# **RAi0**<sup>™</sup>

This document is used for the user guide of "ArduinoDue\_SpiFlashProgramWithSdCard". By pressing the push button which is connected on the Arduino Due MCU Board, user is able to start reading the image data from the indicated "All\_Pic.bin" file which is stored in the SD card and write the image data into the serial flash of RA8876/RA8877 through the SPI interface. The programming procedures can be observed through the serial monitor of Arduino IDE. Regarding the detailed information of this application, please refer to the following description:

## Hardware Requirements

- 1. Arduino Due Board
- 2. SD Card Adapter
- 3. Push Button
- 4. RA8876/RA8877 Evaluation Board

For programming the serial flash, RA8876/RA8877 Evaluation Board should previously keep a serial flash programming port and the setting switch "TEST [2:1]" pin.

5. SD card (the maximum memory capacity is 4GB)

#### Circuitry connection:







RA8876 or RA8877 CHIP

Set RA8876 or RA8877 TEST[2:1] PIN to [01]:

Force SPI master interface pin is kept the floating state. (This manipulation is used for insystem programming of serial flash)

Serial Flash ROM for DMA function

SPI Flash external programming port

1	SPI_CS1
2	SPI_SO
3	SPI_SI
4	SPI_SCLK
5	3.3V
6	GND

Note:

1. The SPI interface, GND pin and 3.3V pin should be connected to this programming port. If the RA8876/RA8877 evaluation board has its own dependent power supply, and then the Anduino Due Board should not provide the 3.3V power to RA8876/RA8877 evaluation board.

- 2. RA8876/RA8877 provides 2 SPI master interfaces, it is recommended that CS0 is connected to the Genitop's Font ROM and CS1 is connected to the serial flash.
- 3. When the TEST[2:1] pin of RA8876/RA8877 are set as [01], the SPI master interface pin will stay in the floating state. If the CS0 is connected to Genitop's Font ROM, then the CS0 pin should be added a pull-up resister to the 3.3V, otherwise the Arduino Due might not get the correct programming operation with serial flash through CS1.

#### Software Requirements

Arduino IDE 1.5.7 RA8876 Image\_Tool\_1.0 http://arduino.cc/en/Main/Software www.raio.com.tw

## **Operating Procedures**

- 1. Please connect the hardware connection as the recommended circuitry above.
- 2. Copy the "All\_Pic.bin" and "wp1.bin" files from the "file2sdcard" folder to the SD card in the PC environment. After finishing the file copy operations, please insert the SD card into SD card slot which is already connected to the Arduino Due Board.

Note. The "All\_Pic.bin(28125KB)" and "wp1.bin(938KB)" are the converted BIN files which are converted by the image converting tool "RA8876Image\_Tool\_1.0".

3. Open the project "ArduinoDue\_SpiFlashProgramWithSdCard.ino" and select the flash type. Now Compiles your code and uploads it to the configured board.



#### RA8876 SPI FLASH PROGRAMMER



Note. If the capacity of serial flash is more than 128Mbit (16MByte), please select the flash type as "#define FLASH\_32BIT\_ADDR".

4. Click the serial monitor button in the toolbar for observing the operating status.



# RA8876 SPI FLASH PROGRAMMER

💿 Spi	💀 SpiFlashProgramWithSdCard   Arduino 1.5.7					
File E	idit Sketch Tools Help					
Spi	Auto Format Ctrl+T  Archive Sketch  FlashProgram  Spl. Flash Program	<u>م</u>				
1	1 Alice la 2 and a contract of the second of					
2	#include Sena Monton Containterior	E				
3	finclude -					
4	#include					
5	rincluda Programmer ►					
6	Burn Bootloader					
7	boolean FlashErase(void);					
8	<pre>boolean ByteProgramBinToFlash24BitAddr(char * filename);</pre>					
9	<pre>boolean ByteProgramBinToFlash32BitAddr(char * filename);</pre>					
10	<pre>boolean PageProgramBinToFlash24BitAddr(char * filename);</pre>					
-11	11 boolean PageProgramBinToFlash32BitAddr(char * filename);					
12	2					
13	boolean VerifyDataFlash32BitAddr(c <mark>har</mark> * filename);					
14						
15	/*sd card scs*/					
16	16 const int SD_CARD_SCS = 4;					
17	7 /*spi flash scs*/					
18	<pre>3 const int XNSCS = 52;</pre>					
19	J /*ra88/6 xnscs and xnreset*/					
20	J //const int RASS/D_ANSLS = 52;					
21	$(\text{CONST INT KNOOTO_KAKLSET = 51},)$					
	4	- F				
Done	uploading.					
Verify successful						
Set	Set boot flash true					
CPU :	PU reset.					
24	24 Arduino Due (Programming Port) on CDM9					



#### RA8876 SPI FLASH PROGRAMMER



In the regular case, we should have gotten a Message Box as shown in the figure above. If not, please check the hardware circuitry connection.

5. After pressing the push button, the serial flash programming will be started, the programming procedures includes "Erase", "Program" and "Verify". In this example, is trying to write a 28125KByte image data into the 256Mbit serial flash memory. Please refer to the following table for the related operating time.

Flash erase	60 seconds
Flash erase + Program 28125KByte	375 seconds
Flash erase + Program 28125Kbyte + Verify	848 seconds



The regular operation should be shown as following.

File Edd Sketch Tools Help  SpiFischFrogramWWinBdCar  SpiFischFrogramWWinBdCar  COM9(Arduno Due(Programming Port)  Spi Flash Program SD card initialized SD card initialized SD card initialized SD card initialized File open okl File open okl Program done! File open okl	💿 Sp	iFlashProgramWithSdCare	I Arduino 1.5.7			
PHPADDPOgramWinBolt   2   //Terlfy   3   3   3   4   analogTri   5   6   analogTri   8   6   analogTri   7   7   7   7   8   6   9   7   7   7   8   9   7   9   7   9   7   9   7   9   7   9   7   10   9   7   10   9   7   10   11   12   13   14   10   16   17   18   19   19   10   11   11   12   13   14   10   14   15   16   17   18   19   19   10   10   11   11   12   13   14   16   101   11   12   12   13   14   14   15   15   16   17   101   1	File E	dit Sketch Tools Help				
SpiFlashProgramWMBBCar   02   14   03   16(Verify   04   05   05   06   07   1   18   08   09   11   12   16   16   17   18   19   10   11   11   12   13   14   16   17   18   19   10   11   10   11   101   102   102   103   104   2   Verify states   104   2   Verify states   105   106   11   107   108   109   100   11   101   102   103   104   2   2   105   106   107   107   108   109   101   101   102   103   104   2   2   4   105   106   107   108   109   109   100   101   101   102   103   104	0			2		
82     /*Werlfy       83     if(Werlfy       84     analogWrl       85     else       86     analogWrl       87     }       88     else       81     Flash Program       82     Click button to start flash program!       87     }       88     else       91     Flash erase start       92     #endif       93     Program 32bit address start       94     #ifdef FLASt       95     SFP.setSeria       96     if(PageProgram)       97     {       98     analogWrl       99     /*Verify start       90     if(Perify       91     Program done!       92     #endif       93     Verify start       94     #ifdef FLASt       97     {       98     analogWrl       99     /*Verify       101     analogWrl       102     else       103     analogWrl       104     }       105     Stort flash true       CM wornell     No knowning world       104     }	Spi	FlashProgramWith8dCard				
<pre>83 if(Verif; 94 analogVr; 85 else SD card initialized 86 analogVr; 7 } 88 else Flash erase start 90 analogVr; 91 } Program 32bit address start 92 #endif Pile open ok! 92 #endif Pile open ok! 93 Verify start 94 #ifdef FLASH Verify done! 95 SFP.setSeria 96 if(PageProgr 97 { 98 analogVr; 99 /*Verify 99 /*Verify 101 analogVr; 102 else 103 analogVr; 104 } 104 } 105 descensial 105 Verify successful 105 verify successful 106 verify 107 else verify 108 verify 109 verify 100 if(Verify 100 if(Verify 100 if(Verify 101 analogVr; 104 } 105 verify successful 106 verify 107 verify 108 verify 109 verify 109 verify 100 verify</pre>	82	/*∀erify	COM9 (Arduino Due (Programming Port))	*		
84     analog*r:     Spi Flash Program       85     else     SD card initialized       86     analog*r:     Click buton to start flash program!       87     }     Flash erase start       88     else     Flash erase start       89     {     File open ok!       90     analog*r:     Program done!       91     }     Program done!       92     #endif     Pile open ok!       93     Verify start     Verify done!       94     #ifdef FLASH     Verify done!       95     SFP.setSeria     Program done!       96     if(PageProgram)     Verify done!       97     {       98     analog*r:       99     /*verify       100     if(Verify       101     analog*r:       102     else       103     analog*r:       104     }       105     Verify straccessful       Verify stracesful     Verime       Verify stracesful     Verime	83	<mark>if</mark> (∛erify	Send			
<pre>SD card initialized analogVri } SD card initialized Click button to start flash program! Plash erase start Plash erase done Flaeh erase done Flaeh erase done Flaeh oras done Flaeh oras done Program done! Program done! Program done! Program done! Verify start Verify done! Verify start Verify done! No inf(Verify analogVri floi analogVri floi analogVri floi</pre>	84	analog∛ri	Spi Flash Program			
<pre>86 analogTr: 87 } Flash erase start 88 else Flash erase done 89 { Flie open ok! 90 analogTri 91 } Program done! 92 #endif Flie open ok! 93 Verify start 94 #ifdef FLSH 95 SFP.setSeria 96 if(PageProgr 97 { 98 analogTri 99 /*Verify 100 if(Verify 101 analogTri 102 else 103 analogTri 104 } 104 } 105 Mokeneming 9600 hout • 106 Mokeneming 9600 hout • 107 Flash tree CFU reset. 107 Flash tree 107 Flash t</pre>	85	else	SD card initialized			
87       }       Plash erase start         88       else       Flash erase done         89       {       File open ok!         90       analogTrt       Progran done!         91       >       Progran done!         92       #endif       File open ok!         93       Verify start         94       #ifdef FLASH         95       SFP.setSeria         96       if(PageProgr         97       {         98       analogTrt         99       /*Verify         101       analogTrt         102       else         103       analogTrt         104       }         Verify successful       Velocendi         Set boot flash true       Velocendi	86	analog∛ri	Click button to start flash program!			
<pre>88 else Flash erase done 89 { File open ok! 90 analogUrit Program 32bit address start 91 } Program done! 92 #endif File open ok! 93 Verify start 94 #ifdef FLASH Verify done! 95 SFP.setSeria 96 if(PageProgr 97 { 98 analogUri 99 /*Verify 100 if(Verify 101 analogUri 102 else 103 analogUri 104 } Verify successful Set boot flash true CPU reset.</pre>	87	}	Flash erase start			
<pre>89 { 90 analogUri 90 analogUri 91 } 91 Program 32bit address start 91 } 92 #endif Pile open ok! 92 #endif Pile open ok! 93 Verify start 94 #ifdef FLASH 95 SFP.setSeria 96 if(PageProgr 97 { 98 analogUri 99 /*Verify 100 if(Verify 101 analogUri 102 else 103 analogUri 104 } * * * * * * * * * * * * * * * * * * *</pre>	88	else	Flash erase done	E		
90       analogUrit       Program 32bit address start         91       }       Program done!         92       #cndif       File open ok!         93       Werify start         94       #ifdef FLASH         95       SFP.setSeria         96       if(PageProgram)         97       {         98       analogUrit         99       /*Verify         90       if(Verify)         101       analogUrit         102       else         103       analogUrit         Verify stoccessful       Vetoccoll         Verify stoccessful       Vetoccoll         Verify reset.	89	{	File open ok!			
<pre>91 } Program done! 92 #endif File open ok! 93 Verify start 94 #ifdef FLASH Verify done! 95 SFP.setSeria 96 if(PageProgr 97 { 98 analogWri 99 /*Verify 100 if(Verify 101 analogWri 102 else 103 analogWri 104 } *  Pone uploading Verify stacessful Set boot flash true CPU reset. * </pre>	90	analogWrit	Program 32bit address start			
92  #endif File open ok! 93  Verify start 94  #ifdef FLASH 95  SFP.setSeria 96  if(PageProgr 97  { 98  analogWri 99  /*Verify 100  if(Verify 101  analogWri 102  else 103  analogWri 104  } Verify stccessful Set boot flash true CPU reset.	91	}	Program done!			
93       Verify start         94       #ifdef FLASH         95       SFP.setSeria         96       if(PageProgr         97       {         98       analogWri         99       /*Verify         100       if(Verify         101       analogWri         102       else         103       analogWri         104       }         ✓       ✓         Verify successful       Motecroll         Set boot flash true          CPU reset.       ✓	92	#endif	File open ok!			
94 #ifdef FLASH 95 SFP.setSeria 96 if(PageProgr 97 { 98 analogWri 99 /*Verify 100 if(Verify 101 analogWri 102 else 103 analogWri 104 } ✓ Pone uploading Verify successful Set boot flash true CPU reset.	93		Verify start			
95 SFP.setSeria 96 if(PageProgr 97 { 98 analogWri 99 /*Verify 100 if(Verify 101 analogWri 102 else 103 analogWri 104 } Coneuploading Verify successful Set boot flash true CPU reset.	94	#ifdef FLASH	Verify done!			
96 if(PageProgr 97 { 98 analogWri 99 /*Verify 100 if(Verify 101 analogWri 102 else 103 analogWri 104 } Cone uploading Verify successful Set boot flash true CPU reset.	95	SFP.setSeria				
97 { 98 analogWri 99 /*Verify 100 if(Verify 101 analogWri 102 else 103 analogWri 104 } Done uploading. Verify successful Set boot flash true CPU reset.	96	<mark>if</mark> (PageProgr				
98 analogWri 99 /*Verify 100 if(Verify 101 analogWri 102 else 103 analogWri 104 } Done uploading. Verify successful Set boot flash true CPU reset.	97	{				
99 /*Verify 100 if(Verify 101 analogWri 102 else 103 analogWri 104 } Done uploading. Verify successful Set boot flash true CPU reset.	98	analogWri				
100 if(Verify 101 analogWri 102 else 103 analogWri 104 } Verify successful Set boot flash true CPU reset.	99	/*∀erify				
101 analogWri 102 else 103 analogWri 104 } Chone uploading. Verify successful Set boot flash true CPU reset.	100	<mark>if</mark> (∛erify				
102 else 103 analogWri 104 } Pone uploading. Verify successful Set boot flash true CPU reset.	101	analog⊮ri				
103 analogWri 104 } ✓ Done uploading. Verify successful Set boot flash true CPU reset.	102	else				
104 } Done uploading. Verify successful Set boot flash true CPU reset.	103	analogWri				
Done uploading. Verify successful Set boot flash true CPU reset.	104	}		-		
Done uploading. Verify successful Set boot flash true CPU reset.						
Verify successful (1000 pade ) Set boot flash true CPU reset.	Done	Done uploading.				
Set boot flash true CPU reset. ▼	Verify successful					
CPU reset.	Set boot flash true					
Ť	CPU reset.					
53 Arduino Due (Programming Port) on COM9	53		Arduino Due (Programming Por	t) on CDM9		



If user wants to save the serial flash programming time, and then can skip the verify procedure.



6. If we have done the serial flash programming procedures, we can display the image data on LCD with the "RA8876\_Lite\_DMA.ino".