

STM32Discovery GT-CP Code Library Evaluation Instructions

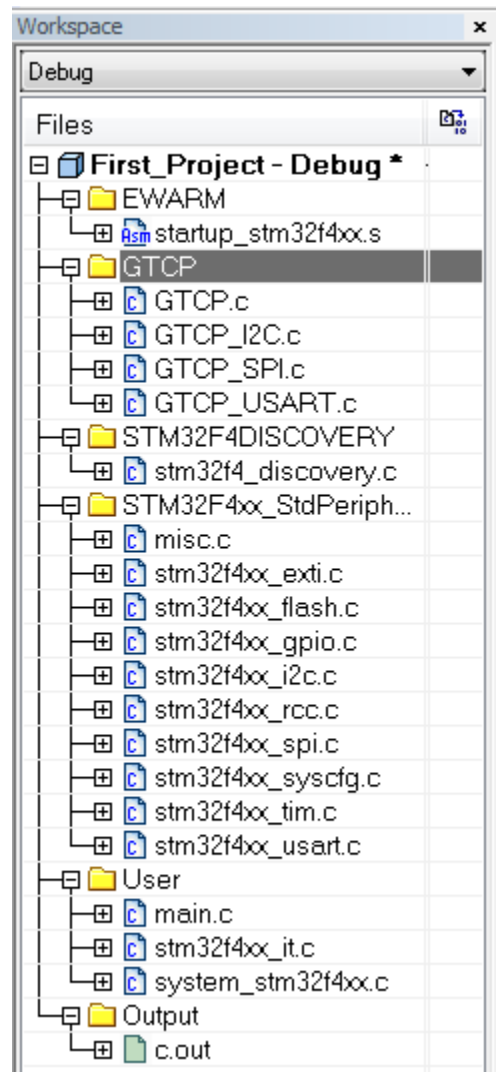
Getting Started

Preface: This library only works with the STM32F407 MCU on the STM32F4-Discovery board.

1. **Download** IAR Embedded Workbench for ARM free trial from: <https://www.iar.com/iar-embedded-workbench/#!?currentTab=free-trials>.
 - a. You will need to **register** with IAR to get a free trial registration key. Be sure to **choose** the **size limited** free trial.
2. **Install** IAR Embedded Workbench for ARM to your workstation.
3. **Launch** IAR Embedded Workbench for ARM and **enter** your free trial registration key if you have not already done so.
4. **Follow** Haithem Chamkhi's video to see how to set-up IAR Embedded Workbench for ARM for STM32Discovery.
 - a. <https://www.youtube.com/watch?v=mDxn-wdixoM>
5. **Move** all of the GT-CP code library files into your project directory (same folder as your main.c file in Windows Explorer). This will automatically include all of the .h files to your project.
 - a. This is only true if you set your project up properly and put \$PROJ_DIR\$ in your additional include directory as instructed by Haithem's tutorial.
6. **Create** a "GTCP" folder in your project workspace.
7. **Right click** on the project node in your workspace, choose **Add > Add Files...** and **add** all of the .c files from the GT-CP code library.
8. **Move** all of the GT-CP code library's .c files to the "GTCP" folder in your project workspace.
9. You will also need to **add** a few peripheral files to your project.
 - a. Make sure these files are included in the STM32F4xx_StdPeriph_Driver folder:
 - i. misc.c
 - ii. stm32f4xx_exti.c
 - iii. stm32f4xx_flash.c
 - iv. stm32f4xx_gpio.c
 - v. stm32f4xx_i2c.c
 - vi. stm32f4xx_rcc.c
 - vii. stm32f4xx_spi.c
 - viii. stm32f4xx_syscfg.c
 - ix. stm32f4xx_tim.c
 - x. stm32f4xx_usart.c
10. **Right click** on the project node, **choose** Options..., **click** on the C/C++ Compiler category, **navigate** to the preprocessor tab, **add** "**HSE_VALUE=8000000**" to the defined symbols section, and **click** "OK". This slows down the STM32's clock speed to match the module's baud rate.
11. You should now be ready to program for your GT-CP module!

Hello World! UART Program

1. Make sure your main.c file has the following includes:
 - a. `#include "stm32f4xx.h"`
 - b. `#include "stm32f4_discovery.h"`
 - c. `#include "stdio.h"`
 - d. `#include "stdint.h"`
 - e. `#include <stddef.h>`
 - f. `#include "GTCP.h"`
 - g. `#include "GTCP_I2C.h"`
 - h. `#include "GTCP_SPI.h"`
 - i. `#include "GTCP_USART.h"`
 - j. Your project workspace should look like the following:



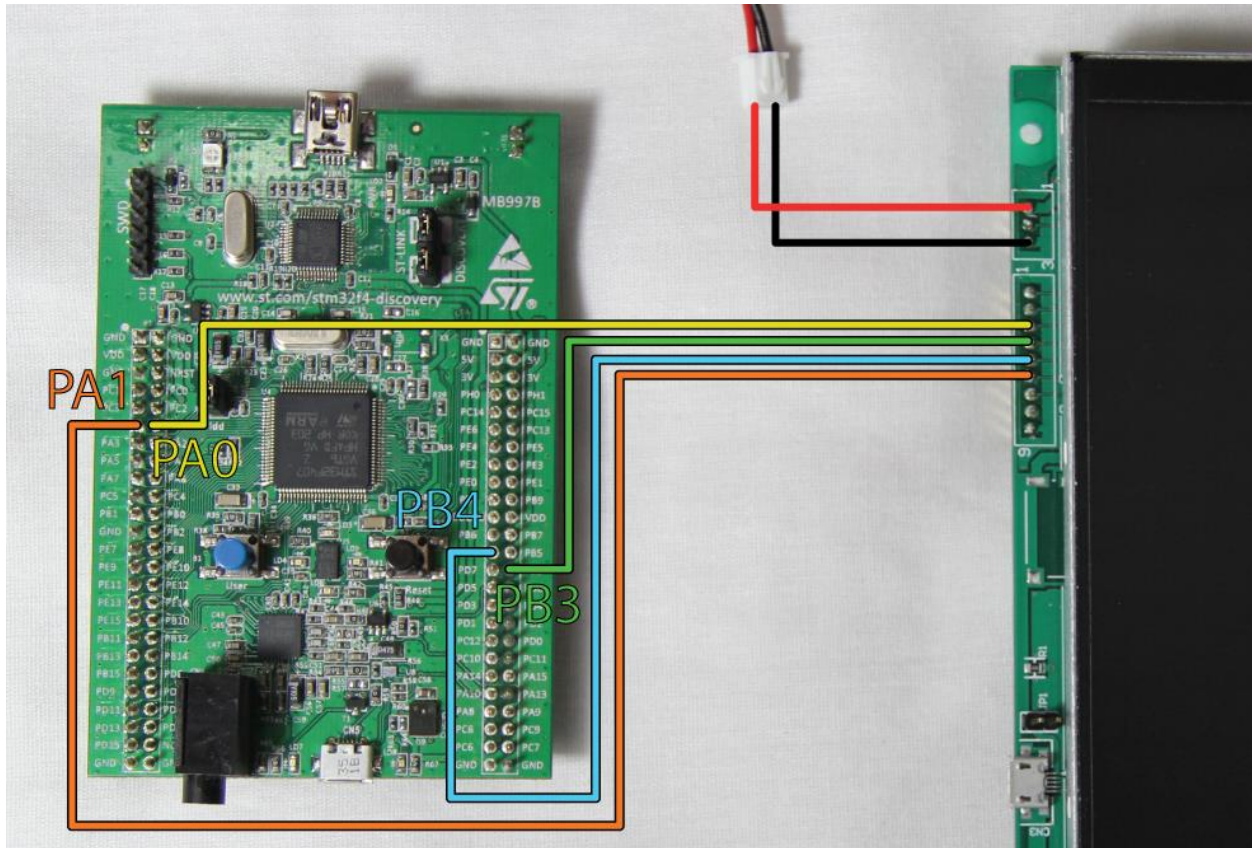
2. In main, **add** these lines of code:
 - a. `GTCP_UART_Init(38400);`

- i. The default baud rate for the GT-CP modules is **38400**. Change this value accordingly if you have modified your module's baud rate setting.
 - b. `GTCP_Init(width, height);`
 - i. The width and height values correspond to the width and height of dots/pixels on your display module. This can be seen in the module's product number (for example, GT**800**X**480**A-C903P has a width of **800** and height of **480**).
 - c. `printUART("Hello World");`
3. **Save** your "main.c" file, **connect** your STM32Discovery to your workstation and **click** on the "Download and Debug" button (the green triangle).
 4. Once your program has been sent to your STM32Discovery, **disconnect** it from your workstation.
 5. **Connect** your STM32Discovery to your GT-CP module as stated below:
 - a. The STM32Discovery should **not** be receiving any power yet.
 - b. **Connect** your GT-CP module to your STM32Discovery in the following way:

Power (CN8)	
GT-CP Pin	Pin
1	External 5V
2	Nothing
3	GND

Serial Interface (CN9)	
GT-CP Pin	STM32f4 Pin
1	Nothing
2	Nothing
3	PA0
4	PB2
5	PB4
6	PA1
7	Nothing
8	Nothing
9	Nothing

c. You can also **refer** to the following image:



6. Additionally, make sure the jumpers (J0 to J4) on the back of the display module are configured as follows:

UART Jumper Setting	
Jumper	Setting
J3	OPEN
J4	OPEN
J5	SHORT
J6	SHORT

7. Make sure that JP1 is **open**. Additionally, make sure your external power source can produce enough **current** for your module.
 - a. You can find the current requirement in your module's hardware datasheet.
8. Once your module is connected to your STM32Discovery, **give** the module power and then **connect** your STM32Discovery to your workstation.
9. You should see "Hello World" on your module!

Troubleshooting

10. If "Hello World" does not immediately show up on your module:
 - a. Try pressing the RESET button on the STM32Discovery.

- b. Make sure JP1 is open. This allows for the module to take power from an external source.