AN4080
Application note

Getting started with STM32F0x1/x2/x8 hardware development

Introduction

This application note is intended for system designers who require a hardware implementation overview of the development board features such as the power supply, the clock management, the reset control, the boot mode settings and the debug management. It shows how to use the STM32F0xxxx product family and describes the minimum hardware resources required to develop your application.

Within the STM32F0xxxx product family, the STM32F0x8 line offers a different embedded power management to support 1.8 V operation.

This document also includes detailed reference design schematics with the description of the main components, interfaces and modes.

Table 1. Applicable products

Type	Part number
Microcontrollers	STM32F031C4, STM32F031C6, STM32F031F4, STM32F031F6, STM32F031G4, STM32F031G6, STM32F031K4, STM32F031K6, STM32F038C6, STM32F038F6, STM32F038G6, STM32F038K6, STM32F042C4, STM32F042C6, STM32F042F4, STM32F042F6, STM32F042G4, STM32F042G6, STM32F042K4, STM32F042K6, STM32F042T4, STM32F042T6, STM32F048C6, STM32F048G6, STM32F048T6 STM32F051C4, STM32F051C6, STM32F051C8, STM32F051K4, STM32F051K6, STM32F051K8, STM32F051R4, STM32F051R6, STM32F051R8, STM32F058C8, STM32F058R8, STM32F071V8, STM32F071CB, STM32F071RB, STM32F071VB, STM32F072C8, STM32F072R8, STM32F072V8, STM32F072CB,STM32F072RB, STM32F072VB, STM32F078CB, STM32F078RB, STM32F078VB, STM32F091CB, STM32F091RB, STM32F091VB, STM32F091CC, STM32F091RC, STM32F091VC, STM32F098CC, STM32F098RC, STM32F098VC

Note: In this document, the notation used for devices is STM32F0xx where xx are the two first digits following F0 standing for entry-level family. Pin count and memory size d not impact this hardware description.

Contents

List of tables

List of figures

1 Power supplies and reset sources of the STM32F0xx family

1.1 Power supply schemes

The STM32F0xx family features different products with various supply schemes. It includes an internal regulator in order to have an internal 1.8 V supply for the core and digital logic. On STM32F0x8 line, the voltage regulator is bypassed, therefore the 1.8 V power supply must be provided to the chip by an external regulator.

There is a variety of power supply schemes:

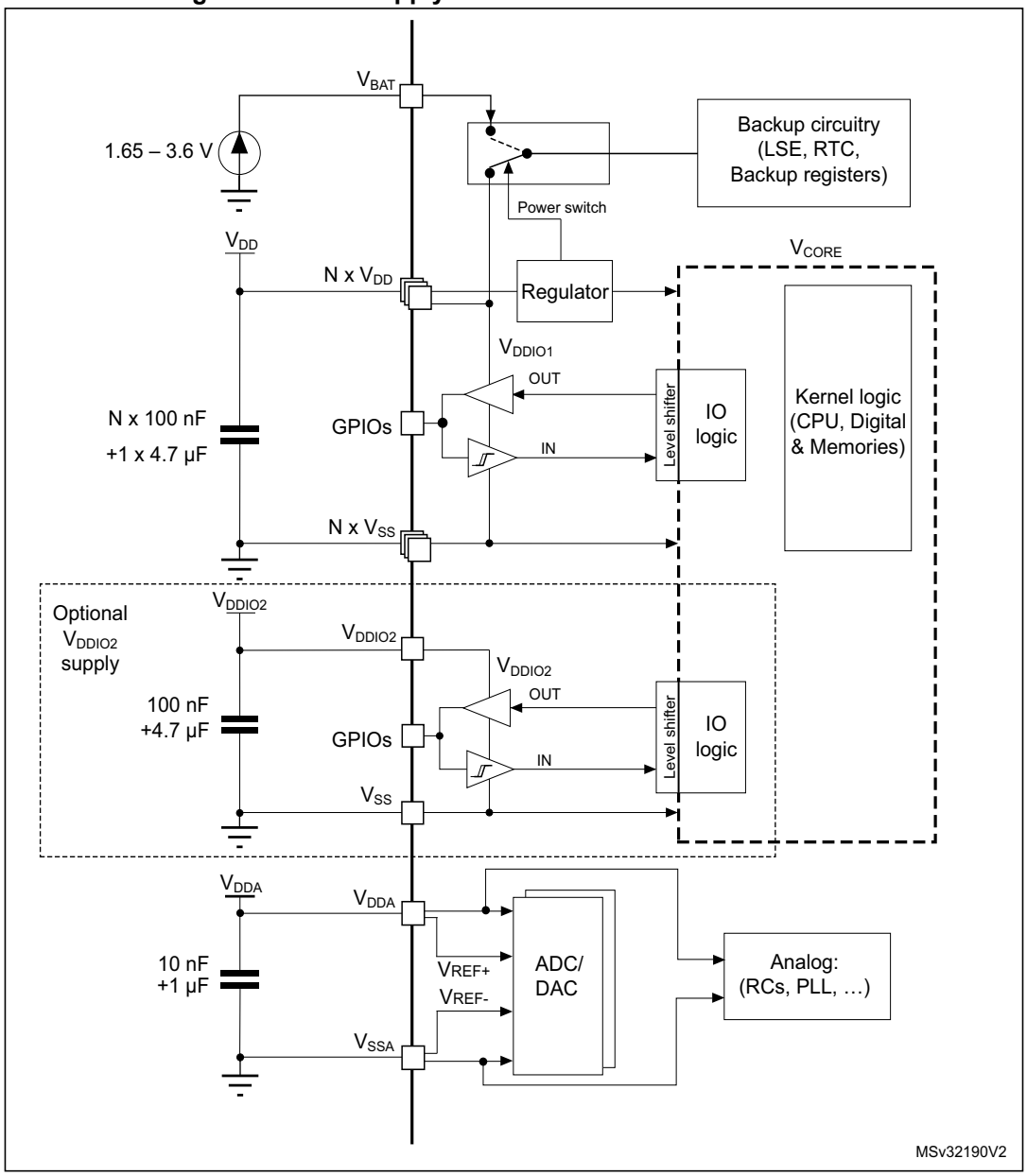
- V_{DD}: external power supply for I/Os and the 1.8 V domain. Provided externally through V_{DD} pins.
- V_{DDA}: external analog power supply for ADC/DAC, Reset blocks, HSI, HSI14, HSI48, LSI and PLL (minimum voltage to be applied to V_{DDA} is 2.4 V when the ADC or DAC are used).
  The V_{DDA} voltage level must always be greater than or equal to the V_{DD} voltage level and must be provided first.
- V_{DDIO2}: external power supply for some dedicated IOs that is provided externally through the VDDIO2 pin. The V_{DDIO2} voltage level is fully independent from V_{DD} or V_{DDA}, but it must not be provided without a valid supply on V_{DD}. Refer to the dedicated device datasheet to find the concerned list of I/Os.
- V_{BAT}: power supply for RTC, LSE 32 kHz oscillator and backup registers (through power switch) when V_{DD} is not present.

Table 2. Power supplies of the STM32F0xx family

	Product family		
Power supply	**STM32F0x1**	**STM32F0x2**	**STM32F0x8**
V_{DD}	2.0 to 3.6 V		1.8 V +/- 8 %
V_{DDA}	V_{DD} to 3.6 V		
V_{DDIO2}(1)	1.65 to 3.6 V		
V_{BAT}	1.65 to 3.6 V		

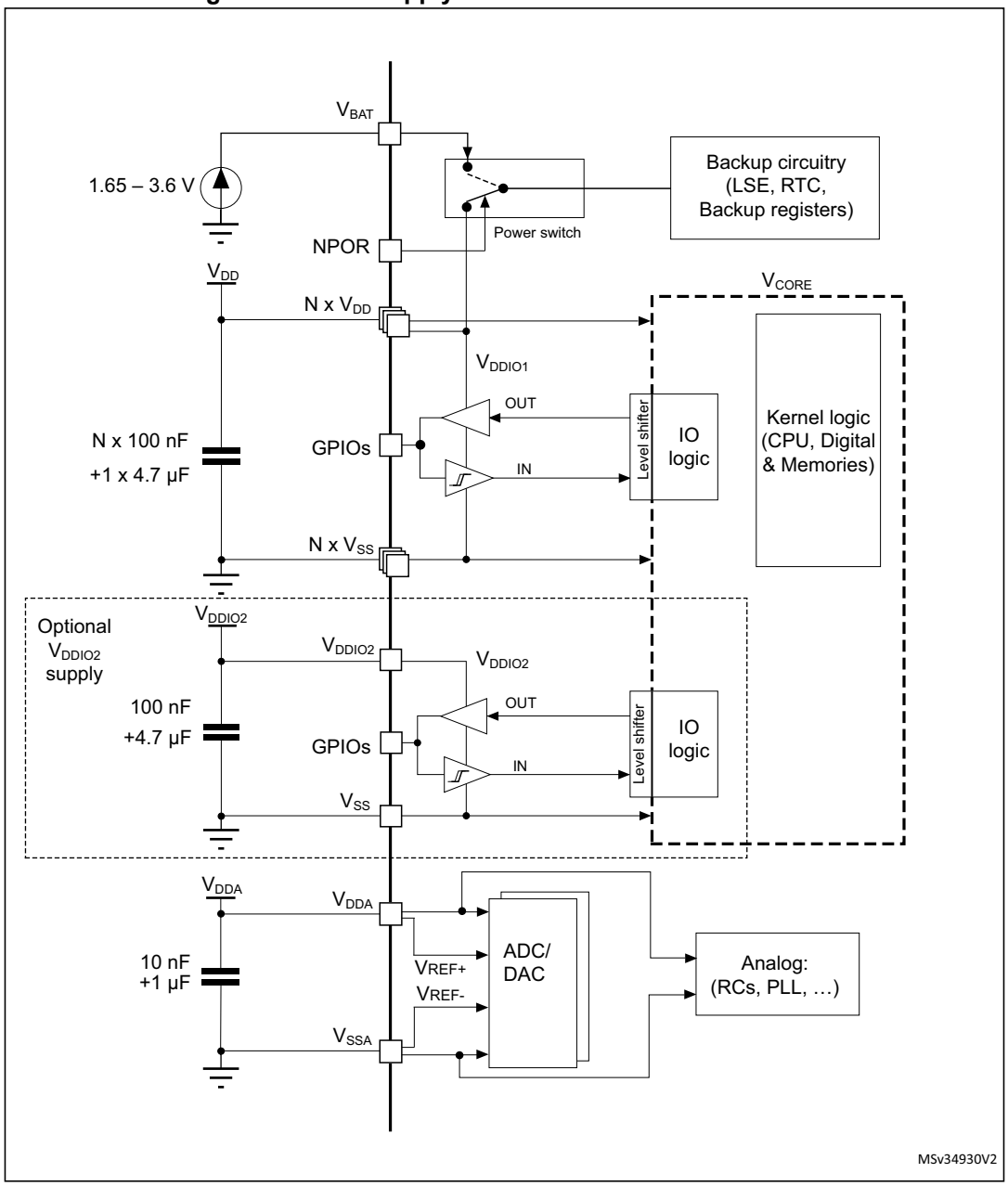
1. V_{DDIO2} power supply is not available on all parts, refer to the dedicated device datasheet to check if it is available.

Figure 1. Power supply scheme of STM32F0x1/x2 devices



1. Refer to the specific device datasheet to know which pins are supplied by V_{DD} or by V_{DDIO2}.

Figure 2. Power supply scheme of STM32F0x8 devices



1. Refer to the specific device datasheet to know which pins are supplied by V_{DD} or by V_{DDIO2}.

1.1.1 Independent analog converter supply

To improve conversion accuracy and to extend the supply flexibility, the analog domain has an independent power supply which can be separately filtered and shielded from noise on the PCB.

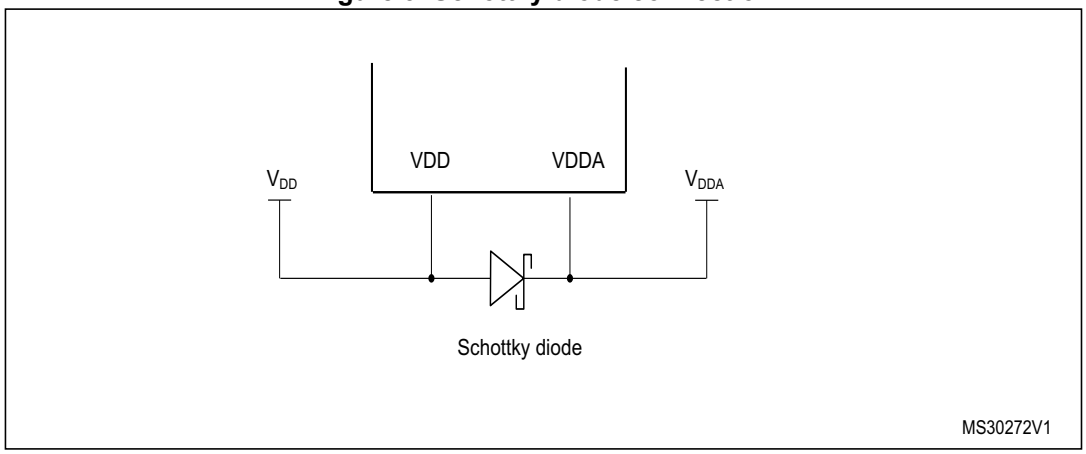
- The ADC and DAC voltage supply input is available on a separate VDDA pin.
- A separated supply ground connection is provided on pin VSSA.

The V_{DDA} supply can be equal to or higher than V_{DD}. This allows V_{DD} to stay low while still providing the full performance for the analog blocks.

When a single supply is used, V_{DDA} must be externally connected to V_{DD}. It is recommended to use an external filtering circuit in order to ensure a noise free V_{DDA}.

When V_{DDA} is different from V_{DD}, V_{DDA} must be always higher or equal to V_{DD}. To keep safe potential difference between V_{DDA} and V_{DD} during power-up/power-down, an external Schottky diode may be used between V_{DD} and V_{DDA}. Refer to the datasheet for the maximum allowed difference.

Figure 3. Schottky diode connection



1.1.2 Battery backup

To retain the content of the Backup registers when V_{DD} is turned off, the V_{BAT} pin can be connected to an optional standby voltage supplied by a battery or another source.

The V_{BAT} pin also powers the RTC unit, allowing the RTC to operate even when the main digital supply (V_{DD}) is turned off.

The switch to the V_{BAT} supply is controlled either by the POR/PDR circuitry embedded in the reset block on STM32F0x1/x2 devices, or by the NPOR pin on STM32F0x8 devices.

If no external battery is used in the application, it is recommended to connect V_{BAT} externally to V_{DD}.

1.1.3 Voltage regulator

The voltage regulator, when available on the device, is always enabled after reset.

It works under two different modes:

- Main (MR) is used in normal operating mode (Run).
- Low power (LPR) can be used in Stop mode where the power demand is reduced.

In standby mode the regulator is in power-down mode. In this mode, the regulator output is in high impedance and the kernel circuitry is powered down, inducing zero consumption and the loss of the register and SRAM contents. However, the following features are available if configured:

- Independent watchdog (IWDG): the IWDG is started by writing to its Key register or by a hardware option. Once started it cannot be stopped except by a reset.
- Real-time clock (RTC): configured by the RTCEN bit in the RTC domain control register (RCC_BDCR).
- Internal low speed oscillator (LSI): configured by the LSION bit in the Control/status register (RCC_CSR).
- External 32.768 kHz oscillator (LSE): configured by the LSEON bit in the RTC domain control register (RCC_BDCR).

Caution: In the STM32F0x8 devices the voltage regulator is bypassed, therefore the standby mode is not available.

1.2 Reset and power supply supervisor

1.2.1 Power-on reset (POR) / power-down reset (PDR) of the STM32F0x1/x2 devices

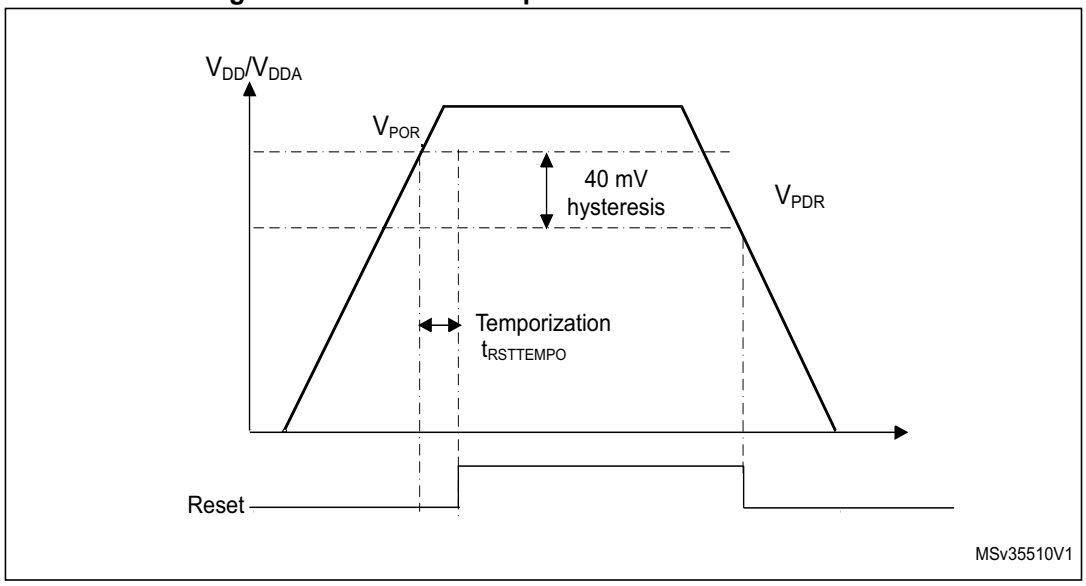
The device has an integrated power-on reset (POR) and power-down reset (PDR) circuits which are always active and ensure proper operation above a threshold of 2 V.

The device remains in Reset mode when the monitored supply voltage is below a specified threshold, $V_{POR/PDR}$, without the need for an external reset circuit.

- The POR monitors only the V_{DD} supply voltage. During the startup phase V_{DDA} must arrive first and be greater than or equal to V_{DD}.
- The PDR monitors both the V_{DD} and V_{DDA} supply voltages. However, the V_{DDA} power supply supervisor can be disabled (by programming a dedicated option bit $V_{DDA\_MONITOR}$) to reduce the power consumption if the application design ensures that V_{DDA} is higher than or equal to V_{DD}.

For more details on the power on / power down reset threshold, refer to the electrical characteristics section in the datasheet.

Figure 4. Power on reset/power down reset waveform



1.2.2 External power-on reset and power-down reset (NPOR) of the STM32F0x8 devices

To guarantee a proper power-on and power-down reset to the device, the NPOR pin must be held low until V_{DD} is valid or before turning off the supply. When V_{DD} is stable, the reset state can be exited by putting the NPOR pin in high impedance. The NPOR pin has an internal pull-up connected to VDDA.

1.2.3 System reset

A system reset sets all registers to their reset values, except the reset flags in the clock controller CSR register and the registers in the RTC domain. A system reset is generated when one of the following events occurs:

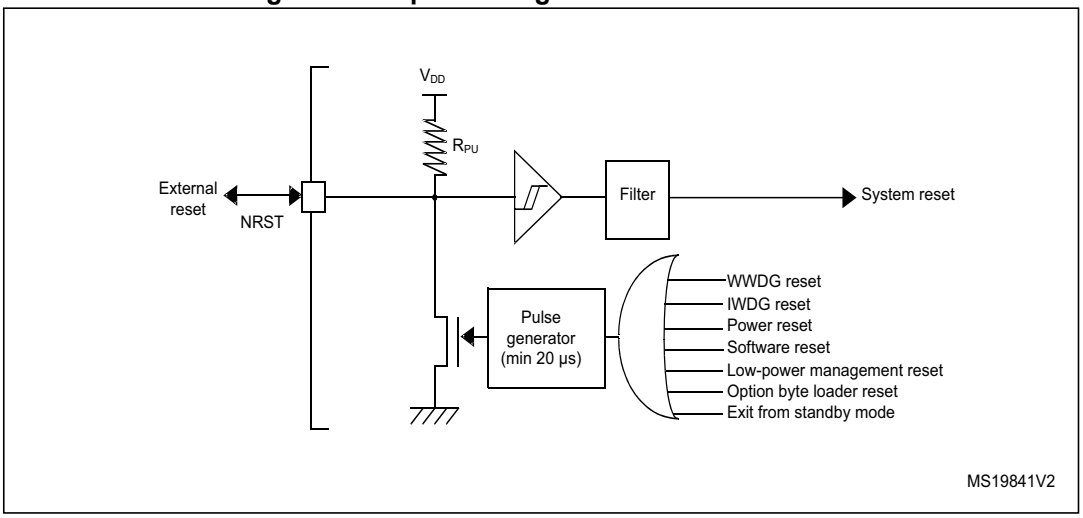
1. A low level on the NRST pin (external reset).
2. System window watchdog event (WWDG reset).
3. Independent watchdog event (IWDG reset).
4. A software reset (SW reset).
5. Low-power management reset.
6. Option byte loader reset.
7. Power reset

The reset source can be identified by checking the reset flags in the Control/Status register, RCC_CSR.

The RESET service routine vector is fixed at address 0x0000_0004 in the memory map.

The system reset signal provided to the device is output on the NRST pin. The pulse generator guarantees a minimum reset pulse duration of 20 µs for each internal reset source. In the case of an external reset, the reset is generated while the NRST pin is asserted low.

Figure 5. Simplified diagram of the reset circuit



Software reset

The SYSRESETREQ bit in Cortex-M0 Application Interrupt and Reset Control Register must be set to force a software reset on the device. Refer to the *Cortex®-M0 technical reference manual* for more details.

Low-power mode security reset

To prevent that critical applications mistakenly enter a low-power mode, two low-power mode security resets are available. If enabled in Option bytes, the resets are generated in the following conditions:

1. Entering Standby mode: this type of reset is enabled by resetting nRST_STDBY bit in User Option Bytes. In this case, whenever a Standby mode entry sequence is successfully executed, the device is reset instead of entering Standby mode.
2. Entering Stop mode: this type of reset is enabled by resetting nRST_STOP bit in User Option Bytes. In this case, whenever a Stop mode entry sequence is successfully executed, the device is reset instead of entering Stop mode.

Caution: In the STM32F0x8 devices, the voltage regulator is bypassed, therefore the standby mode is not available.

Option byte loader reset

The option byte loader reset is generated when OBL_LAUNCH (bit 13) is set in the FLASH_CR register. This bit launches the option byte loading by software.

Power reset

A power reset sets all registers to their reset values, except the RTC domain.

In the STM32F0x1/x2 devices, the power reset is generated when one of the following events occurs.

1. Power-on/power-down reset (POR/PDR reset)
2. Exiting Standby mode

For the STM32F0x8 devices, the Power-on reset must be provided externally. It is generated while the NPOR pin is low. For more information concerning NPOR, see *Section 1.2.2: External power-on reset and power-down reset (NPOR) of the STM32F0x8 devices*.

RTC domain reset

The RTC domain reset only affects the RTC, LSE, LSI and the backup registers. It is generated when one of the following events occurs.

1. Software reset, triggered by setting the BDRST bit in the RTC domain control register (RCC_BDCR).
2. POR reset if V_{BAT} has been disconnected when V_{DD} was low.

1.2.4 Programmable voltage detector (PVD) of the STM32F0x1/x2 devices

You can use the PVD to monitor the V_{DD} power supply by comparing it to a threshold selected by the PLS[2:0] bits in the Power control register (PWR_CR).

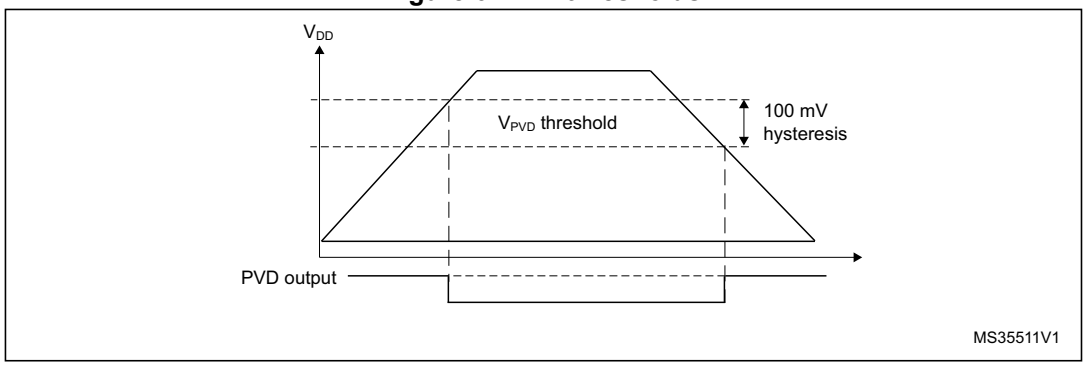
The PVD is enabled by setting the PVDE bit.

When enabled, the typical current consumption of this feature is 0.15 µA.

A PVDO flag is available, in the Power control/status register (PWR_CSR), to indicate if V_{DD} is higher or lower than the PVD threshold.

- This event is internally connected to the EXTI line16 and can generate an interrupt if enabled through the EXTI registers.
- The PVD output interrupt can be generated when V_{DD} drops below the PVD threshold and/or when V_{DD} rises above the PVD threshold depending on EXTI line16 rising/falling edge configuration. As an example the service routine could perform emergency shutdown tasks.

Figure 6. PVD thresholds



Caution: In the STM32F0x8 devices the input voltage must be fixed at 1.8 V, therefore the programmable voltage detector is not available.

2 Clocks

Different clock sources can be used to drive the system clock (SYSCLK):

- HSI 8 MHz RC oscillator clock (high-speed internal clock signal),
- HSE oscillator clock (high-speed external clock signal),
- PLL clock,
- HSI 48 MHz RC oscillator (high-speed internal clock signal) available in STM32F04x/7x/9x devices.

The devices have other secondary clock sources:

- 40 kHz low-speed internal RC (LSI RC) that drives the independent watchdog and, optionally, the RTC used for Auto-wakeup from the Stop/Standby modes.
- 32.768 kHz low-speed external crystal (LSE crystal) that optionally drives the RTC
- HSI 14MHz RC oscillator (HSI14) dedicated for ADC

Each clock source can be switched on or off independently when it is not used, to optimize the power consumption. Refer to the STM32F0x1/STM32F0x2/STM32F0x8 advanced ARM-based 32-bit MCUs reference manual (RM0091) for a description of the clock tree.

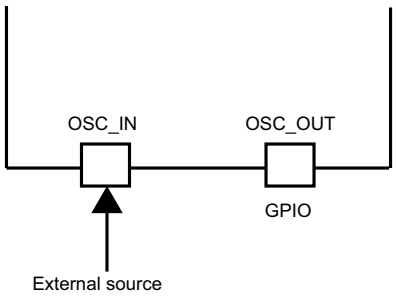
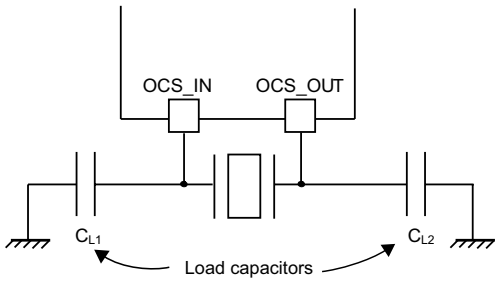
2.1 High speed external clock signal (HSE) OSC clock

The high speed external clock signal can be generated from two possible clock sources:

- HSE external crystal/ceramic resonator
- HSE user external clock

The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected resonator.

Figure 7. HSE/ LSE clock sources

Clock source	Hardware configuration
External clock	
Crystal/Ceramic resonators	

External crystal/ceramic resonator (HSE crystal)

The 4 to 32 MHz external oscillator has the advantage of producing a very accurate frequency on the main clock. Refer to the electrical characteristics section of the datasheet for more details about the associated hardware configuration.

The HSERDY flag in the Clock control register (RCC_CR) indicates if the HSE oscillator is stable or not. At startup, the clock is not released until this bit is set by hardware. An interrupt can be generated if enabled in the Clock interrupt register (RCC_CIR).

The HSE Crystal can be switched on and off using the HSEON bit in the Clock control register (RCC_CR).

External source (HSE bypass)

In this mode, an external clock source must be provided. It can have a frequency of up to 32 MHz. You select this mode by setting the HSEBYP and HSEON bits in the *Clock control register (RCC_CR)*. The external clock signal (square, sinus or triangle) with ~40-60% duty cycle depending on the frequency (refer to the datasheet) has to drive the OSC_IN pin while the OSC_OUT pin can be used a GPIO. See *Figure 7*.

2.2 LSE clock

The LSE crystal is a 32.768 kHz Low Speed External crystal or ceramic resonator. It has the advantage of providing a low-power but highly accurate clock source to the real-time clock peripheral (RTC) for clock/calendar or other timing functions.

The LSE crystal is switched on and off using the LSEON bit in RTC domain control register (RCC_BDCR). The crystal oscillator driving strength can be changed at runtime using the LSEDRV[1:0] bits in the RTC domain control register (RCC_BDCR) to obtain the best compromise between robustness and short start-up time on one side and low power-consumption on the other.

The LSERDY flag in the RTC domain control register (RCC_BDCR) indicates whether the LSE crystal is stable or not. At startup, the LSE crystal output clock signal is not released until this bit is set by hardware. An interrupt can be generated if enabled in the Clock interrupt register (RCC_CIR).

External source (LSE bypass)

In this mode, an external clock source must be provided. It can have a frequency of up to 1 MHz. You select this mode by setting the LSEBYP and LSEON bits in the RTC domain control register (RCC_BDCR). The external clock signal (square, sinus or triangle) has to drive the OSC32_IN pin while the OSC32_OUT pin can be used as GPIO. See *Figure 7*.

2.3 HSI 8 MHz clock

The HSI clock signal is generated from an internal 8 MHz RC oscillator and can be used directly as a system clock or divided by 2 to be used as PLL input. The HSI RC oscillator has the advantage of providing a clock source at low cost (no external components). It also has a faster startup time than the HSE crystal oscillator however, even with calibration, the frequency is less accurate than an external crystal oscillator or ceramic resonator.

Calibration

RC oscillator frequencies can vary from one chip to another due to manufacturing process variations, This is why each device is factory calibrated by ST for 1% accuracy at T_A=25°C.

Furthermore, it is possible to route the HSI clock to the MCO multiplexer. The clock can then be input to Timer 14 to allow the user to calibrate the oscillator.

2.4 LSI clock

The LSI RC acts as an low-power clock source that can be kept running in Stop and Standby mode for the independent watchdog (IWDG) and RTC. The clock frequency is around 40 kHz (between 30 kHz and 60 kHz). For more details, refer to the electrical characteristics section of the datasheets.

2.5 ADC clock

The ADC clock is either the dedicated 14 MHz RC oscillator (HSI14) or PCLK divided by 2 or 4. When the ADC clock is derived from PCLK, it is in an opposite phase with PCLK. The 14 MHz RC oscillator can be configured by software either to be turned on/off ("auto-off mode") by the ADC interface or to be always enabled.

2.6 HSI 48 MHz clock

2.6.1 HSI 48 MHz RC oscillator

The internal HSI 48 MHz RC oscillator is mainly dedicated to provide a high-precision reference clock to the USB by means of a special clock recovery circuitry (see *Section 2.6.2*), which could use the USB SOF signal or the LSE or an external pulse to adjust the frequency. This source can also be used as a system clock source when the system is in run mode; it is disabled as soon as the system enters in STOP or Standby mode. The CRS can be enabled or not depending on the status of the USB and the synchronization source selected.

2.6.2 HSI 48 MHz clock recovery system (CRS)

The clock recovery system is an advanced digital control interface to the internal fine-granularity trimmable RC oscillator HSI 48. The CRS is a powerful tool for evaluating the oscillator output frequency based on comparison with a selectable synchronization signal. It can adjust automatically the oscillator trimming based on the measured frequency error value, while keeping also the possibility of a manual trimming.

2.7 Clock security system (CSS)

The clock security system can be activated by software. In this case, the clock detector is enabled after the HSE oscillator startup delay, and disabled when this oscillator is stopped.

- If a failure is detected on the HSE oscillator clock, the oscillator is automatically disabled.
  - A clock failure event is sent to the break inputs of TIM1 advanced control timer and TIM15, TIM16 and TIM17 general purpose timers.
  - An interrupt is generated to inform the software about the failure (clock security system interrupt CSSI), allowing the MCU to perform recovery operations.
  - CSSI is linked to the Cortex®-M0 NMI (non-maskable interrupt) exception vector.
- If the HSE oscillator is used directly or indirectly as the system clock (indirectly means that it is used as the PLL input clock, and the PLL clock is used as the system clock), a detected failure causes a switch of the system clock to the HSI oscillator and the disabling of the external HSE oscillator. If the HSE oscillator clock (divided or not) is the clock entry of the PLL that is being used as a system clock when the failure occurs, the PLL is disabled too.

For details refer to STM32F0x1/STM32F0x2/STM32F0x8 advanced ARM-based 32-bit MCUs reference manual (RM0091) available from STMicroelectronics website *www.st.com*.

3 Boot configuration

In the STM32F0xx, three different boot modes can be selected through the BOOT0 pin and boot configuration bits nBOOT1, BOOT_SEL and nBOOT0 in the User option byte, as shown in the following table.

Table 3. Boot modes(1)

Boot mode configuration				Mode
nBOOT1 bit	BOOT0 pin	BOOT_SEL bit	nBOOT0 bit	
x	0	1	x	**Main Flash memory** is selected as boot space(2)
1	1	1	x	**System memory** is selected as boot space
0	1	1	x	**Embedded SRAM** is selected as boot space
x	x	0	1	**Main Flash memory** is selected as boot space
1	x	0	0	**System memory** is selected as boot space
0	x	0	0	**Embedded SRAM** is selected as boot space

1. Grey options are available on STM32F04x devices only.
2. For STM32F04x devices, see also Empty check description.

The boot mode configuration is latched on the 4th rising edge of SYSCLK after a reset. It is up to the user to set boot mode configuration related to the required boot mode. The boot mode configuration is also re-sampled when exiting from Standby mode. Consequently they must be kept in the required Boot mode configuration in Standby mode. After this startup delay has elapsed, the CPU fetches the top-of-stack value from address 0x00000000, then starts code execution from the boot memory at 0x00000004.

Depending on the selected boot mode, main Flash memory, system memory or SRAM is accessible as follows:

- Boot from main Flash memory: the main Flash memory is aliased in the boot memory space (0x00000000), but still accessible from its original memory space (0x08000000). In other words, the Flash memory contents can be accessed starting from address 0x00000000 or 0x08000000.
- Boot from system memory: the system memory is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x1FFF EC00 on STM32F03x and STM32F05x devices, 0x1FFF C400 on STM32F04x devices and 0x1FFF C800 on STM32F07x devices).
- Boot from the embedded SRAM: the SRAM is aliased in the boot memory space (0x0000 0000), but it is still accessible from its original memory space (0x2000 0000).

Empty check

On STM32F04x devices only, internal empty check flag is implemented to allow easy programming of the virgin devices by the boot loader. This flag is used when BOOT0 pin is defining Main Flash memory as the target boot space. When the flag is set, the device is considered as empty and System memory (boot loader) is selected instead of the Main Flash as a boot space to allow user to program the Flash memory.

This flag is updated only during Option bytes loading: it is set when the content of the address 0x08000 0000 is read as 0xFFFF FFFF, otherwise it is cleared. It means a power on or setting of OBL_LAUNCH bit in FLASH_CR register is needed to clear this flag after programming of a virgin device to execute user code after System reset.

Note: If the device is programmed for a first time but the Option bytes are not reloaded, the device will still select System memory as a boot space after a System reset. The boot loader code is able to detect this situation and will change the boot memory mapping to Main Flash and perform a jump to user code programmed there.

Embedded boot loader

The embedded boot loader is located in the System memory, programmed by ST during production. It is used to reprogram the Flash memory using one of the following serial interfaces:

- USART on pins PA14/PA15 or PA9/PA10,
- I2C on pins PB6/PB7 (STM32F04x and STM32F07x devices only),
- USB DFU interface (STM32F04x, STM32F072 and STM32F078 devices only).

For further details, please refer to AN2606.

4 Debug management

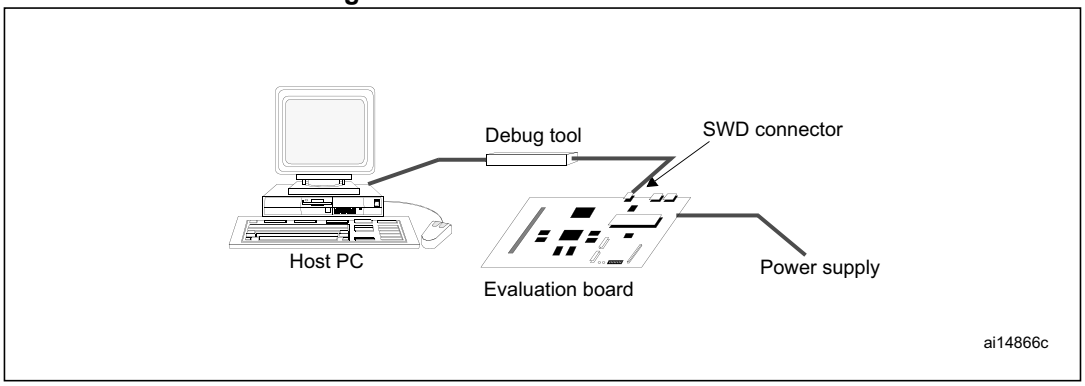
4.1 Introduction

The host/target interface is the hardware equipment that connects the host to the application board. This interface is made of three components: a hardware debug tool, an SWD connector and a cable connecting the host to the debug tool.

Figure 8 shows the connection of the host to the evaluation board (STM320518_EVAL).

The STM320518_EVAL evaluation board embeds the debug tools (ST-LINK). Consequently, it can be directly connected to the PC through a USB cable.

Figure 8. Host-to-board connection



4.2 SWD port (serial wire debug)

The STM32F0xx core integrates the serial wire debug port (SW-DP). It is an ARM® standard CoreSight™ debug port with a 2-pin (clock + data) interface to the debug access port.

4.3 Pinout and debug port pins

The STM32F0xx MCU is offered in various packages with varying numbers of available pins.

4.3.1 Serial wire debug (SWD) pin assignment

The same SWD pin assignment is available on all STM32F0xx packages.

Table 4. SWD port pins

SWD pin name	SWD port		Pin assignment
	Type	Debug assignment	
SWDIO	I/O	Serial wire data input/output	PA13
SWCLK	I	Serial wire clock	PA14

4.3.2 SWD pin assignment

After reset (SYSRESETn or PORESETn), the pins used for the SWD are assigned as dedicated pins which are immediately usable by the debugger host.

However, the MCU offers the possibility to disable the SWD, therefore releasing the associated pins for general-purpose I/O (GPIO) usage. For more details on how to disable SWD port, refer to the RM0091 section on I/O pin alternate function multiplexer and mapping.

4.3.3 Internal pull-up and pull-down on SWD pins

Once the SWD I/O is released by the user software, the GPIO controller takes control of these pins. The reset states of the GPIO control registers put the I/Os in the equivalent states:

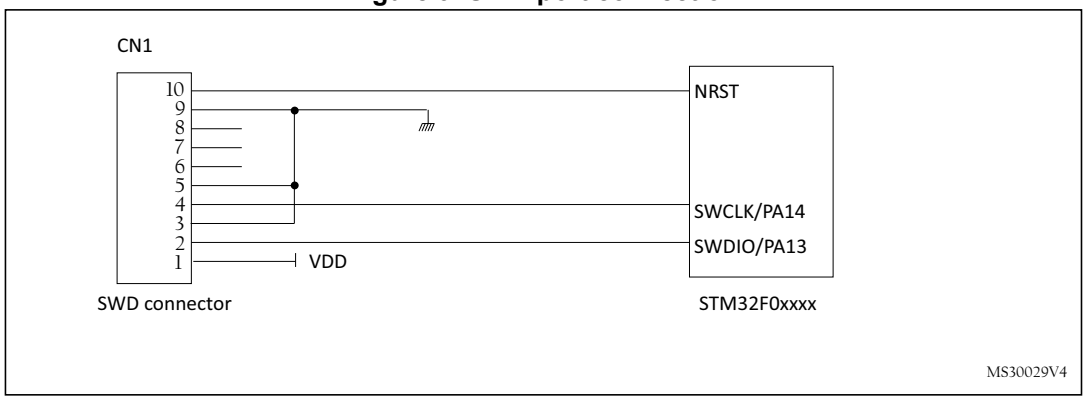
- SWDIO: alternate function pull-up
- SWCLK: alternate function pull-down

Having embedded pull-up and pull-down resistors removes the need to add external resistors.

4.3.4 SWD port connection with standard SWD connector

Figure 9 shows the connection between the STM32F0xx and a standard SWD connector.

Figure 9. SWD port connection



Caution: In STM32F07x/04x devices, the SWCLK and SWDIO pins are supplied by the VDDIO2 domain. Therefore, you must provide VDDIO2 supply instead of VDD to the SWD connector in order to ensure correct behavior of the debugger tool.

5 Recommendations

5.1 Printed circuit board

For technical reasons, it is best to use a multilayer printed circuit board (PCB) with a separate layer dedicated to ground (V_{SS}) and another dedicated to the V_{DD} supply. This provides good decoupling and a good shielding effect. For many applications, economical reasons prohibit the use of this type of board. In this case, the major requirement is to ensure a good structure for ground and for the power supply.

5.2 Component position

A preliminary layout of the PCB must make separate circuits:

- High-current circuits
- Low-voltage circuits
- Digital component circuits
- Circuits separated according to their EMI contribution. This will reduce cross-coupling on the PCB that introduces noise.

5.3 Ground and power supply (V_{DD}, V_{DDA}, V_{DDIO2})

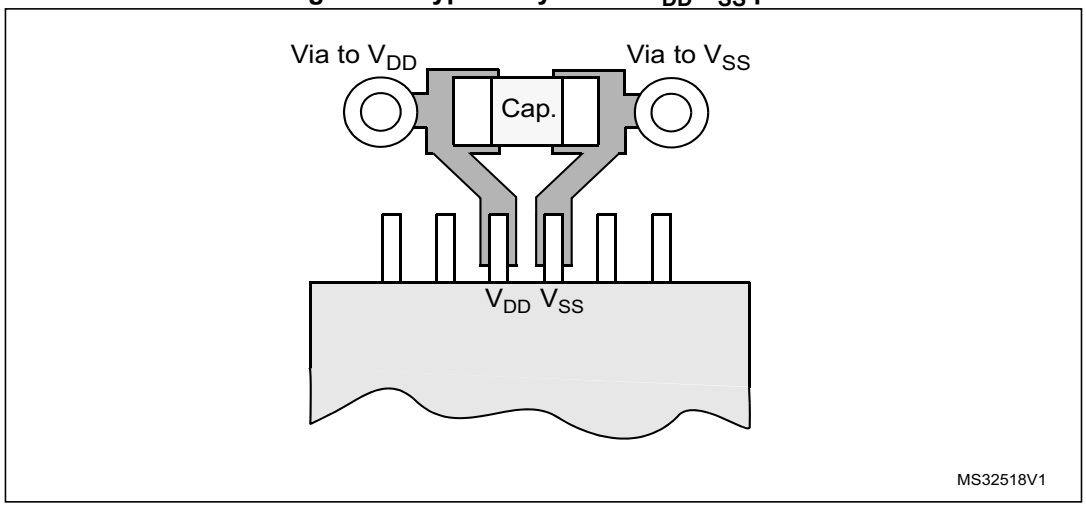
Every block (noisy, low-level sensitive, digital, etc.) should be grounded individually and all ground returns should be to a single point. Loops must be avoided or have a minimum area. In order to improve analog performance, you must use separate supply sources for V_{DD}/V_{DDIO2} and V_{DDA}, and place the decoupling capacitors as close as possible to the device. The power supplies should be implemented close to the ground line to minimize the area of the supplies loop. This is due to the fact that the supply loop acts as an antenna, and is therefore the main transmitter and receiver of EMI. All component-free PCB areas must be filled with additional grounding to create a kind of shielding (especially when using single-layer PCBs).

5.4 Decoupling

All power supply and ground pins must be properly connected to the power supplies. These connections, including pads, tracks and vias should have as low an impedance as possible. This is typically achieved with thick track widths and, preferably, the use of dedicated power supply planes in multilayer PCBs.

In addition, each power supply pair should be decoupled with 100 nF filtering ceramic capacitor and a chemical capacitor of about 4.7 µF connected between the supply pins of the STM32F0xx device. These capacitors need to be placed as close as possible to, or below, the appropriate pins on the underside of the PCB. Typical values are 10 nF to 100 nF, but exact values depend on the application needs. *Figure 10* shows the typical layout of such a V_{DD}/V_{SS} pair.

Figure 10. Typical layout for V_{DD}/V_{SS} pair



5.5 Other signals

When designing an application, the EMC performance can be improved by closely studying:

- Signals for which a temporary disturbance affects the running process permanently (such as interrupts and handshaking strobe signals, but not LED commands). For these signals, a surrounding ground trace, shorter lengths and the absence of noisy and sensitive traces nearby (crosstalk effect) improve EMC performance.
- Digital signals: the best possible electrical margin must be reached for the two logical states and slow Schmitt triggers are recommended to eliminate parasitic states.
- Noisy signals (clock, etc.)
- Sensitive signals (high impedance, etc.)

5.6 Unused I/Os and features

All microcontrollers are designed for a variety of applications and often a particular application does not use 100% of the MCU resources.

To increase EMC performance and avoid extra power consumption, unused clocks, counters or I/Os, should not be left free. I/Os should be connected to a fixed logic level of 0 or 1 by an external or internal pull-up or pull-down on the unused I/O pin. The other option is to configure GPIO as output mode using software. Unused features should be frozen or disabled, which is their default value.

6 Reference design

6.1 Description

The reference design shown in *Figure 11*, introduces the STM32F051, a highly integrated microcontroller running at 48 MHz, that combines the Cortex®-M0 32-bit RISC CPU core with 64 Kbytes of embedded Flash memory and 8 Kbytes of SRAM.

For the STM32F058 reference design, refer to *Figure 12*. These reference design can be tailored to any other STM32F058 device with a different package, using the pin correspondence given in the corresponding datasheet.

For the STM32F072 reference design, refer to *Figure 13*. These reference design can be tailored to any other STM32F072 device with a different package, using the pin correspondence given in the corresponding datasheet.

6.1.1 Clock

Two clock sources are used for the microcontroller:

- HSE: X1– 8 MHz crystal for the STM32F0xx microcontroller
- LSE: X2– 32.768 kHz crystal for the embedded RTC

Refer to *Section 2: Clocks on page 15*.

6.1.2 Reset

The reset signal in *Figure 11* or *Figure 12* is active low. The reset sources include:

- Reset button (B1)
- Debugging tools via the connector CN1

Refer to *Section 1.2.3: System reset on page 12*.

6.1.3 STM32F0x8 power-on reset

On the STM32F0x8 line, the power-on reset signal is active low and must be provided externally by the application on NPOR pin. It is maintained to V_{DDA} through an integrated pull-up resistor.

Refer to *Section 1.2.2: External power-on reset and power-down reset (NPOR) of the STM32F0x8 devices on page 11*.

6.1.4 Boot mode

The boot option is configured by setting BOOT0 through switch SW1 and option bit nBOOT1. Refer to *Section 3: Boot configuration on page 20*.

6.1.5 SWD interface

The reference design shows the connection between the STM32F0xx and a standard SWD connector. Refer to *Section 4: Debug management on page 22*.

Note: It is recommended to connect the reset pin in order to be able to reset the application from the tool.

6.1.6 Power supply

Refer to *Section 1.1: Power supply schemes*.

6.1.7 Pinouts and pin description

Please refer to the STM32F0xx datasheets available on www.st.com for the pinout information and pin description of each device.

6.2 Component references

Table 5. Mandatory components

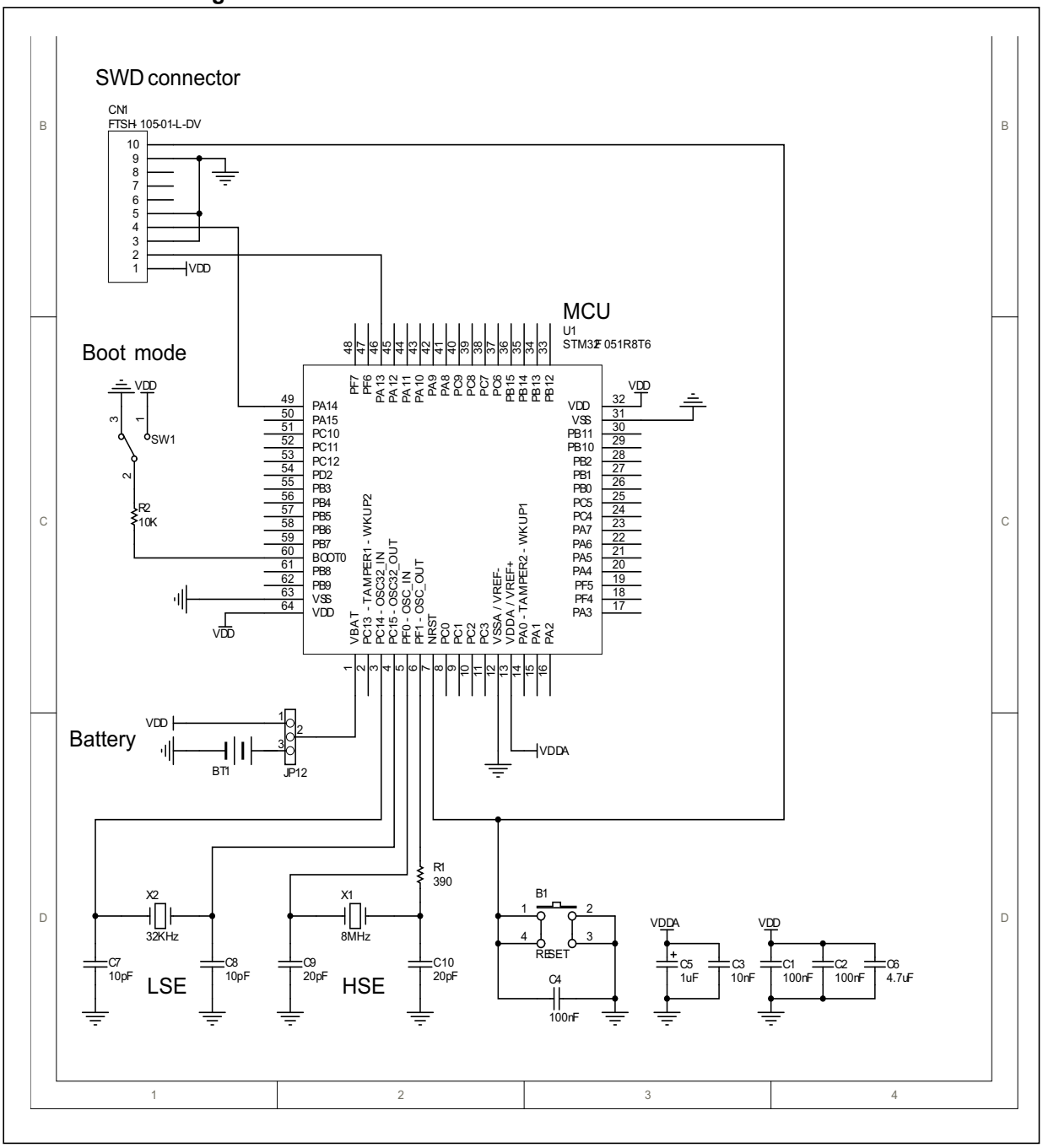
Component	Reference	Value	Quantity	Comments
Microcontroller	U1, U2 or U3	STM32F051R8, STM32F058R8 or STM32F072R8	1	64-pin package
Capacitor	C1/C2	100 nF	2	Ceramic capacitors (decoupling capacitors)
Capacitor	C3	10 nF	1	Ceramic capacitor (decoupling capacitor)
Capacitor	C5	1 µF	1	Used for VDDA
Capacitor	C6	4.7 µF	1	Used for VDD

Note: Depending on the microcontroller used, refer to the corresponding reference schematics.

Table 6. Optional components

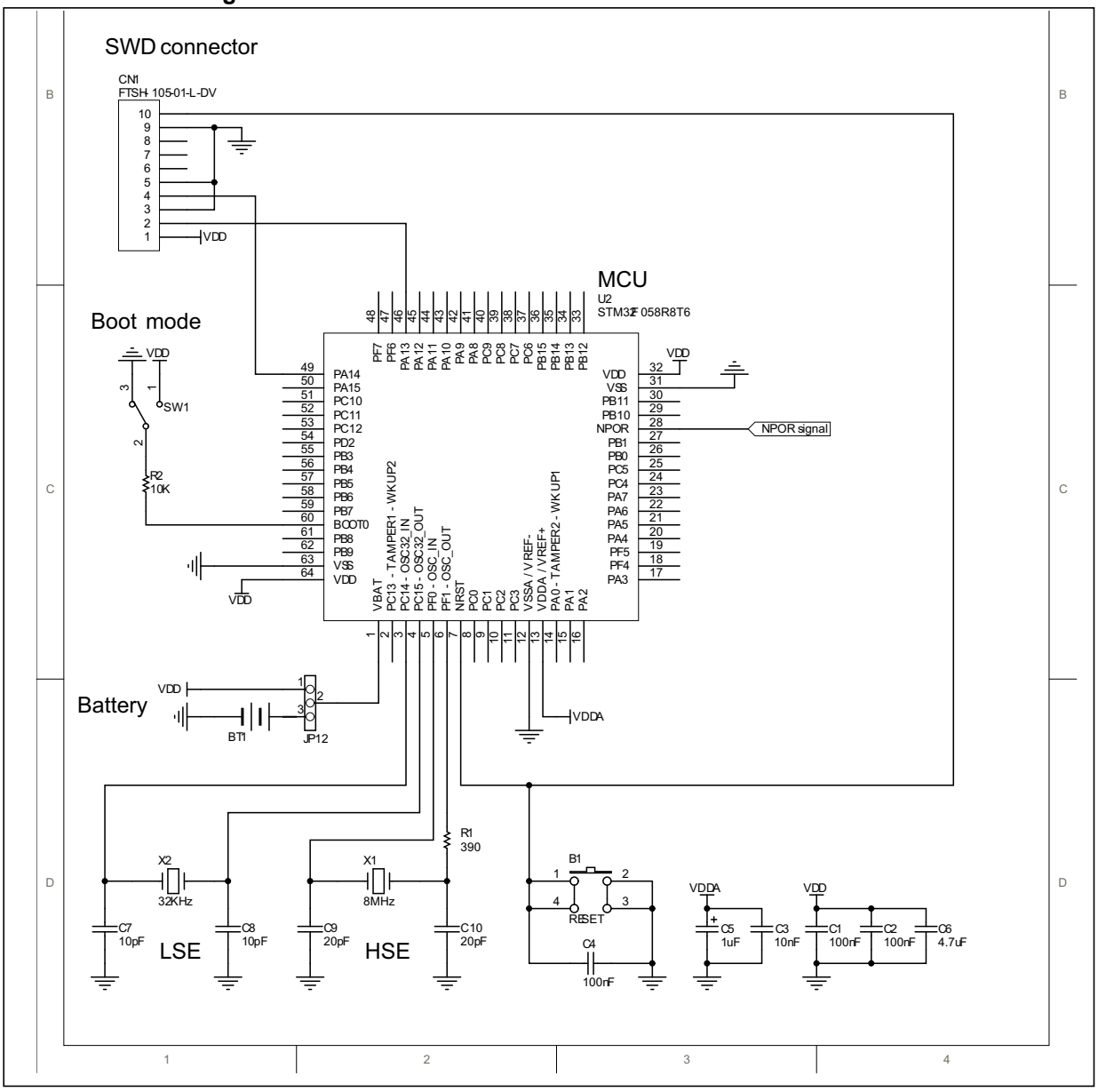
Component	Reference	Value	Quantity	Comments
Resistor	R1	390 Ω	1	Used for HSE: the value depends on the crystal characteristics. This value is given only as a typical example.
Resistor	R2	10 KΩ	1	Used for BOOT0 pin
Capacitor	C4	100 nF	1	Ceramic capacitor for RESET button
Capacitor	C13	4.7 µF	1	Used for VDDIO2
Capacitor	C12	100 µF	1	Ceramic capacitor (decoupling capacitor)
Capacitor	C7/C8	10 pF	2	Used for LSE: the value depends on the crystal characteristics.
Capacitor	C9/C10	20 pF	2	Used for HSE: the value depends on the crystal characteristics.
Quartz	X1	8 MHz	1	Used for HSE
Quartz	X2	32 kHz	1	Used for LSE
Battery	BT1	3V	1	If no external battery is used in the application, it is recommended to connect V_{BAT} externally to V_{DD}
Switch	SW1	-	1	Used to select the correct boot mode.
Push-button	B1	-	1	Used as reset button
SWD connector	CN1	FTSH-105-01-L-DV	1	Used for program/debug of the MCU

Figure 11. STM32F051R8 microcontroller reference schematic



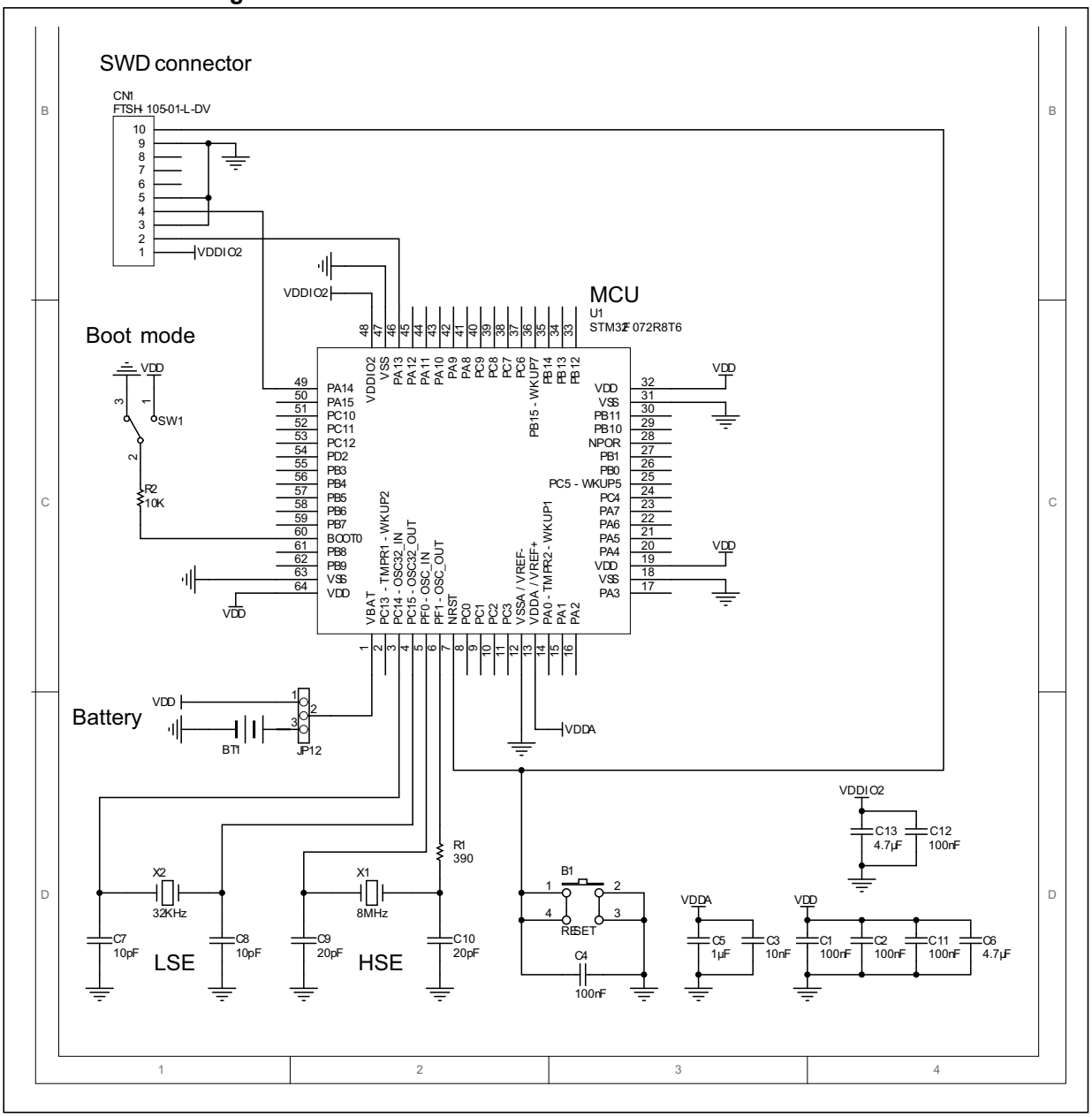
Note: If no external battery is used in the application, it is recommended to connect V_{BAT} externally to V_{DD}.

Figure 12. STM32F058R8 microcontroller reference schematic



Note: If no external battery is used in the application, it is recommended to connect V_{BAT} externally to V_{DD}.

Figure 13. STM32F072 microcontroller reference schematics



7 Hardware migration from STM32F1 to STM32F0 series

The entry-level STM32F0xx and general-purpose STM32F1xxx families are pin-to-pin compatible. All peripherals shares the same pins in the two families, but there are some minor differences between packages.

The transition from the STM32F1 series to the STM32F0 series is simple as only a few pins are impacted. The impacted pins are shown in bold in the following tables.

Table 7. STM32F1 series and STM32F03x/5x device pinout differences

LQFP64	LQFP48	STM32 F1 series	STM32F03x/5x devices	
		Pinout	Pinout STM32F031/51	Pinout STM32F038/58
5	5	**PD0** - OSC_IN	**PF0** - OSC_IN	**PF0** - OSC_IN
6	6	**PD1** - OSC_OUT	**PF1** - OSC_OUT	**PF1** - OSC_OUT
18	-	**VSS_4**	**PF4**	**PF4**
19	-	**VDD_4**	**PF5**	**PF5**
28	20	**BOOT1 - PB2**	PB2	**NPOR**
47	35	**VSS_2**	**PF6**	**PF6**
48	36	**VDD_2**	**PF7**	**PF7**

Table 8. STM32F1 series and STM32F04x device pinout differences

LQFP48	STM32F1 series	STM32F04x devices	
	Pinout	Pinout STM32F042	Pinout STM32F048
5	**PD0** - OSC_IN	**PF0** - OSC_IN	**PF0** - OSC_IN
6	**PD1** - OSC_OUT	**PF1** - OSC_OUT	**PF1** - OSC_OUT
20	**BOOT1 - PB2**	PB2	**NPOR**
35	**VSS_2**	**PF6**	**PF6**
36	**VDD_2**	**VDDIO2**	**VDDIO2**
44	BOOT0	BOOT0 **- PF11**	BOOT0 - **PF11**

Table 9. STM32F1 series and STM32F07x device pinout differences

LQFP100	STM32 F1 series	STM32F07x devices	
	Pinout	Pinout STM32F071/72	Pinout STM32F078
10	**VSS_5**	**PF9**	**PF9**
11	**VDD_5**	**PF10**	**PF10**
12	OSC_IN	**PF0 -** OSC_IN	**PF0 -** OSC_IN

Table 9. STM32F1 series and STM32F07x device pinout differences (continued)

LQFP100	STM32 F1 series	STM32F07x devices	
	Pinout	Pinout STM32F071/72	Pinout STM32F078
13	OSC_OUT	**PF1 -** OSC_OUT	**PF1 -** OSC_OUT
19	**VSSA**	**PF2**	**PF2**
20	**VREF -**	**VSSA**	**VSSA**
21	**VREF +**	**VDDA**	**VDDA**
22	**VDDA**	**PF3**	**PF3**
37	**BOOT1 -** PB2	PB2	**NPOR**
73	**NC**	**PF6**	**PF6**
75	**VDD_2**	**VDDIO2**	**VDDIO2**

Table 10. STM32F1 series and STM32F09x device pinout differences

LQFP100	STM32 F1 series	STM32F09x devices	
	Pinout	Pinout STM32F091	Pinout STM32F098
10	**VSS_5**	**PF9**	**PF9**
11	**VDD_5**	**PF10**	**PF10**
12	OSC_IN	**PF0 -** OSC_IN	**PF0 -** OSC_IN
13	OSC_OUT	**PF1 -** OSC_OUT	**PF1 -** OSC_OUT
19	**VSSA**	**PF2**	**PF2**
20	**VREF -**	**VSSA**	**VSSA**
21	**VREF +**	**VDDA**	**VDDA**
22	**VDDA**	**PF3**	**PF3**
37	**BOOT1 -** PB2	PB2	**NPOR**
73	**NC**	**PF6**	**PF6**
75	**VDD_2**	**VDDIO2**	**VDDIO2**
94	BOOT0	BOOT0-**PF11**	BOOT0-**PF11**

8 Revision history

Table 11. Document revision history

Date	Revision	Changes
11-Jul-2012	1	Initial release.
11-Feb-2013	2	Added *Chapter 3: Power supplies of the STM32F06x family*. Modified *Chapter 6: Reference design*. Modified *Table 7: STM32F1 series and STM32F03x/5x device pinout differences*.
02-Oct-2014	3	Document reformatted. Renamed part numbers. Updated: – *Section 1: Power supplies and reset sources of the STM32F0xx family*: grouped together information for STM32F05x and STM32F06x devices. – *Section 2: Clocks* – *Section 4.3.4: SWD port connection with standard SWD connector* – *Section 5.3: Ground and power supply (VDD, VDDA, VDDIO2)* – *Section 3: Boot configuration* – *Table 1: Applicable products* and *Table 7: STM32F1 series and STM32F03x/5x device pinout differences* – *Figure 1: Power supply scheme of STM32F0x1/x2 devices*, *Figure 5: Simplified diagram of the reset circuit* and *Figure 9: SWD port connection* – *Section 7: Hardware migration from STM32F1 to STM32F0 series* – *Figure 11: STM32F051R8 microcontroller reference schematic* Added: – *Figure 2: Power supply scheme of STM32F0x8 devices* – *Section 2.6: HSI 48 MHz clock* – *Table 9: STM32F1 series and STM32F07x device pinout differences* – *Table 8: STM32F1 series and STM32F04x device pinout differences* – *Figure 12: STM32F058R8 microcontroller reference schematic* – *Figure 13: STM32F072 microcontroller reference schematics* – *Table 10: STM32F1 series and STM32F09x device pinout differences*
07-Nov-2014	4	Improved the rendering of *Figure 7: HSE/ LSE clock sources*.