

Home

STM32F4 links

SPL libs

HAL libs

Tutorials

ESP8266 & ESP32

About

Q

10%

Let's
possi
don

STM32F4 Discovery

Libraries and tutorials for STM32Fxxx series by Tilen Majerle.

Bus
The
For t

Egnyte.com/Fi

About

LinkedIn

Twitter

Google+

Facebook

Github

Instagram

FOLLOW:

f

in

g+

NEXT STORY

Library 05- SPI for STM32F4

PREVIOUS STORY

Library 03- STM32F4 system clock and delay functions

TOP POSTS

STM32 tutorial: Efficiently receive UART data using DMA

STM32F4 External interrupts tutorial

STM32F4 PWM tutorial with TIMERS

All STM32F4 tutorials

All STM32 HAL libraries

Library 21- Read SD card with FatFs on STM32F4

RECENT COMMENTS

STM32 delay ms function : Software delay vs HAL Delay function on HAL library 3- DELAY for STM32Fxxx

Nucleo-L053R8 S2LP Sub-1GHz (Sigfox Geolocation) and BLE Mobile App – PikaNotes on STM32F4 External interrupts tutorial

UART Circular buffer using DMA and IDLE line detection » ControllersTech on STM32 tutorial: Efficiently receive UART data using

STM32F4 / STM32F4 DISCOVERY / STM32F429 / STM32F429 DISCOVERY

370 COMMENTS


Library 04- USART for STM32F4

BY [TILZOR](#) · APRIL 11, 2014

Business File Sharing

Maximize Security Via Encryption, Permission Controls and Auditing.

Egnyte.com/File Sharing



A lot of times when you work on some project, you want to display data on computer. This can be done with USART peripheral on MCU. With USART you can connect more than just computer, you can connect with GSM modules, GPRS, bluetooth and so much more. Our discovery board supports up to **8** USART channels. In this tutorial we will use **USART1** to show principle how to use USART in our project. But first, we have to initialize our pins and peripheral.

	Pins pack 1		Pins pack 2		Pins pack 3		
U(S)ARTx	TX	RX	TX	RX	TX	RX	APB
USART1	PA9	PA10	PB6	PB7			2
USART2	PA2	PA3	PD5	PD6			1
USART3	PB10	PB11	PC10	PC11	PD8	PD9	1
UART4	PA0	PA1	PC10	PC11			1
UART5	PC13	PD3					1

MORE

SUBSCRIBE

Subscribe if you want to be notified about new posts and other events on this site.

Name

Email*

Subscribe

POLL

Standard peripheral drivers vs HAL drivers

☐ I like standard peripheral drivers and I want to stay on them.

☐ I want to start with new HAL system

☐ What are standard peripheral drivers?





☐ What are HAL drivers?

Vote

View Results

SUPPORT

Donate



This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish.

Accept

[Read More](#)

1 of 9

2020-02-09 22:00

Internal RTC in STM32 » ControllersTech
on Library 19- Use internal RTC on
STM32F4

STM32F407 RFID project full source code
– Home on Library 23- Read RFID tag with
MF RC522 on STM32F4

CATEGORIES

ARM Cortex-M

ESP8266 & ESP32

Random

STM32F4

STM32F4 Discovery

STM32F429

STM32F429 Discovery

STM32F7

STM32F7 Discovery

Tutorials

Uncategorized



RECENT POSTS

Manage embedded software libraries
with STM32CubeMX

Getting started with STM32 step-by-step

C code style and coding rules

Tutorial: Control WS2812B leds with
STM32

1-Wire over UART

ST acquired Atollic to enrich STM32
ecosystem

CALENDAR

April 2014

M	T	W	T	F	S	S
	1	2	3	4	5	6

UART7	PE8	PE7	PF7	PF6	1
UART8	PE1	PE0			1

USART1 uses pins

- PA9 for transmitting data
- PA10 for receiving data

so they must have enabled clock, set for alternating function and set them this alternating function.

```
1 GPIO_InitTypeDef GPIO_InitStructure;  
2  
3 // Enable clock for GPIOA  
4 RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);  
5  
6 /**  
7 * Tell pins PA9 and PA10 which alternating function you will use  
8 * @important Make sure, these lines are before pins configuration!  
9 */  
10 GPIO_PinAFConfig(GPIOA, GPIO_PinSource9, GPIO_AF_USART1);  
11 GPIO_PinAFConfig(GPIOA, GPIO_PinSource10, GPIO_AF_USART1);  
12 // Initialize pins as alternating function  
13 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9 | GPIO_Pin_10;  
14 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;  
15 GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;  
16 GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;  
17 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;  
18 GPIO_Init(GPIOA, &GPIO_InitStructure);
```

Then, when you initialized pins, we have to initialize USART peripheral too

```
1 USART_InitTypeDef USART_InitStructure;  
2 NVIC_InitTypeDef NVIC_InitStructure;  
3  
4 /**  
5 * Enable clock for USART1 peripheral  
6 */  
7 RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);  
8  
9 /**  
10 * Set Baudrate to value you pass to function  
11 * Disable Hardware Flow control  
12 * Set Mode To TX and RX, so USART will work in full-duplex mode  
13 * Disable parity bit  
14 * Set 1 stop bit  
15 * Set Data bits to 8  
16 */  
17 * Initialize USART1  
18 * Activate USART1  
19 */  
20 USART_InitStructure.USART_BaudRate = baudrate;  
21 USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;  
22 USART_InitStructure.USART_Mode = USART_Mode_Tx | USART_Mode_Rx;  
23 USART_InitStructure.USART_Parity = USART_Parity_No;  
24 USART_InitStructure.USART_StopBits = USART_StopBits_1;  
25 USART_InitStructure.USART_WordLength = USART_WordLength_8b;  
26 USART_Init(USART1, &USART_InitStructure);  
27 USART_Cmd(USART1, ENABLE);  
28  
29 /**  
30 * Enable RX interrupt  
31 */  
32 USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);  
33  
34 /**  
35 * Set Channel to USART1  
36 * Set Channel Cmd to enable. That will enable USART1 channel in NVIC  
37 * Set Both priorities to 0. This means high priority  
38 */  
39 * Initialize NVIC  
40 */  
41 NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;  
42 NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;  
43 NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;  
44 NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;  
45 NVIC_Init(&NVIC_InitStructure);
```

Because our USART uses interrupt for incoming data, we have to initialize **NVIC** (Nested Vector Interrupt Controller). Everytime when data will come, our main program will stop executing and our interrupt routine will jump in.

Library

Features

- Operate with up to 8 U(S)ARTs in UART mode
- Selectable pins for USART
- Receive interrupt handler

SEARCH

Q To search type and hit enter

TWITTER

Tweets by @tilz0R



Tilen Majerle
@tilz0R

Despite its humble name, LwMEM is advanced and portable library for managing dynamic memory, optimized for embedded systems.
[#embedded](#) [#memory](#)
github.com/Majerle/lwmem

Majerle/lw...
Lightweight...
github.com

May 3, 2019



Tilen Majerle
@tilz0R

Learn how to write clean and clear C code with this coding style tutorial.
github.com/Majerle/c_code...
[#C](#) [#languageC](#)
[#codingrules](#) [#programmer](#)
[#realife](#)

[Embed](#) [View on Twitter](#)

GOOGLE+

Follow me on Google+

M	T	W	T	F	S	S
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
					May »	

ARCHIVES

October 2018

September 2018

June 2018

April 2018

December 2017

November 2017

July 2017

April 2017

March 2017

November 2016

July 2016

April 2016

January 2016

December 2015

November 2015

October 2015

September 2015

August 2015

July 2015

June 2015

May 2015

April 2015

March 2015

February 2015

January 2015

December 2014

November 2014

October 2014

September 2014

August 2014

July 2014

June 2014

■ CMSIS

- STM32F4xx
- STM32F4xx RCC
- STM32F4xx GPIO
- STM32F4xx USART

■ TM

- defines.h



defines.h configuration example

134865 downloads

0.00 KB

Download

- attributes.h



attributes.h

22810 downloads

0.85 KB

Download

- TM GPIO



TM STM32F4 GPIO Library

68070 downloads

0.00 KB

Download

Functions and enumerations

```
1 /**
2  * @defgroup TM_USART_Typedefs
3  * @brief USART Typedefs
4  * @{
5  */
6
7 /**
8  * @brief USART PinsPack enumeration to select pins combination
9  */
10 typedef enum {
11     TM_USART_PinsPack_1,      /*!< Select PinsPack1 from Pinout table
12     TM_USART_PinsPack_2,      /*!< Select PinsPack2 from Pinout table
13     TM_USART_PinsPack_3,      /*!< Select PinsPack3 from Pinout table
14     TM_USART_PinsPack_Custom /*!< Select custom pins for specific
15 } TM_USART_PinsPack_t;
16
17 /**
18  * @brief USART Hardware flow control selection
19  * @note Corresponding pins must be initialized in case you don't
20  */
21 typedef enum {
22     TM_USART_HardwareFlowControl_None = 0x0000, /*!< No flow control
23     TM_USART_HardwareFlowControl_RTS = 0x0100, /*!< RTS flow control
24     TM_USART_HardwareFlowControl_CTS = 0x0200, /*!< CTS flow control
25     TM_USART_HardwareFlowControl_RTS_CTS = 0x0300 /*!< RTS and CTS
26 } TM_USART_HardwareFlowControl_t;
27
28 /**
29  * @}
30  */
31
32 /**
33  * @defgroup TM_USART_Functions
34  * @brief USART Functions
35  * @{
36  */
37
38 /**
39  * @brief Initializes USARTx peripheral and corresponding pins
40  * @param *USARTx: Pointer to USARTx peripheral you will use
41  * @param pinspack: This parameter can be a value of @ref TM_USART_PinsPack_t
42  * @param baudrate: Baudrate number for USART communication
43  * @retval None
44  */
45 void TM_USART_Init(TM_USART_TypeDef* USARTx, TM_USART_PinsPack_t pinspack, uint32_t baudrate);
46
47 /**
48  * @brief Initializes USARTx peripheral and corresponding pins with hardware flow control
49  * @note Hardware flow control pins are not initialized. Easy to use with software flow control.
50  * @param *USARTx: Pointer to USARTx peripheral you will use
51  * @param pinspack: This parameter can be a value of @ref TM_USART_PinsPack_t
52  * @param baudrate: Baudrate number for USART communication
53  * @param FlowControl: Flow control mode you will use. This parameter can be a value of @ref TM_USART_HardwareFlowControl_t
54  * @retval None
55  */
56 void TM_USART_InitWithFlowControl(TM_USART_TypeDef* USARTx, TM_USART_PinsPack_t pinspack, uint32_t baudrate, TM_USART_HardwareFlowControl_t FlowControl);
57
58 /**
59  * @brief Puts character to USART port
60  * @param *USARTx: Pointer to USARTx peripheral you will use
61  * @param c: character to be send over USART
62  * @retval None
63  */
64 static __INLINE void TM_USART_Putc(TM_USART_TypeDef* USARTx, volatile uint8_t c)
65 {
66     /* Check USART */
67     if ((USARTx->CR1 & USART_CR1_UE) == 0) {
68         /* Wait to be ready, buffer empty */
69         USART_WAIT(USARTx);
70         /* Send data */
71         USART->DR = (uint16_t)(c & 0x01FF);
72         /* Wait to be ready, buffer empty */
73         USART_WAIT(USARTx);
74     }
75 }
76 /**
```

May 2014

April 2014

```

80  * @retval None
81  */
82  void TM_USART_Puts(USART_TypeDef* USARTx, char* str);
83
84  /**
85   * @brief Sends data array to USART port
86   * @param *USARTx: Pointer to USARTx peripheral you will use
87   * @param *dataArray: Pointer to data array to be sent over USART
88   * @param count: Number of elements in data array to be send over
89   * @retval None
90   */
91  void TM_USART_Send(USART_TypeDef* USARTx, uint8_t* dataArray, uint16_t count);
92
93  /**
94   * @brief Gets character from internal USART buffer
95   * @param *USARTx: Pointer to USARTx peripheral you will use
96   * @retval Character from buffer, or 0 if nothing in buffer
97   */
98  uint8_t TM_USART_Getc(USART_TypeDef* USARTx);
99
100 /**
101  * @brief Get string from USART
102  *
103  * This function can create a string from USART received data
104  *
105  * It generates string until "\n" is not recognized or buffer is full
106  *
107  * @note As of version 1.5, this function automatically adds 0x00 at the end of the string
108  * @param *USARTx: Pointer to USARTx peripheral you will use
109  * @param *buffer: Pointer to buffer where data will be stored from USART
110  * @param bufsize: maximal number of characters we can add to your buffer
111  * @retval Number of characters in buffer
112  */
113  uint16_t TM_USART_GetString(USART_TypeDef* USARTx, char* buffer, uint16_t bufsize);
114
115 /**
116  * @brief Check if character c is available in internal buffer
117  * @param *USARTx: Pointer to USARTx peripheral you will use
118  * @param c: character to check if it is in USARTx's buffer
119  * @retval Character status:
120  * - 0: Character was not found
121  * - > 0: Character has been found in buffer
122  */
123  uint8_t TM_USART_FindCharacter(USART_TypeDef* USARTx, uint8_t c);
124
125 /**
126  * @brief Checks if internal USARTx buffer is empty
127  * @param *USARTx: Pointer to USARTx peripheral you will use
128  * @retval Buffer empty status:
129  * - 0: Buffer is not empty
130  * - > 0: Buffer is empty
131  */
132  uint8_t TM_USART_BufferEmpty(USART_TypeDef* USARTx);
133
134 /**
135  * @brief Checks if internal USARTx buffer is full
136  * @param *USARTx: Pointer to USARTx peripheral you will use
137  * @retval Buffer full status:
138  * - 0: Buffer is not full
139  * - > 0: Buffer is full
140  */
141  uint8_t TM_USART_BufferFull(USART_TypeDef* USARTx);
142
143 /**
144  * @brief Clears internal USART buffer
145  * @param *USARTx: Pointer to USARTx peripheral you will use
146  * @retval None
147  */
148  void TM_USART_ClearBuffer(USART_TypeDef* USARTx);
149
150 /**
151  * @brief Sets custom character for @ref TM_USART_Getc() function
152  * @param *USARTx: Pointer to USARTx peripheral you will use
153  * @param Character: Character value to be used as string end
154  * @note Character will also be added at the end for your buffer
155  * @retval None
156  */
157  void TM_USART_SetCustomStringEndCharacter(USART_TypeDef* USARTx, uint8_t Character);
158
159 /**
160  * @brief Callback for custom pins initialization for USARTx.
161  * When you call TM_USART_Init() function, and if you pass custom pins,
162  * then this function will be called where you can initialize custom pins.
163  * @note With __weak parameter to prevent link errors if not defined
164  * @param *USARTx: Pointer to USARTx peripheral you will use for initialization
165  * @retval None
166  */
167  void TM_USART_InitCustomPinsCallback(USART_TypeDef* USARTx);
168
169 /**
170  * @brief Callback function for receive interrupt on USART1 in code
171  * @note With __weak parameter to prevent link errors if not defined
172  * @param c: character received via USART
173  * @retval None
174  */
175  __weak void TM_USART1_ReceiveHandler(uint8_t c);
176
177 /**
178  * @brief Callback function for receive interrupt on USART2 in code
179  * @note With __weak parameter to prevent link errors if not defined
180  * @param c: character received via USART
181  * @retval None
182  */
183  __weak void TM_USART2_ReceiveHandler(uint8_t c);
184
185 /**
186  * @brief Callback function for receive interrupt on USART3 in code
187  * @note With __weak parameter to prevent link errors if not defined
188  * @param c: character received via USART
189  * @retval None
190  */
191  __weak void TM_USART3_ReceiveHandler(uint8_t c);
192
193 /**

```

This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. [Accept](#) [Read More](#)

```

197  * @retval None
198  */
199  __weak void TM_UART4_ReceiveHandler(uint8_t c);
200
201  /**
202  * @brief Callback function for receive interrupt on UART5 in case of RXNE
203  * @note With __weak parameter to prevent link errors if not defined
204  * @param c: character received via USART
205  * @retval None
206  */
207  __weak void TM_UART5_ReceiveHandler(uint8_t c);
208
209  /**
210  * @brief Callback function for receive interrupt on USART6 in case of RXNE
211  * @note With __weak parameter to prevent link errors if not defined
212  * @param c: character received via USART
213  * @retval None
214  */
215  __weak void TM_USART6_ReceiveHandler(uint8_t c);
216
217  /**
218  * @brief Callback function for receive interrupt on UART7 in case of RXNE
219  * @note With __weak parameter to prevent link errors if not defined
220  * @param c: character received via USART
221  * @retval None
222  */
223  __weak void TM_UART7_ReceiveHandler(uint8_t c);
224
225  /**
226  * @brief Callback function for receive interrupt on UART8 in case of RXNE
227  * @note With __weak parameter to prevent link errors if not defined
228  * @param c: character received via USART
229  * @retval None
230  */
231  __weak void TM_UART8_ReceiveHandler(uint8_t c);
232
233  /**
234  * @}
235  */

```

Hardware

Your computer probably has not RS232 connector, but it has USB for sure. You need **USB to serial** converter. I prefer [FTDI](#) which is very stable and support 3V3 levels. You can buy converter on ebay for about 5\$. **Connect converter's TX with board's RX and converter's RX with board's TX.** To display data, you need some terminal, I use one from [Bray](#). Open program and set:

- Select your COM port
- baudrate to **9600**
- Data bits to **8**
- Parity to **none**
- Stop bits to **1**

Example

```

1  /**
2  *   Keil project for USART
3  *
4  *   Before you start, select your target, on the right of the "Load
5  *
6  *   @author      Tilen Majerle
7  *   @email       tilen@majerle.eu
8  *   @website     http://stm32f4-discovery.net
9  *   @ide         Keil uVision 5
10  *   @packs       STM32F4xx Keil packs version 2.2.0 or greater
11  *   @stdperiph   STM32F4xx Standard peripheral drivers version 1
12  */
13  /* Include core modules */
14  #include "stm32f4xx.h"
15  /* Include my libraries here */
16  #include "defines.h"
17  #include "tm_stm32f4_usart.h"
18
19  int main(void) {
20      uint8_t c;
21
22      /* Initialize system */
23      SystemInit();
24
25      /* Initialize USART1 at 9600 baud, TX: PB6, RX: PB7 */
26      TM_USART_Init(USART1, TM_USART_PinsPack_2, 9600);
27
28      /* Put string to USART */
29      TM_USART_Puts(USART1, "Hello world\n\r");
30
31      while (1) {
32          /* Get character from internal buffer */
33          c = TM_USART_Getc(USART1);
34          if (c) {
35              /* If anything received, put it back to terminal */
36              TM_USART_Putc(USART1, c);
37          }
38      }
39  }

```

Project available on [Github](#), download library below.



USART library for all USART peripherals

Download

9.54 KB

24238 downloads

Share this to other users:



Tags: coide coocox gpio library microcontrollers stm32f429
stm32f429discovery stm32f4discovery stm32f4xx uart usart usart library

**tilzOR**

Owner of this site. Application engineer, currently employed by STMicroelectronics. Exploring latest technologies and owner of different libraries posted on Github.

See How to Log In Python

Free Guide to Quickly Learn What to Log and How to Log It Correctly in Python

YOU MAY ALSO LIKE...



71

How to properly set clock speed for STM32F4xx devices

22 JAN, 2015



99

HAL Library 21- Multi purpose USB library for STM32Fxxx

10 AUG, 2015



3

HAL library 09- GENERAL for STM32Fxxx

29 JUL, 2015

Read before commenting!



Before you make a new comment, make sure you agree with things listed below:

- - Read post to make sure if it is already posted what you are asking for,
- - Make sure you have the latest version of libraries used in your project,
- - Make a clean and grammatically correct written message,
- - Report as many details as possible, including what have you done so far,
- - Do NOT post any code here. Use [Pastebin](#),
- - Do NOT post any error codes here. Use [Pastebin](#),
- - Specify STM32Fxxx family and used Discovery/EVAL/Nucleo or custom made board,
- - Make sure your clock is set correct for PLL,
- - If you are using my HAL drivers, please [check this post](#) how

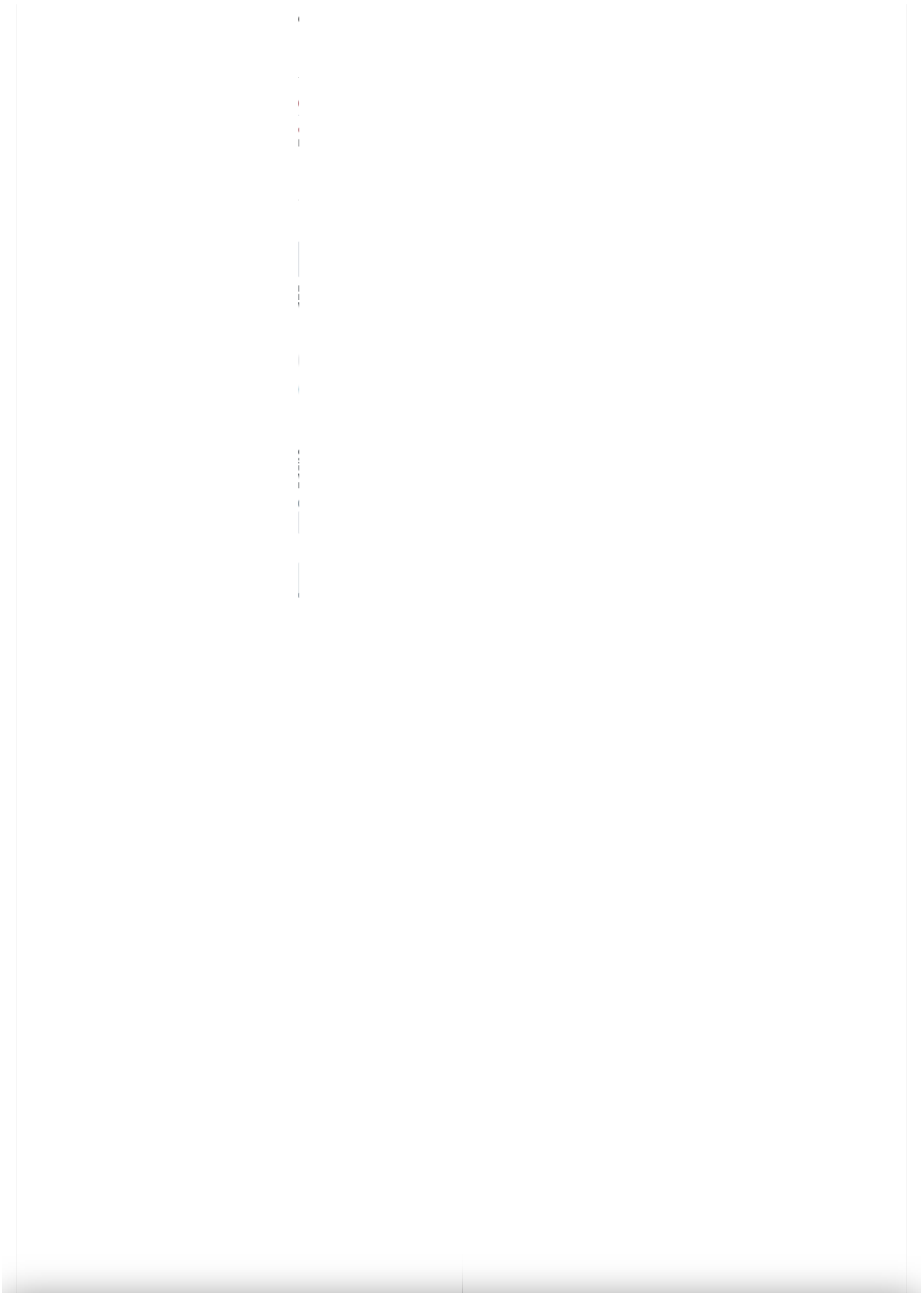
This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. [Accept](#) [Read More](#)

Comment will be deleted on breaking these rules without notification!

How to Fix Thinning Hair

Have You Tried
Everything To Grow
Stronger Fuller Hair.
See The 3 Things You
Need To Avoid

Hair La Vie



This website uses cookies to improve your experience. We'll assume you're ok with this, but you can opt-out if you wish. [Accept](#) [Read More](#)

Today's Refinance Rate

3.07%

APR 15 Year Fixed

Select Loan Amount

\$

\$225.000

[About](#)
[LinkedIn](#)
[Twitter](#)
[Google+](#)
[Facebook](#)
[Github](#)
[Instagram](#)



STM32F4 Discovery © 2020. All Rights Reserved.

Copying content from this site is stricly forbidden. This does not apply for library downloads.

