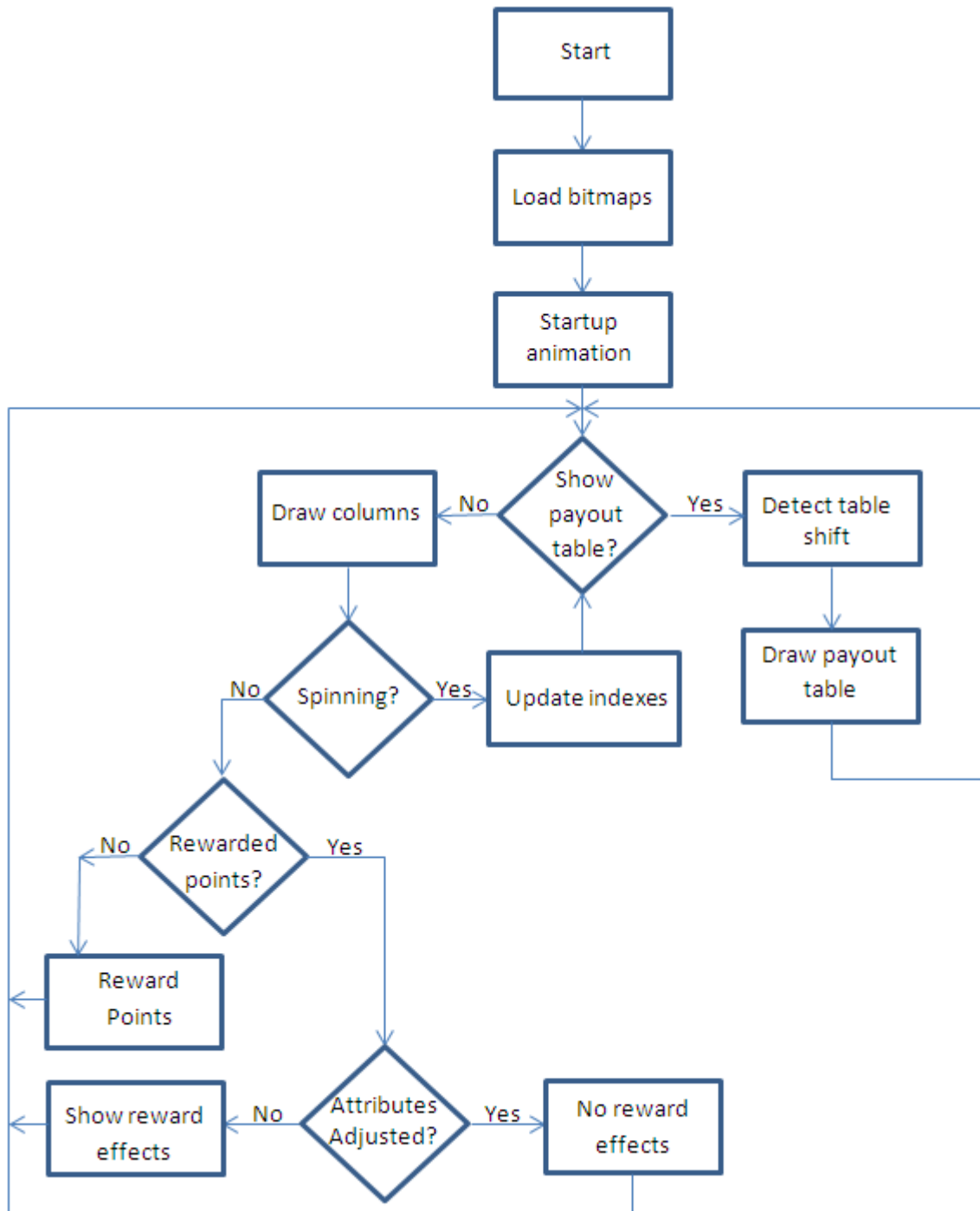


Figure 2-1 Application flowchart

3 Description

3.1 Initialization

3.1.1 Global Arrays

Numerous arrays were used for the UI construction.

- spinning_column_t - Holds the information about each of the spinning columns.
 - curIndex – This variable holds the index in the iconArray which is the first icon to be displayed for the column.
 - velocity – The purpose of this variable is to randomize the icons. After the spin button has been released, velocity decreases at a fixed rate in each update loop. The column stops when the velocity becomes 0.
 - iconArray – This array holds bitmap handles in a random order.
- bet_line_t- This predefined table holds the coordinates for each bet line and the line color.
- payout table - A simple array which the bitmap handle corresponds to the index of the array and the value of the index is the payout amount.
- bitmap_header_t - It holds the bitmap attributes such as: width, height, stride and location in the memory.

3.1.2 Load Bitmap

This application uses bitmaps extensively. All bitmaps used in this application are in RGB565 format except the spinning column overlay which is in L8 format. Bitmaps are loaded into the ram before the game loop.

This application also demonstrates the usage of various types of supported bitmap formats.

- .jpg - Many photo editing software packages support this format so quick modifications are easy, but this format doesn't support transparency. The user can convert Jpeg images to one of the supported bitmap formats that retain transparency such as ARGB4. An image conversion tool is available at: http://www.ftdichip.com/Support/Utilities/EVE/img_cvt_0.5.zip and more information regarding supported bitmap formats can be found in the [FT800 Programming Guide](#).
- .bin - This binary format is compressed by ZLIB algorithm.
- .raw - This binary format can be downloaded into FT800 graphics memory directly without decoding and decompressing. Large bitmaps should be loaded into the ram directly to prevent hanging during the loading stage.

Note 1: Bitmaps with the same dimensions and the format which they will be used can be edited into a long vertical bitmap strip. The single bitmap would only takes up one bitmap handle and the individual bitmap can be accessed by the CELL parameter.

Note 2: When all the user definable bitmap handles have been exhausted, the bitmap drawing method of specifying the source, layout, and size is required.

3.2 Functionality

This application starts up with an animation of spinning columns and ends up on the highest reward combination of symbols. Other than the startup animation the status bar is fixed at the lower portion of the screen. It is constructed by repeating a single bitmap line across the screen width and buttons and status text are then drawn on top of it.

This demo application has two playing modes. One mode is the version where the middle icons of the selected columns are the only ones being considered for rewards. Touch of the + or –

button under the *Column* field will increase or decrease the number of selected columns respectively. The other mode is betting on the predefined lines. Touching one of the numbered buttons on either side of the screen will automatically include all the other lower bet lines. Similarly, touch the + or - button under the *Line* field will also increase or decrease the bet lines respectively. The winning combination is rewarded according to the payout table that appears when the *payout* button on the lower left side of the screen is pressed.

One of the main screens is the *spinning-columns* screen. The spinning-columns screen displays the spinning columns and it is the default screen for this demo application and the steps to construct the screen are as follows:

```
{
draw coin background;
update the current_icon index for each column and decrease velocity;
for(i=0;i<number of spin columns;i++){
    if(column velocity == 0)
        this column's drawn index = current_icon index + 1;
    for(j=0;j<3;j++){
        draw the index of (current_icon index + j);
        draw L8 formatted overlay bitmap;
        draw outer overlay;
        highlight selected spinning column with two dots;
        Y coordinate += ICON_HEIGHT;
    }
    X coordinate += (ICON_WIDTH + gap);
    Y coordinate = starting Y coordinate;
}
draw line bet buttons on both sides of the spinning columns;
}
```

The background is a single 16*16 bitmap and it fills up the whole screen by using REPEAT as the value for both wrapx and wrapy parameters in the BITMAP_SIZE function. REPEAT should only be used if the corresponding axis dimension is power of two, otherwise the result is undefined.

While the spin button is pressed the *current_icon* index is randomly generated during each update loop. After the *spin* button is released, a random velocity is then generated for each column and the velocity decreases at a fixed rate till it reaches 0, which is when the column stops.

The purpose of the L8 formatted overlay bitmap is to provide a smooth color transition from the outer edges of the icons to the inner edges of the outer overlay bitmap by blending with the already drawn icons.



Figure 3-1
Plain icons



Figure 3-2 L8
overlay
bitmap
applied



Figure 3-3
Outer overlay
bitmap applied



Figure 3-4
White dots as
the selected-
column
indicator



Figure 3-5 Spinning-column screen

The other screen is the payout screen which displays icons and their respective payout amount in a scrollable fashion. The application draws the payout table when the *payout* button is pressed. The button text turns to “exit” whenever the payout table screen is shown and pressing the *exit* button returns to the *spinning-columns* screen. The payout table is displayed on a pixel basis as oppose to the spinning columns' full icon per update so the scrolling would be smooth. The payout table is portrayed as a single image strip with all the icons stacked on top of each other in the order of their bitmap handle. The *topPoint* variable keeps track of the current pixel position in the payout table during scrolling. The construction of the payout table is as follows:

```
int currentMultiplier; /*how many multiples of the current drawing icon*/
int startingX=starting X coordinate of drawing pane;
int startingY;
int topIndex;
static int topPoint=0; /*current pixel location of the table*/
static int movement_shift; /*how much has scrolled*/
static int multiplier=5; /*the multiplier of the current top icon*/

update movement_shift from scrolling;
add or subtract topPoint with movement_shift, depends on movement shift's signage, and then
decrease movement shift;
update multiplier and topPoint according to topPoint's position; /* if(topPoint<0) then increase
multiplier and topPoint=table height + topPoint; if(topPoint>table height) then decrease
multiplier and topPoint%=table height;*/
topIndex=topPoint/ICON_HEIGHT;
startingY=(topPoint%ICON_HEIGHT)*(-1); /*hide part of an icon, if necessary, by drawing it in an
off-screen position. VERTEX2F command is needed to draw negative coordinates*/
currentMultiplier=multiplier;
construct drawing pane to confine the bitmaps;
for(i=0;i<number of icons that can appear in drawing pane;i++){
    cap and update topIndex and currentMultiplier;
    draw the topIndex bitmap handle at startingX and startingY;
    update startingX for the drawing of currentMultiplier and payout amount;
    update startingY for next icon;
    reset startingX for next icon;
    topIndex++;
}
```



```
{
int lookingFor=number of selected columns; /*check for multiples first*/
int tempIndex[total icons]; /*holds the drawn indexes for each column*/
int winningIndex[number of columns]; /*holds the winning bitmap handles in corresponding
column*/
int rewardPoints;
for(i=0;i< number of selected spin columns;i++){
    tempIndex[bitmap handle of this column's drawn index]++;
}

for(i=0;i<total icons;){
    if(tempIndex[i]==lookingFor){
        for(j=0;j<number of selected spin columns;j++){
            if(bitmap handle for the current column's drawn index == i)
                queue bitmap handle in winningIndex; //winning indexes have colored boxes on
top.
        }
        if(lookingFor>2)
            return points earned;

        (lookingFor==2)
            points+=points earned;
        }
    i++;
    if(i==total icons){
        lookingFor--;
        i=0;
        if(lookingFor<2){
            if(points earned > 0)
                return rewardPoints;
            break;
        }
    }
}

check for fruit icons;
return fruit icon points earned;
}
```

3.2.1 Fonts

All fonts used in this application are FT800 built-in fonts. Various fonts are used for the status text on the status bar and in the payout table.

3.2.2 Audio

The sound effects used in this application are from the FT800's sound synthesizer. Sound effects are used during column spinning, reward coin collision, button pressed, and points rewarding.

3.2.3 Buttons

All buttons used in this application are FT800 built-in widget buttons.

4 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited
(USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited
(Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited
(China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

<http://ftdichip.com>

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

Appendix A – References

Document References

Acronyms and Abbreviations

Terms	Description

NOTE – put terms in alphabetical order.

Appendix B – List of Tables & Figures

List of Tables

No table of figures entries found.

List of Figures

Figure 1-1 Slot Machine	2
Figure 2-1 Application flowchart.....	3
Figure 3-1 Plain icons	6
Figure 3-2 L8 overlay bitmap applied.....	6
Figure 3-3 Outer overlay bitmap applied	6
Figure 3-4 White dots as the selected-column indicator	6
Figure 3-5 Spinning-column screen	6
Figure 3-6 Payout screen	8
Figure 3-7 Bouncing coins	8

Appendix C – Revision History

Document Title: AN_306 FT800 Jackpot Application Note
Document Reference No.: FT_001008
Clearance No.: FTDI# 379
Product Page: <http://www.ftdichip.com/FTProducts.htm>
Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	2014-03-25