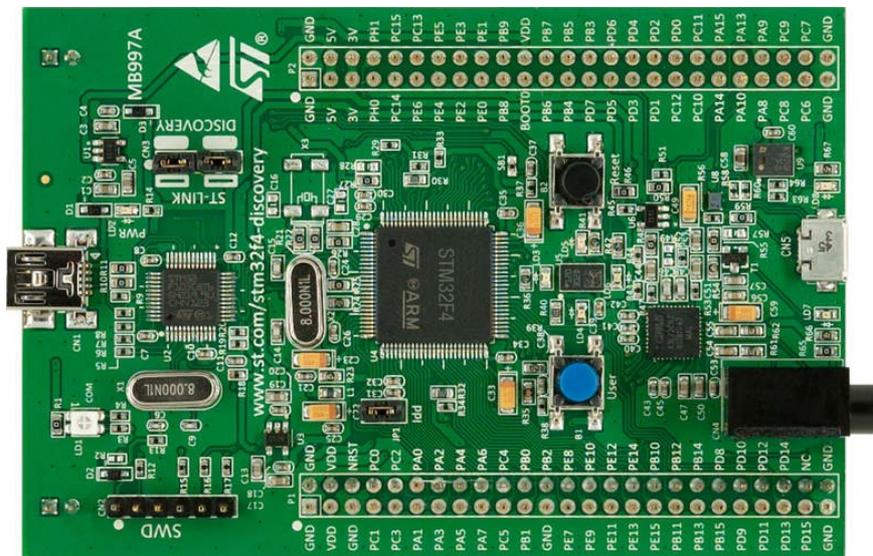


Guía de iniciación al kit de evaluación

# St STM32F4 Discovery



Revisión: 2012/11/09

**Este documento se ha escrito con permiso de Stmicroelectronics y se basa en la guías:**

UM1467: Getting started with software and firmware environments for the STM32F4DISCOVERY Kit

UM1472: STM32F4DISCOVERY STM32F4 high-performance discovery board



# Índice de contenido

1	Introducción.....	3
2	Características.....	3
3	El microcontrolador STM32F407VGT6 .....	4
4	Requisitos y conexión del kit.....	7
5	El depurador ST-LINK/V2.....	7
5.1	Uso en Keil MDK-ARM.....	8
5.1.1	Instalación.....	8
5.1.2	Configurando las opciones de depuración.....	8
5.1.3	Depuración más avanzada.....	10
6	Uso del entorno de desarrollo MDK-ARM (Keil <sup>™</sup> ).....	10
6.1	Versión demo de Keil.....	10
6.2	Construir un proyecto ya existente.....	10
6.3	Ejecutar y depurar un proyecto MDK-ARM.....	11
6.4	Crear un proyecto con el entorno MDK-ARM.....	12
6.4.1	Crear el proyecto.....	12
6.4.2	Configurar el proyecto.....	13
6.4.3	Añadiendo archivos fuente.....	14
7	El “STM32F4DISCOVERY board firmware package.....	16
7.1	Carpeta libraries.....	16
7.1.1	Subcarpeta CMSIS.....	16
7.1.2	STM32F4xx_StdPeriph_Driver.....	17
7.2	Carpeta de proyecto.....	17
7.2.1	Subcarpeta demonstration .....	17
7.2.2	Subcarpeta Peripheral_Examples.....	17
7.3	Carpeta Utilities.....	17
8	Esquema eléctrico.....	19



# 1 Introducción

La placa *STM32F4DISCOVERY* permite descubrir las características del STM32F4 y desarrollar aplicaciones.

Esta guía pretende simplificar al máximo la tarea de toma de contacto e iniciación con este kit. Una vez iniciados, lo aconsejable es desechar esta guía y basarse en los documentos originales de St referenciados en la portada.

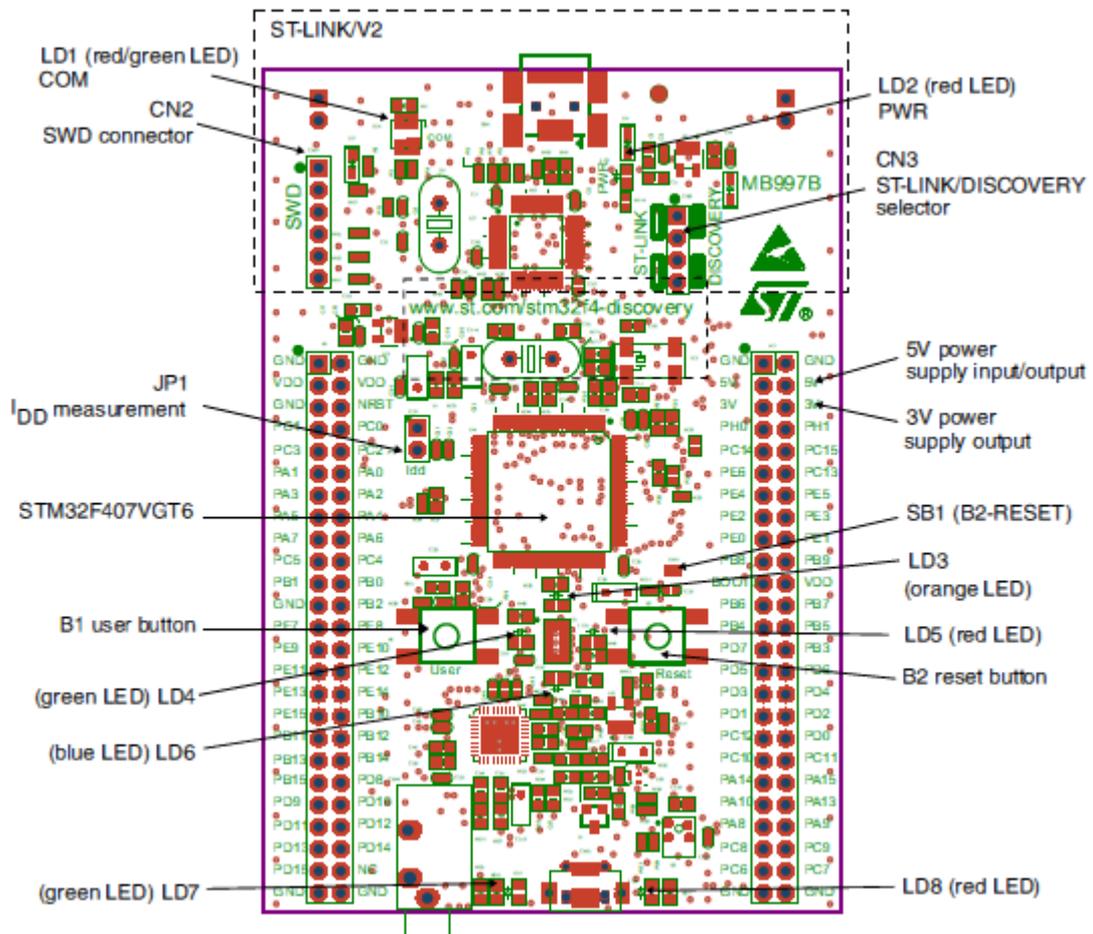
Suerte.

## 2 Características

La placa *STM32F4DISCOVERY* ofrece las siguientes características:

- 1 Microcontrolador *STM32F407VGT6* con 1 MB de memoria flash, 192 KB de RAM, encapsulado *LQFP100*.
- 2 *ST-LINK/V2* incorporado con selector usar el kit como un *ST-LINK/V2* independiente (con conector *SWD* para programación y depuración).
- 3 Fuente de alimentación: a través del bus USB o desde una fuente de alimentación externa de 5V.
- 4 Sensor de movimiento *ST MEMS LIS302DL*, acelerómetro con salida digital de 3 ejes
- 5 Sensor de audio *ST MEMS MP45DT02*, micrófono digital omnidireccional
- 6 Audio *DAC CS43L22* con controlador integrado de altavoz clase D
- 7 Ocho LEDs:
  - LD1 (rojo / verde) para la comunicación USB
  - LD2 (rojo) alimentación 3,3 V
  - Cuatro LEDs de usuario, LD3 (naranja), LD4 (verde), LD5 (rojo) y LD6 (azul)
  - 2 LEDs *USB OTG* LD7 (verde), VBus y LD8 (rojo)
- 8 Dos pulsadores (usuario y reset)
- 9 *USB OTG* con conector micro-AB





Nota: El pin 1 de CN2, CN3, JP1, y de los conectores P1 y P2 están indicados mediante un cuadro.

Figura 3. STM32F4DISCOVERY Layout

### 3 El microcontrolador STM32F407VGT6

Este procesador *ARM Cortex-M4 32-bit* con *FPU* tiene 210 DMIPS, 1 MB Flash, 196 KB RAM, *USB OTG HS/FS*, *Ethernet*, 17 TIMs y 3 ADCs.

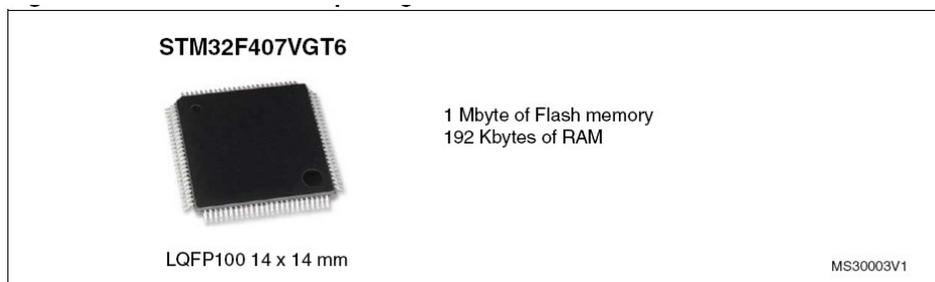


Figura 4  
1 Mbyte de memoria Flash  
192 Kbytes de RAM

#### STM32F407VGT6

Este dispositivo proporciona las siguientes características:

- 168 MHz/210 DMIPS Cortex-M4 con un solo ciclo DSP MAC y unidad de coma flotante:



Ejecución mejorada de algoritmos de control

Facilidad de uso

Mejor eficiencia de código

Reducción del *"time to market"*

- Diseñado para un alto rendimiento y muy elevada transferencia de datos; Acelerador ART, 32-bit, AHB bus matriz de 7 capas con 7 maestros y 8 esclavos, incluyendo 2 bloques de SRAM, controladores DMA: 2 de propósito general, uno para USB HS, uno para Ethernet. Un bloque de SRAM dedicado al core.
- Eficiencia energética, ultra-bajo consumo de energía, *RTC* <1  $\mu$ A en modo *VBAT*, de 3,6 V hasta 1,7 V *VDD*, el regulador de tensión con capacidad de potencia adaptable, proporcionando una mayor flexibilidad para reducir el consumo de energía en aplicaciones que requieren tanto mucho procesamiento y baja consumo funcionando en baja tensión o con baterías recargables.
- Integración máxima: Hasta 1 Mbyte de memoria Flash on-chip, 192 Kbytes de SRAM, circuito de reset, RCs interno, PLLs, encapsulado *WLCSP* disponible, etc.
- Los periféricos innovadores que ofrecen nuevas posibilidades para conectar y comunicar datos a alta velocidad, así como mayor precisión debido a su alta resolución.
- Extensas herramientas y soluciones software que proporcionan una amplia gama dentro de los ecosistemas *STM32* para desarrollar tus aplicaciones



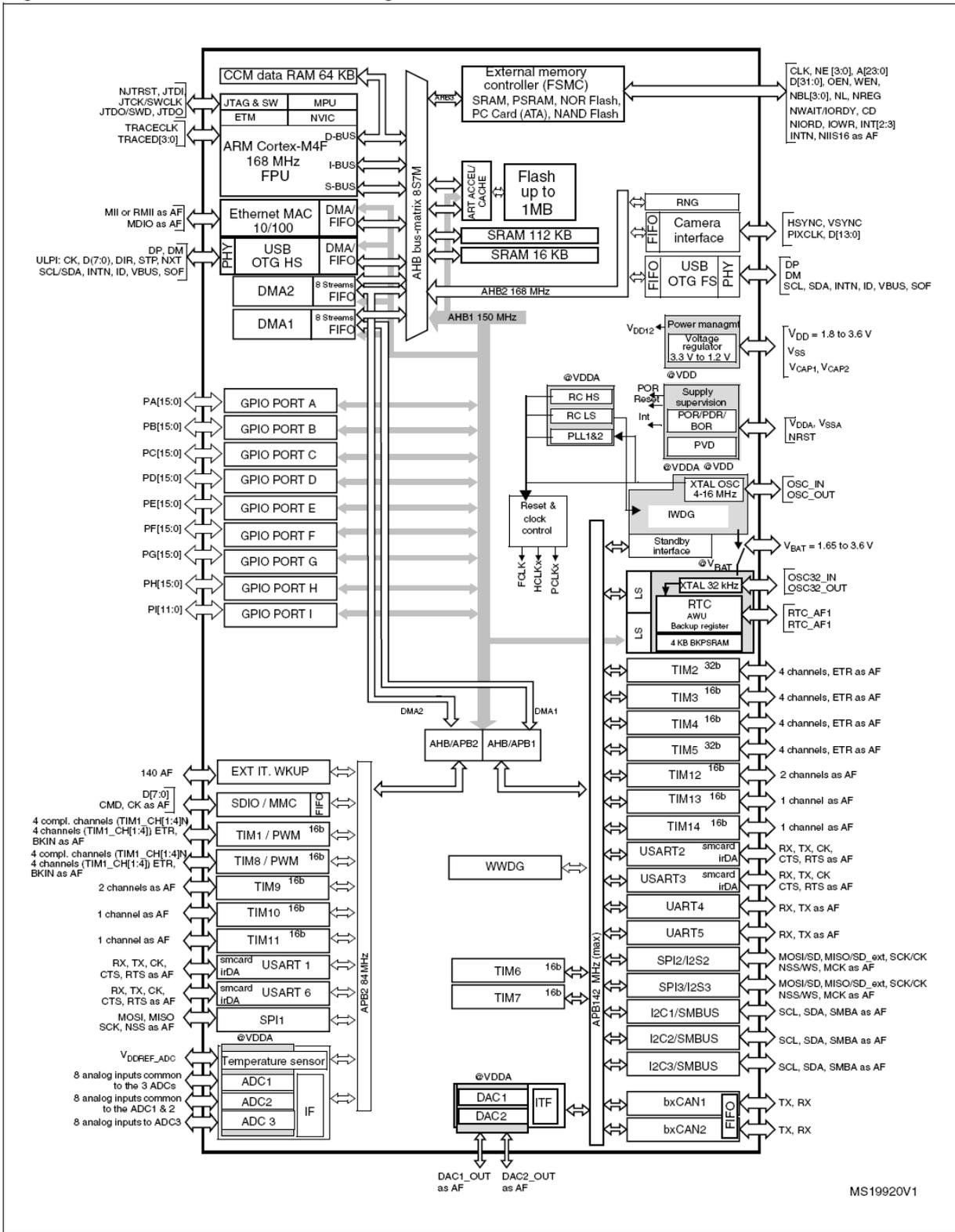


Figura 6. Diagrama de bloques del STM32F407VGT6



## 4 Requisitos y conexión del kit

Para ejecutar y desarrollar aplicaciones en tu placa *STM32F4DISCOVERY*, los requisitos mínimos son:

- PC con Windows (2000, XP, Vista, 7).
- cable USB tipo A a Mini-B, utilizado para alimentar la placa (a través del conector USB CN1) desde el PC y conectarse al *ST-LINK/V2* para depuración y programación.

La placa se debe conectar al PC de la siguiente manera. Recuerda que es necesario instalar los controladores para poder utilizarla.

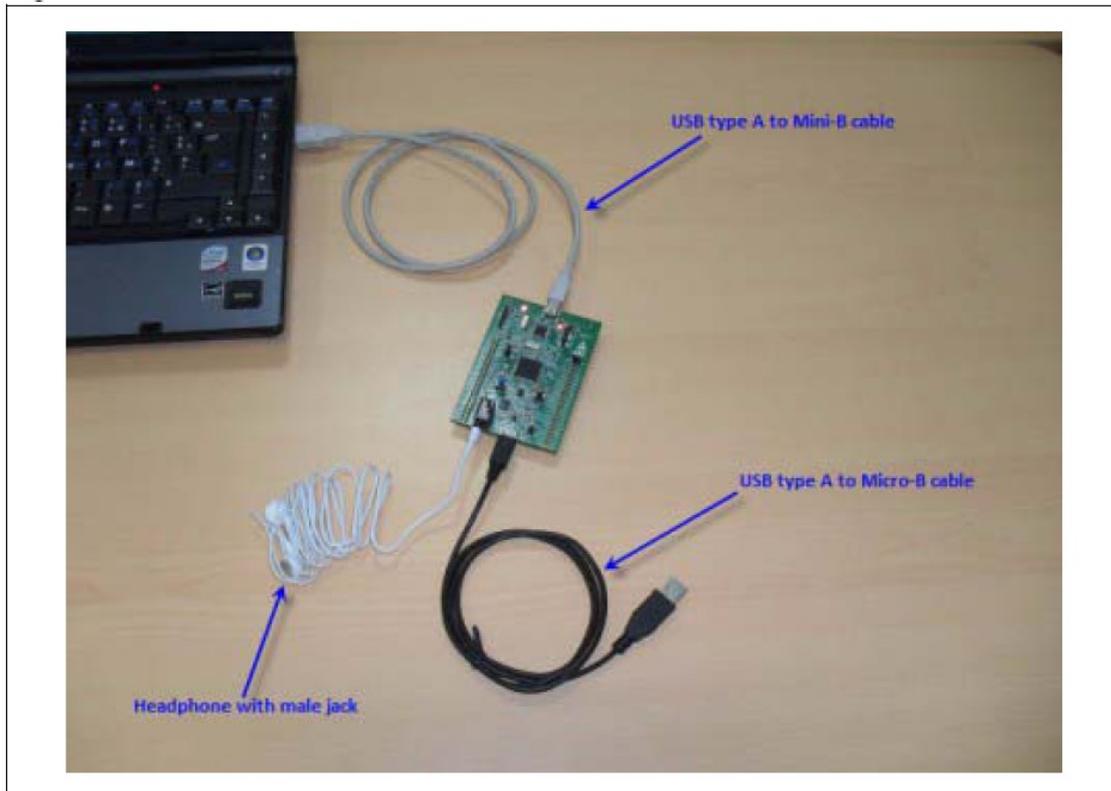


Figura x. Entorno hardware

## 5 El depurador ST-LINK/V2

Una parte de la placa *STM32F4DISCOVERY* es un depurador *ST-LINK/V2* que es soportada por diversos entornos de desarrollo.

Antes de poder emplear la placa con su depurador es necesario instalar los controladores. Comprobar que los jumpers estén de la siguiente manera.



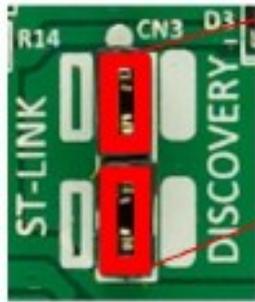


Figura x. *Jumpers* CN3

## 5.1 *Uso en Keil MDK-ARM*

### 5.1.1 **Instalación**

Keil viene con el controlador ST-LINK/V2. Para instalarlo debemos localizar la instalación de Keil que, por defecto, es en el directorio `C:\Keil` del disco duro local y acudir al directorio `ARM\STLink\USBDriver`. Ejecutar el driver y aceptar la instalación aunque salgan mensajes sospechosos.

A continuación, conectar la placa y comprobar que el asistente de controladores Windows encuentra y, aceptando la búsqueda automática, instala los controladores.

### 5.1.2 **Configurando las opciones de depuración**

Cada proyecto Keil se configura individualmente, así que suele ser necesario reconfigurarlo manualmente cada vez que se mueva de ordenador.

Para ello, seguir estos pasos:

1. En el menú **Project**, selecciona **Options for Target 1** para mostrar el cuadro de diálogo **Target Options**.
2. Abre la pestaña **Debug**, haz clic en **Use** y selecciona **ST-Link Debugger**. A continuación, haz clic en **Settings** y selecciona el protocolo SWD. Clic en **OK** para guardar la configuración.
3. Marca la opción **“Run to main()”**.



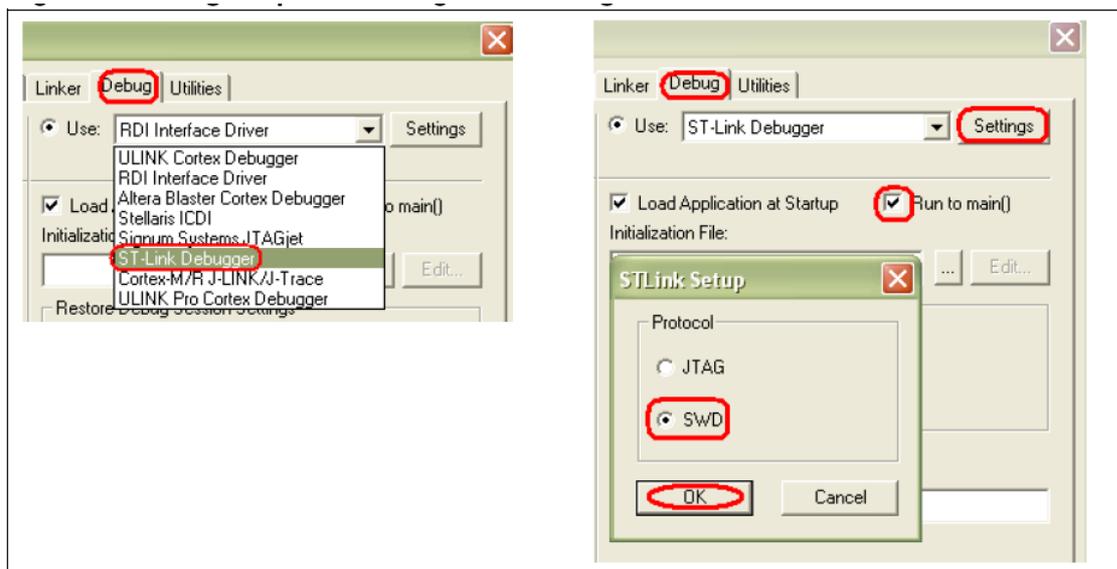


Figura x. Cuadro de diálogo *Target Options* – Pestaña *Debug*

4. Abre la pestaña **Utilities**, selecciona **Use Target Driver for Flash Programming** y elige **ST-Link Debugger** en el desplegable.
5. Verifica que la opción **Update Target before Debugging option** está seleccionada.
6. Haga clic en **OK** para guardar la configuración.

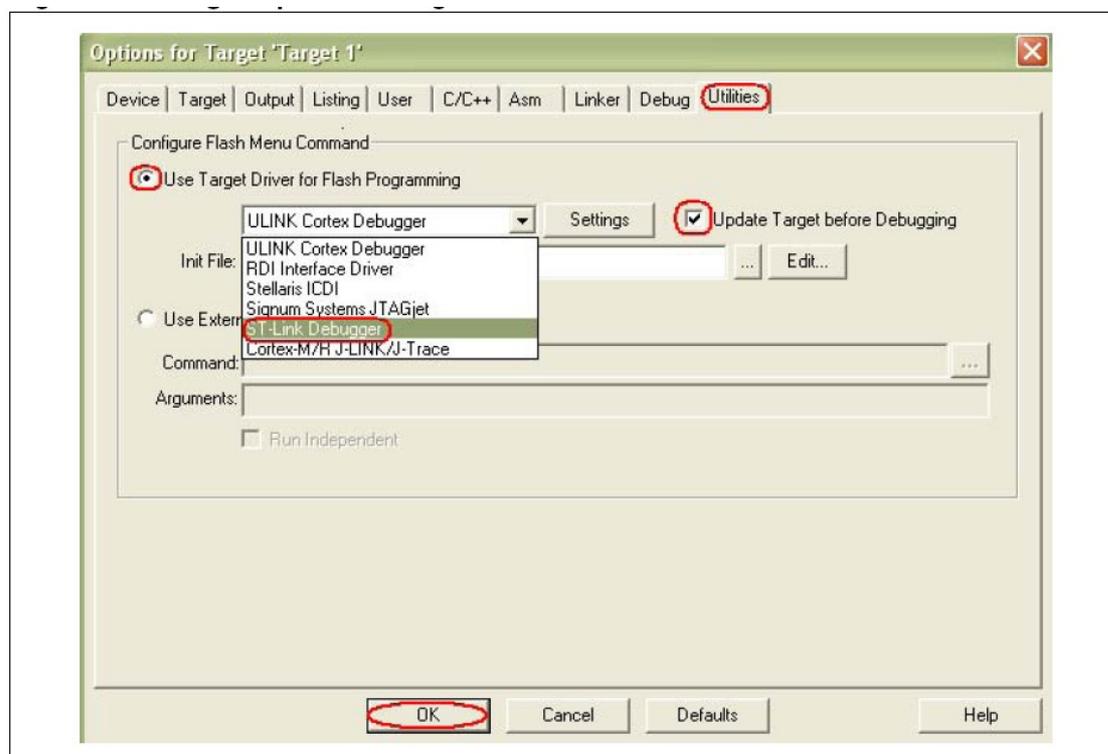


Figura x. Cuadro de diálogo *Target Options* – Pestaña *Utilities*



### 5.1.3 Depuración más avanzada

Esto está aún verde, pero es importante y sustituirá a lo anterior un día de estos.

La guía provisional (en inglés) está en <http://armcortexm.blogs.upv.es/stm32f4-discovery-and-printf-redirect-to-debug-viewer-in-keil/>

## 6 Uso del entorno de desarrollo MDK-ARM (Keil™)

### 6.1 Versión demo de Keil

En [www.keil.com](http://www.keil.com) se puede descargar una versión limitada de MDK-ARM previo registro. Una limitaciones se basa en permitir un tamaño de código máximo de 32K, lo cual es suficiente probar todos los ejemplos.

### 6.2 Construir un proyecto ya existente

1. Abrir el *MDK-ARM μVision4 IDE*,

La figura 3 muestra los nombres de las ventanas referenciadas en este apartado.

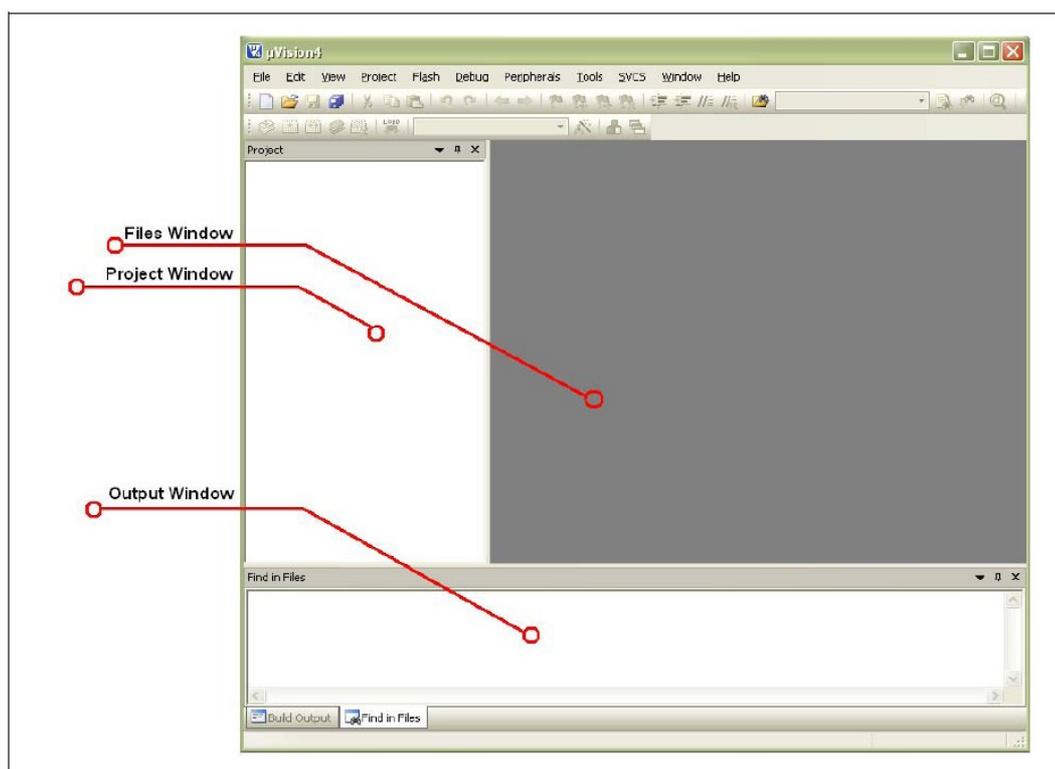


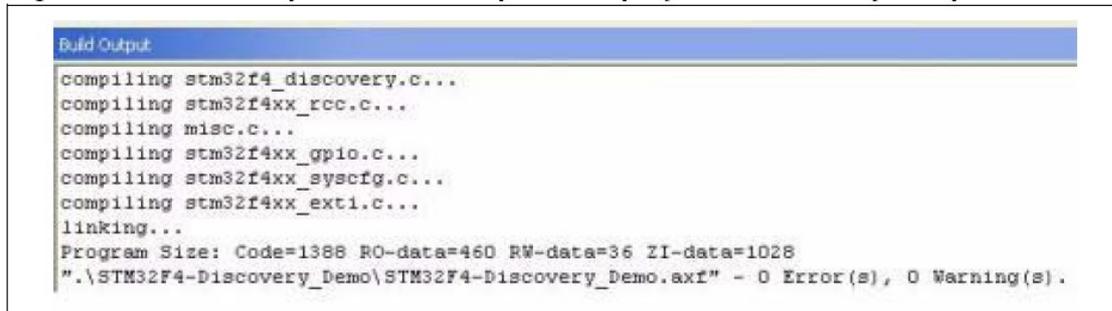
Figura 3. Entorno MDK-ARM μVision4 IDE

2. En el menú **Project**, seleccione abrir proyecto (**Open Project ...**) para mostrar el cuadro de diálogo de selección de fichero de proyecto. Seleccionar el archivo de proyecto *STM32F4-Discovery.uvproj* y haga clic en abrir (**Open**).

3. En el menú **Project**, selecciona **Rebuild all target files** para compilar el proyecto.



4. Si tu proyecto se ha compilado con éxito, aparecerá en pantalla la siguiente ventana **Build Output** (figura 4):



```
Build Output
compiling stm32f4_discovery.c...
compiling stm32f4xx_rcc.c...
compiling misc.c...
compiling stm32f4xx_gpio.c...
compiling stm32f4xx_syscfg.c...
compiling stm32f4xx_exti.c...
linking...
Program Size: Code=1388 RO-data=460 RW-data=36 ZI-data=1028
".\STM32F4-Discovery_Demo\STM32F4-Discovery_Demo.axf" - 0 Error(s), 0 Warning(s).
```

Figura 4. Build Output - MDK-ARM  $\mu$ Vision4. Proyecto compilado con éxito.

### 6.3 Ejecutar y depurar un proyecto MDK-ARM

(Aço hi haurà que ampliar-ho a classe)

En el entorno MDK-ARM  $\mu$ Vision4 IDE, haga clic en la lupa para programar la memoria Flash e iniciar la depuración, como se muestra a continuación en la figura 5.



Figura 5. Inicio de una sesión de depuración en MDK-ARM  $\mu$ Vision4

El depurador en el MDK-ARM IDE se puede utilizar para depurar el código fuente en C, establecer puntos de ruptura (*breakpoints*), monitorizar variables individuales y observar eventos durante la ejecución del código, como se muestra en la figura 6.



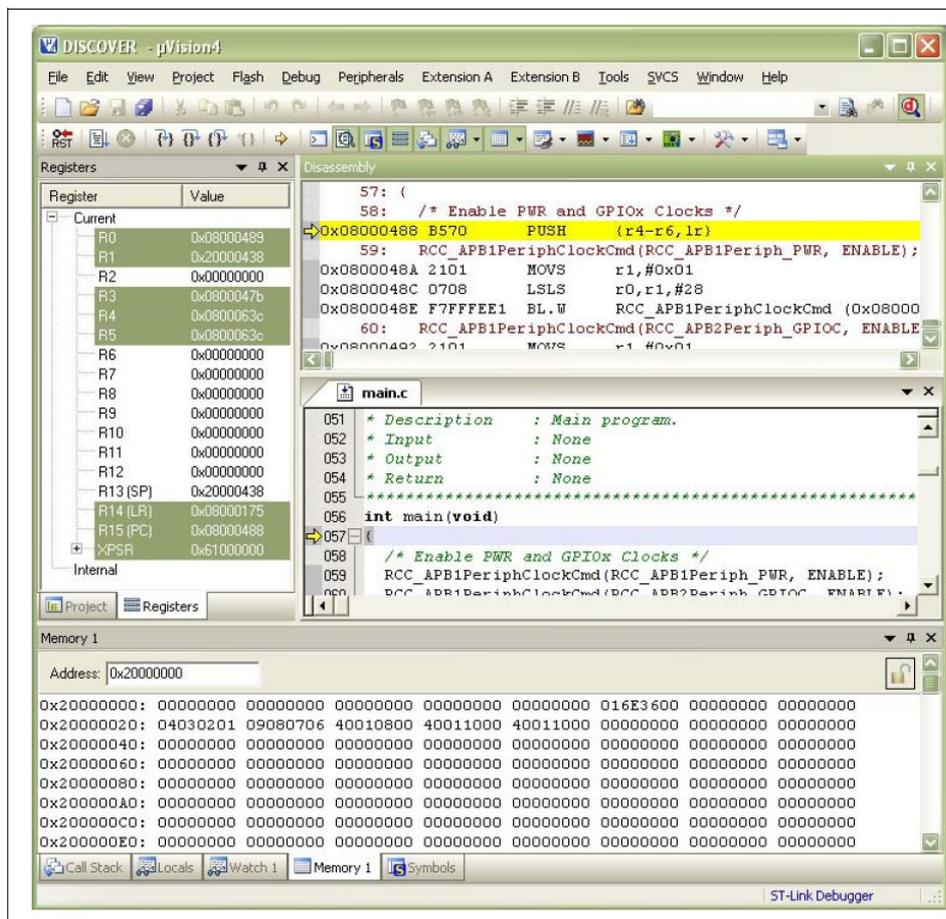


Figura 6. Espacio de trabajo del IDE MDK-ARM

## 6.4 Crear un proyecto con el entorno MDK-ARM

### 6.4.1 Crear el proyecto

1. En el menú **Project**, selecciona **New  $\mu$ Vision Project...** para visualizar el cuadro de diálogo de creación de proyecto **Create Project File**. Nombra el nuevo proyecto y haz clic en guardar (**Save**).

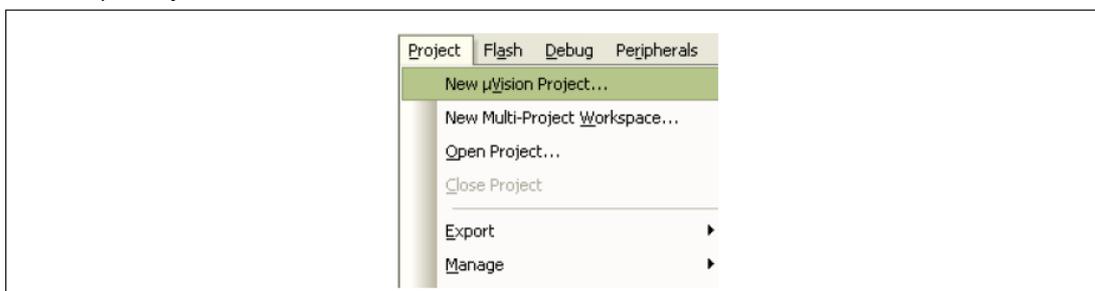


Figura 7. Creación de un nuevo proyecto

2. Cuando se guarda un nuevo proyecto, el IDE muestra el cuadro de diálogo de selección de dispositivos, donde seleccionaremos nuestro dispositivo. En este caso vamos a utilizar el **STMicroelectronics** montado en la placa **STM32F4DISCOVERY**, con lo que habrá que hacer



doble clic en **STMicroelectronics**, seleccionar el dispositivo **STM32F407VGT6** y hacer clic en **OK** para guardar la configuración.

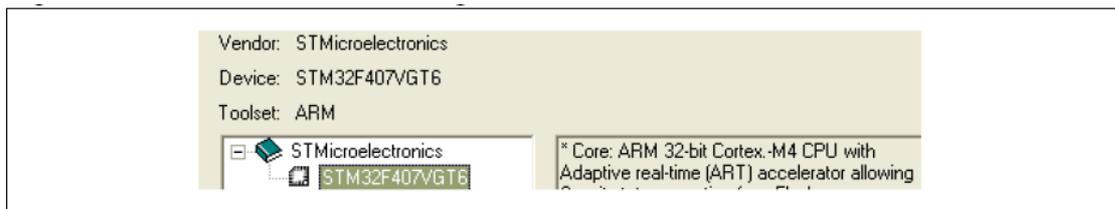


Figura 8. Cuadro de diálogo selección de dispositivo

3. Haz clic en el botón “**Yes**” para copiar el **STM32 Startup Code** a la carpeta del proyecto y agregar el archivo al proyecto como se muestra en la figura 9.



Figura 9. Cuadro de diálogo **STM32 Startup Code**

Nota: El fichero por defecto del **STM32 Startup Code** incluye la función *SystemInit*. Se puede comentar este fichero para que no se utilice o añadir el archivo *system\_stm32f4xx.c* desde la biblioteca *STM32f4xx firmware library*.

## 6.4.2 Configurar el proyecto

1. En el menú **Project**, selecciona **Options for Target 1** para mostrar el cuadro de diálogo **Target Options**.

2. Abre el **Target tab** e introduce la configuración de inicio (*start*) y tamaño (*size*) de **IROM1** e **IARM1** como se muestra en la figura 13.



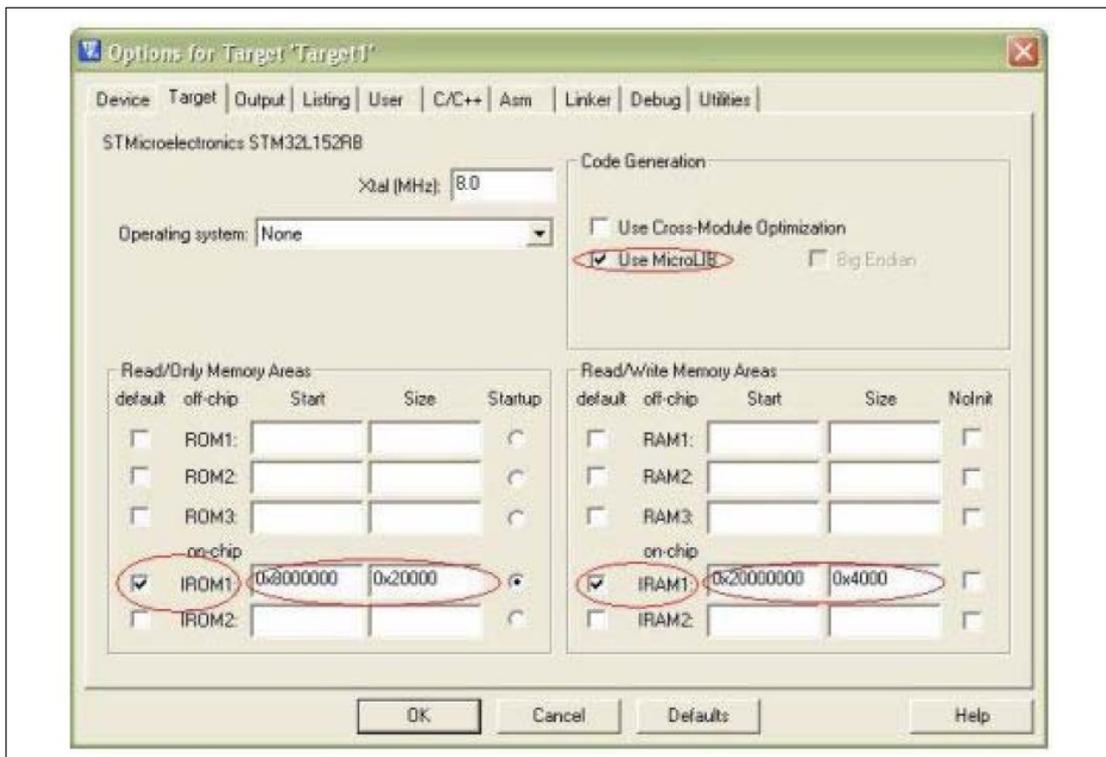


Figura 13. Cuadro de diálogo *Target Options* - Pestaña *Target*

3. Siguiendo las instrucciones del apartado *ST-Link/V*”, configura las opciones de depuración.

### 6.4.3 Añadiendo archivos fuente

Para crear un nuevo fichero fuente, selecciona **New** en el menú **File**, y se abrirá una ventana de edición vacía donde introducir el código fuente.

El entorno *MDK-ARM* resalta en color la sintaxis de C cuando se guarda el archivo con (**File**> **Save As...**) extensión **\*.c**. En el ejemplo de la figura 10, el archivo se guarda como **main.c**.

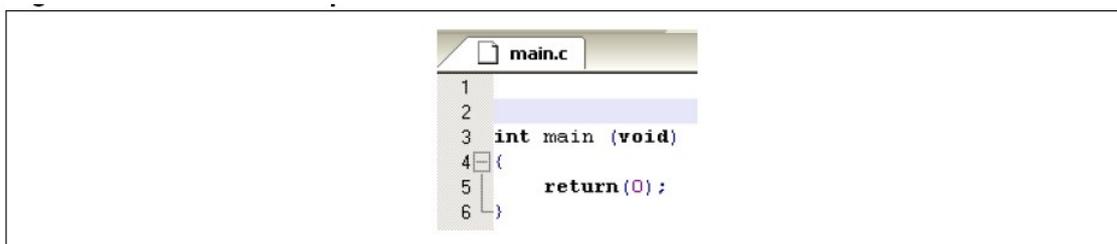


Figura 10. Ejemplo de archivo *main.c*

El entorno *MDK-ARM* ofrece varias maneras de agregar ficheros fuente a un proyecto. Por ejemplo, se puede seleccionar el grupo de archivos en **Project Window** > **Files** y haciendo clic con el botón derecho del ratón para abrir el menú contextual. Selecciona la opción **Add Files...**, y selecciona finalmente el archivo **main.c** creado previamente.



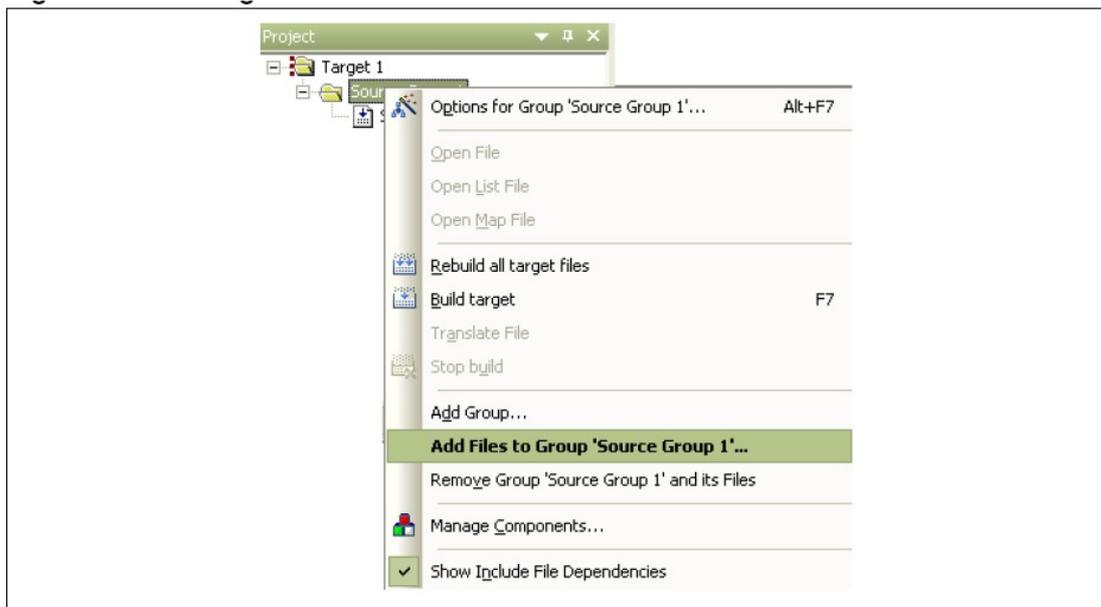


Figura 11. Añadir ficheros fuente

Si el archivo se agrega correctamente, se muestra la siguiente ventana (figura 12).



Figura 12. Estructura nuevo proyecto

8. En el menú **Project**, selecciona **Build Target**.

9. Si el proyecto se compila con éxito, se abrirá la siguiente ventana (figura 16):

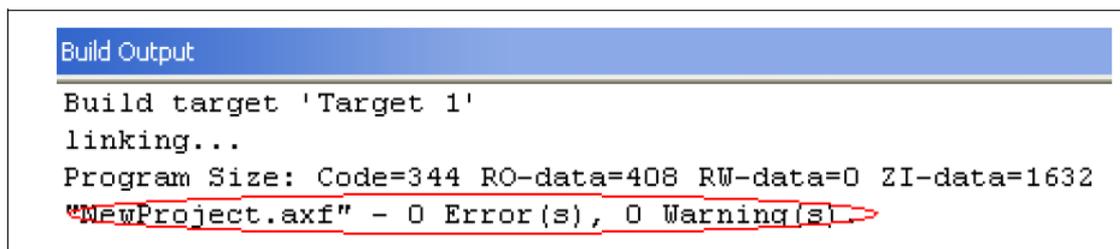


Figura 16. Proyecto compilado con éxito en MDK-ARM μVision4

10. Antes de ejecutar la aplicación, conecta la placa *STM32F4DISCOVERY* como se describe en la sección 1.

11. Para programar la memoria Flash y empezar a depurar, sigue las instrucciones que figuran en la sección 4.2: “depurar y ejecutar tu proyecto”.



## 7 EI “STM32F4DISCOVERY board firmware package

St proporciona un paquete de software para el kit que debe descargarse de su web. Dicho paquete incorpora las bibliotecas CMSIS, las bibliotecas específicas de periféricos y un conjunto de ejemplos que permiten hacerse una idea de las capacidades del microcontrolador. El paquete viene en comprimido y, al descomprimir, el archivo zip se crea una carpeta, *STM32F4-Discovery\_FW\_VX.YZ*, que contiene las siguientes subcarpetas:

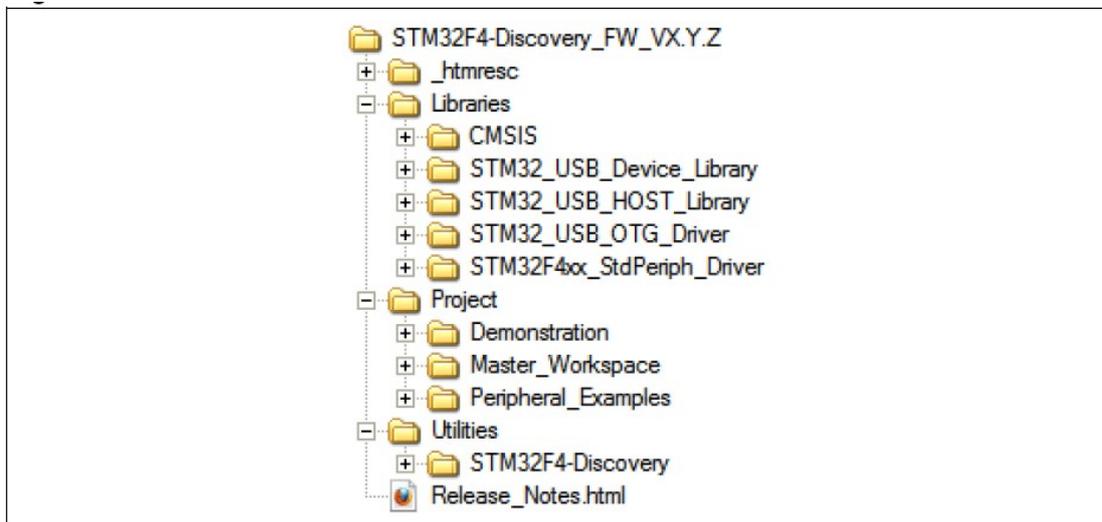


Figura x. Estructura directorios

### 7.1 Carpeta libraries

Esta carpeta contiene la capa de abstracción del hardware (HAL) para los dispositivos *STM32F4xx*.

#### 7.1.1 Subcarpeta CMSIS

Esta subcarpeta contiene los archivos CMSIS *STM32F4xx* y *Cortex-M4F*.

**Los archivos *CMSIS Cortex-M4F* consisten en:**

- *Core Peripheral Access Layer*: que contiene las definiciones de nombres, direcciones y funciones de ayuda para acceder a los registros y periféricos Cortex-M4F. Además, define una interfaz independiente del dispositivo para *kernels RTOS* que incluye definiciones de canales de depuración.

- *CMSIS DSP Software Library*: cuenta con una suite de funciones de procesamiento de señal para su uso en dispositivos basados en el procesador *Cortex-M*.

La biblioteca está completamente escrita en C y es totalmente compatible con *CMSIS*.

**Los archivos *STM32F4xx CMSIS* consisten en:**

- *stm32f4xx.h*: este archivo contiene las definiciones de todos los registros de periféricos, bits, y mapeo de memoria para dispositivos *STM32F4xx*. Este archivo es el único *include* utilizado en el código fuente de la aplicación, por ejemplo, *main.c*.



- *system\_stm32f4xx.c/h*: Este archivo contiene la configuración del reloj para los dispositivos *STM32F4xx*. Exporta la función *SystemInit()* que configura el reloj, los factores multiplicadores y divisores PLL, AHB / APBx *prescalers* y *Flash*. Esta función es invocada en la inicialización justo después del reset y antes de comenzar el programa principal. La llamada se realiza dentro del archivo *startup\_stm32f4xx.s*.

- *startup\_stm32f4xx.s*: Proporciona el código de inicialización del *Cortex-M4F* y los vectores de interrupción para todos los manejadores de interrupción del dispositivo *STM32F4xx*.

## 7.1.2 STM32F4xx\_StdPeriph\_Driver

Esta subcarpeta contiene los fuentes de los controladores de periféricos del *STM32F4xx* (excepto USB y Ethernet).

Cada *driver* consiste en un conjunto de rutinas y estructuras de datos que cubren (casi) todas las funcionalidades de los periféricos. El desarrollo de cada *driver* está guiado por una API común, que estandariza la estructura del *driver*, las funciones y los nombres de los parámetros.

Cada periférico tiene un archivo de código fuente, *stm32f4xx\_ppp.c*, y un archivo de cabecera, *stm32f4xx\_ppp.h*. El archivo *stm32f4xx\_ppp.c* contiene todas las funciones firmware necesarias para utilizar el periférico "PPP".

## 7.2 Carpeta de proyecto

Esta carpeta contiene los fuentes de las aplicaciones *STM32F4DISCOVERY*.

### 7.2.1 Subcarpeta demonstration

Esta subcarpeta contiene los fuentes de una aplicación demo con el proyecto preconfigurado para los entornos IAR *EWARM*, Keil *MDK-ARM*, Atollic *TrueSTUDIO* y *TASKING*.

En la subcarpeta *Binary* se proporcionan los binarios (\*.hex y \*.dfu) de esta aplicación. Pudiéndose utilizar el *STM32F4xx Bootloader* o cualquier herramienta de programación para reprogramar la aplicación demo usando estos ficheros.

### 7.2.2 Subcarpeta Peripheral\_Examples

Esta subcarpeta contiene ejemplos para algunos periféricos con proyectos preconfigurados para los entornos *EWARM*, *MDK-ARM*, *TrueSTUDIO* y *TASKING*.

## 7.3 Carpeta Utilities

Esta carpeta contiene una capa de abstracción específica para el *STM32F4DISCOVERY*. La idea de esta capa es proporcionar servicio para manejar los dispositivos disponibles. Proporciona los siguientes controladores:

- *stm32f4\_discovery.c*: proporciona funciones para manejar el pulsador de usuario y 4 LEDs (*LD3.LD6*)



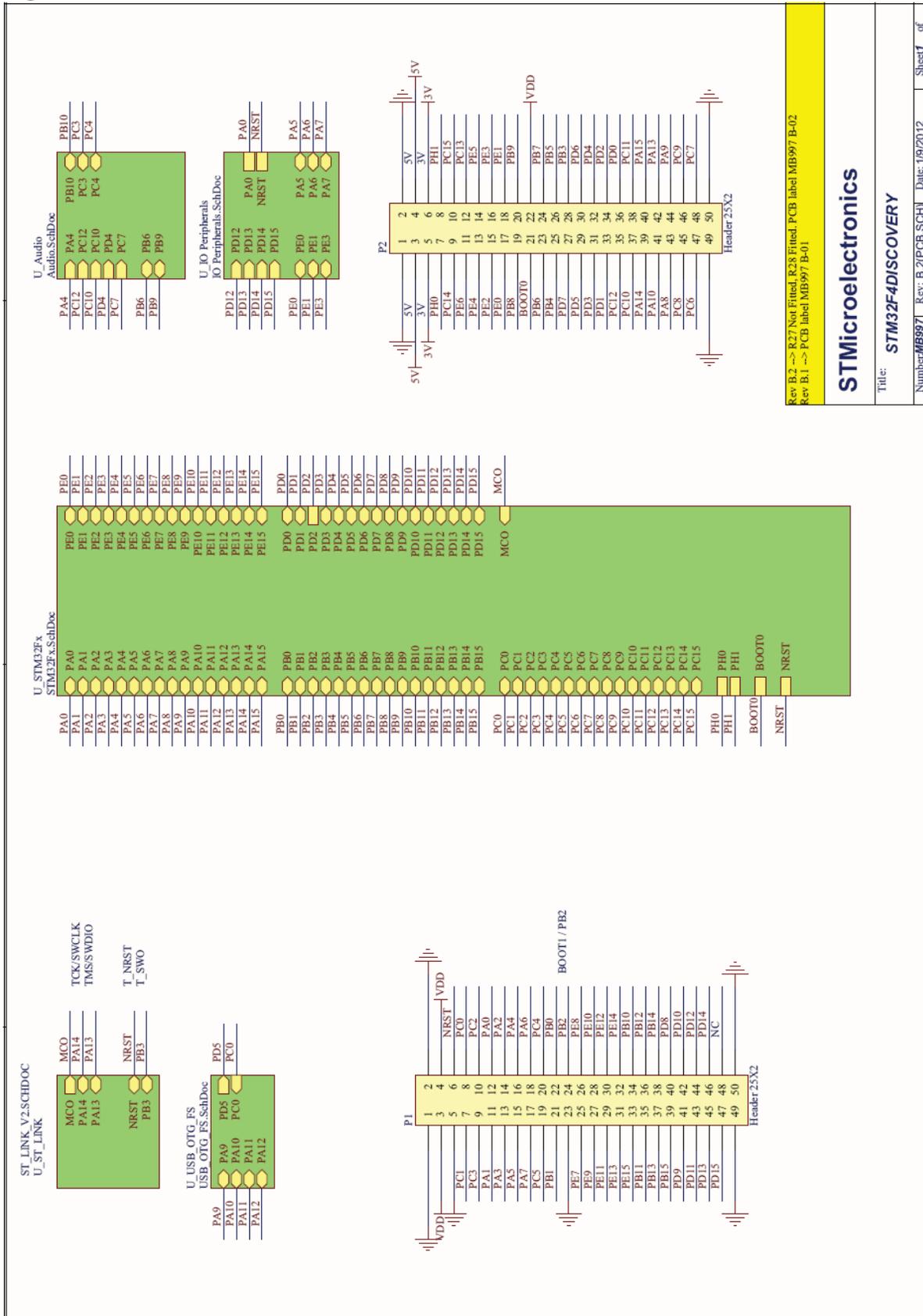
- *stm32f4\_discovery\_audio\_codec.c/.h*: proporciona funciones para manejar el audio DAC (CS43L22)

- *stm32f4\_discovery\_lis302dl.c/.h*: proporciona funciones para manejar el acelerómetro MEMS (LIS302DL).



# 8 Esquema eléctrico

Figura x: STM32F4DISCOVERY



Rev B.2 -> B.27 Not Fitted, R.28 Fitted, PCB label MB997 B-02  
 Rev B.1 -> PCB label MB997 B-01

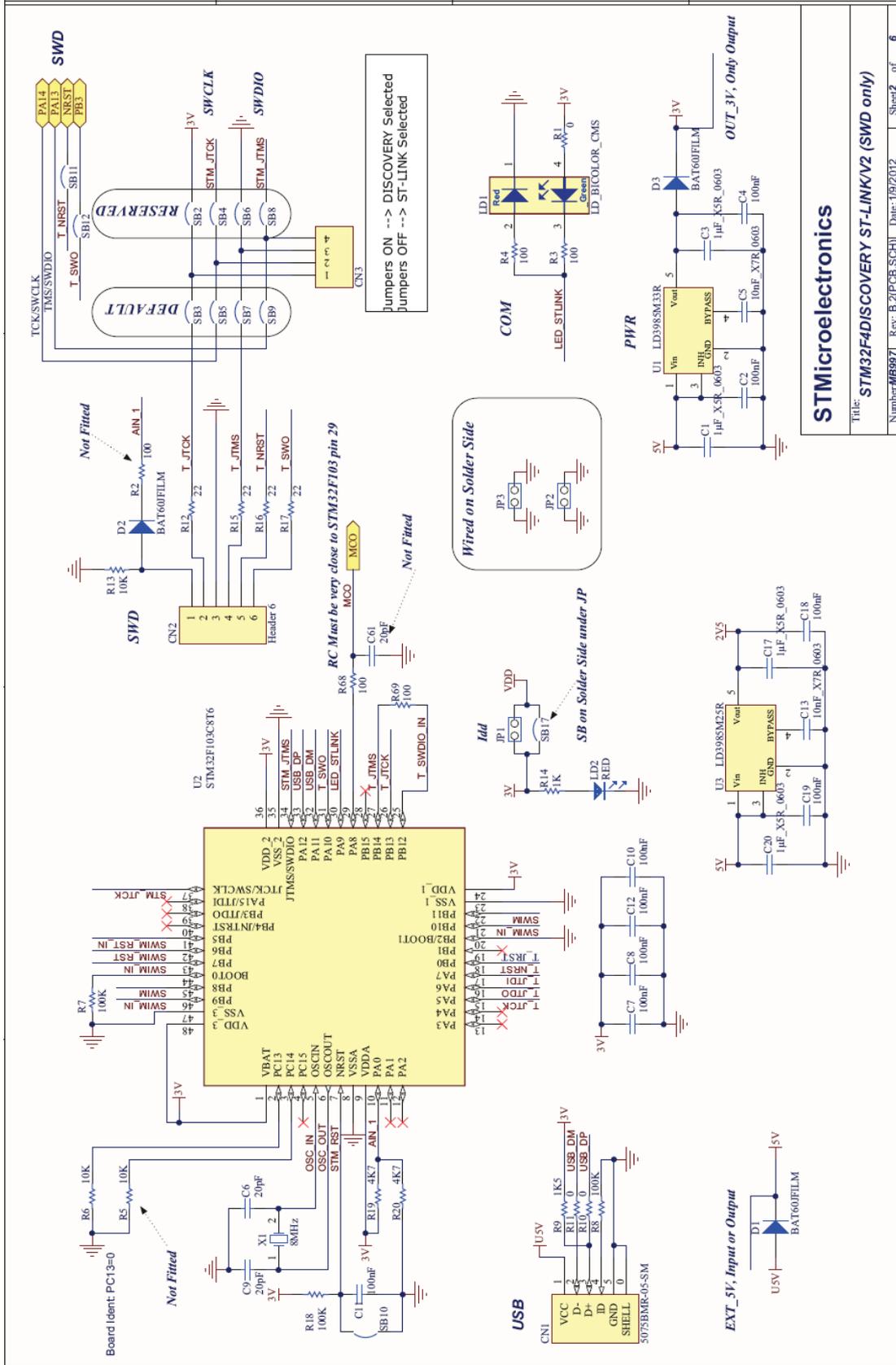
**STMicroelectronics**

Title: **STM32F4DISCOVERY**

Number: **MB997** Rev: **B.2[PCB.SCH]** Date: **1/9/2012** Sheet **7** of



Figura 12: ST-LINK/V2 (solo SWD)



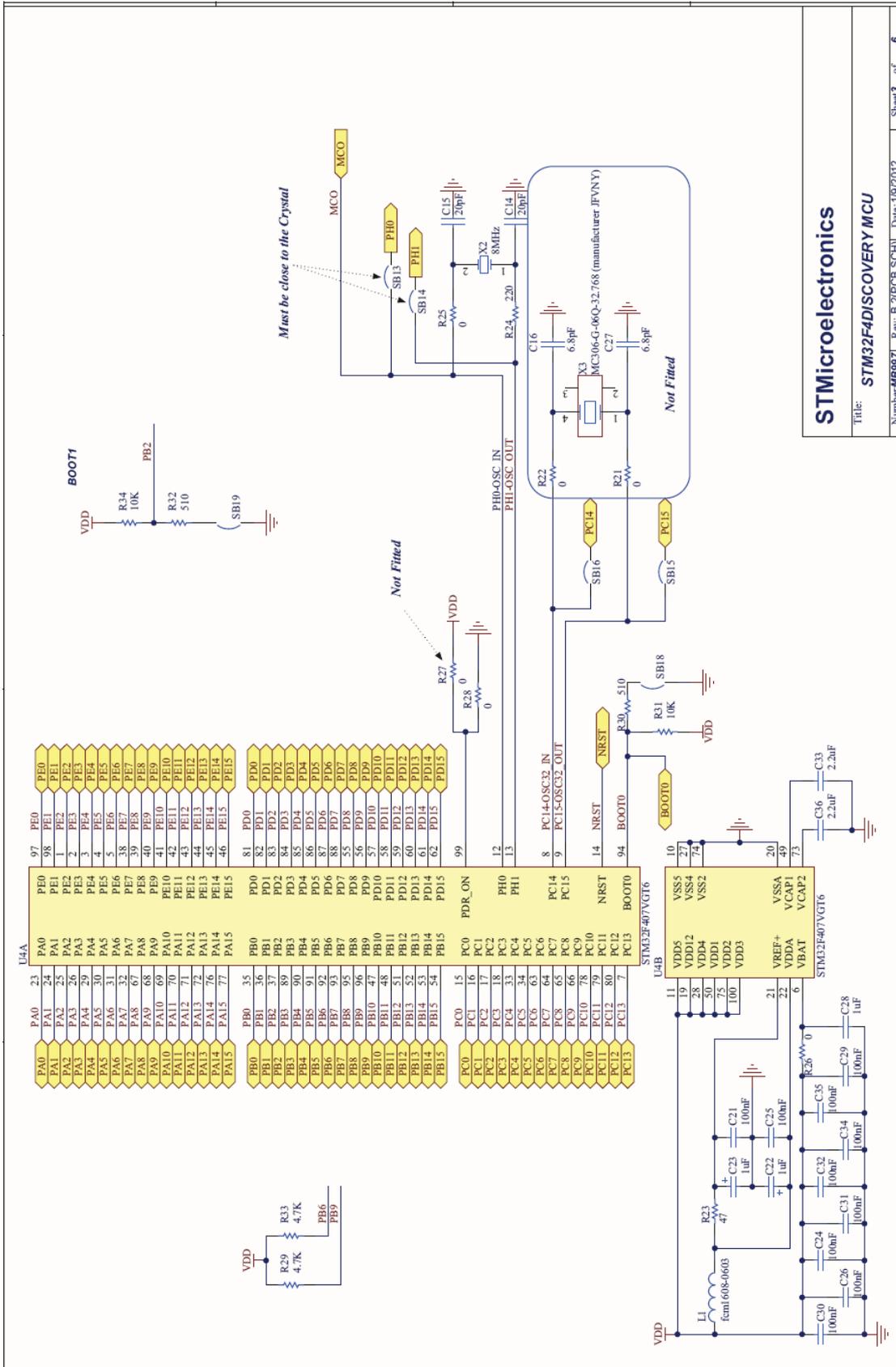
STMicroelectronics

Title: STM32F4DISCOVERY ST-LINK/V2 (SWD only)

Number: M9997 Rev: B.2[PCB SCH] Date: 19/2012 Sheet 2 of 6



Figura 13. MCU

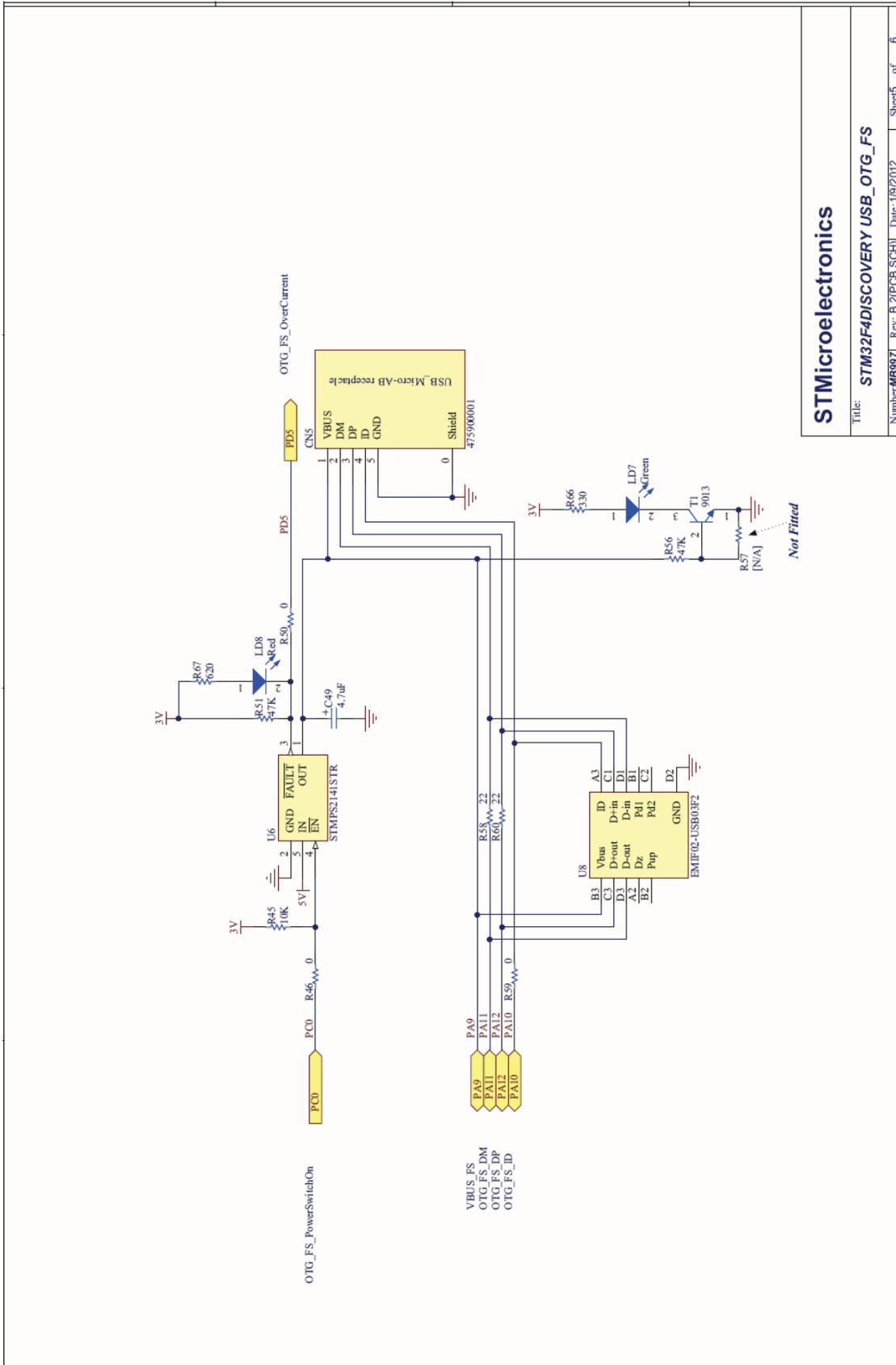


**STMicroelectronics**  
 Title: STM32F4DISCOVERY MCU  
 Number: MB997 | Rev: B.2(PCB.SCH) | Date: 19/2012 | Sheet 3 of 6





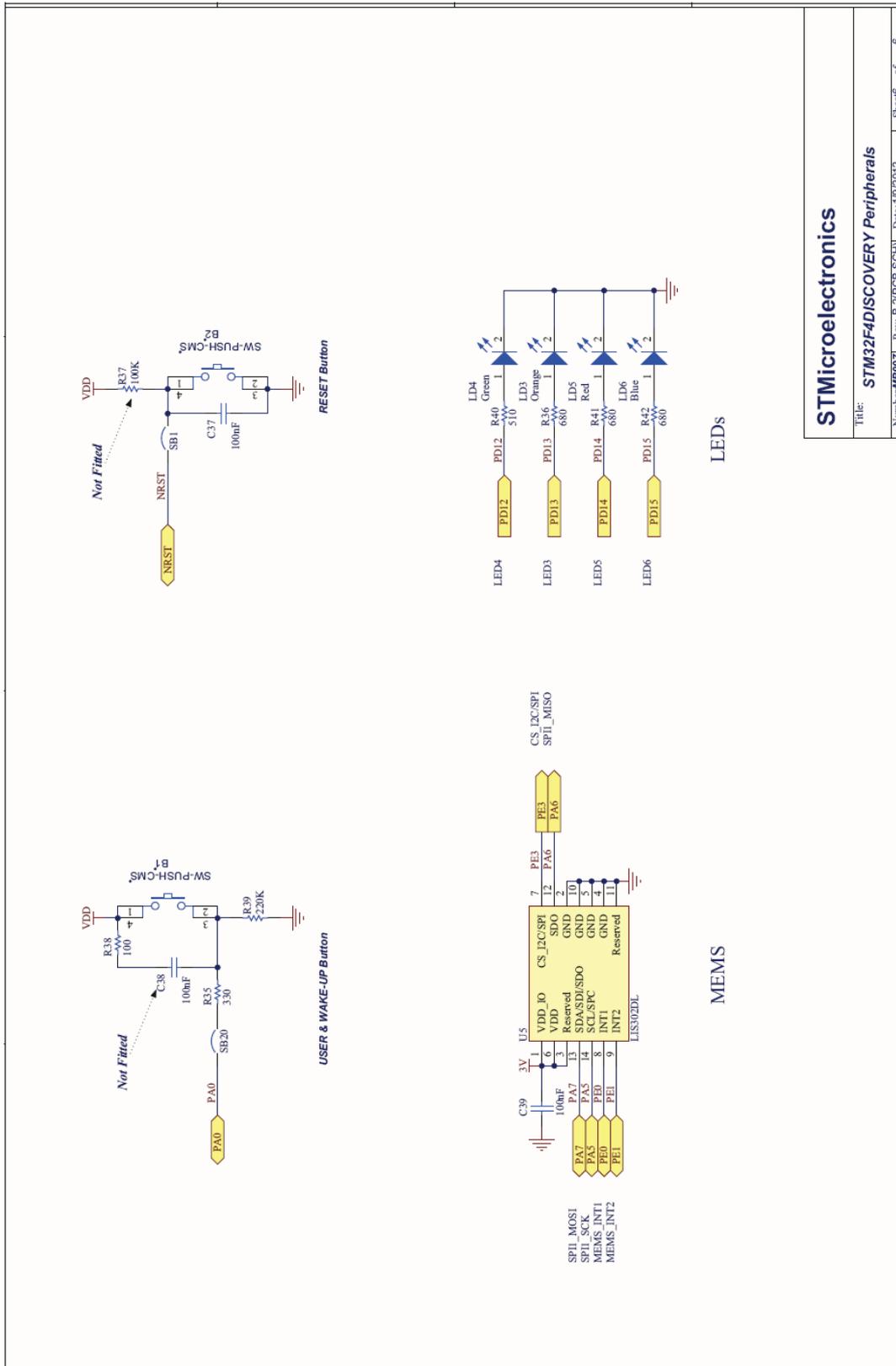
Figura 15: *USB\_OTG\_FS*



**STMicroelectronics**  
 Title: **STM32F4DISCOVERY USB\_OTG\_FS**  
 Number: **MB997** Rev: **B.2[PCB\_SCH]** Date: **19/2/2012** Sheet **5** of **6**



Figura 16: Periféricos



STMicroelectronics

Title: STM32F4DISCOVERY Peripherals  
 Number: MB9971 Rev.: B.2(PCB SCH) Date: 19/2/12 Sheet6 of 6

