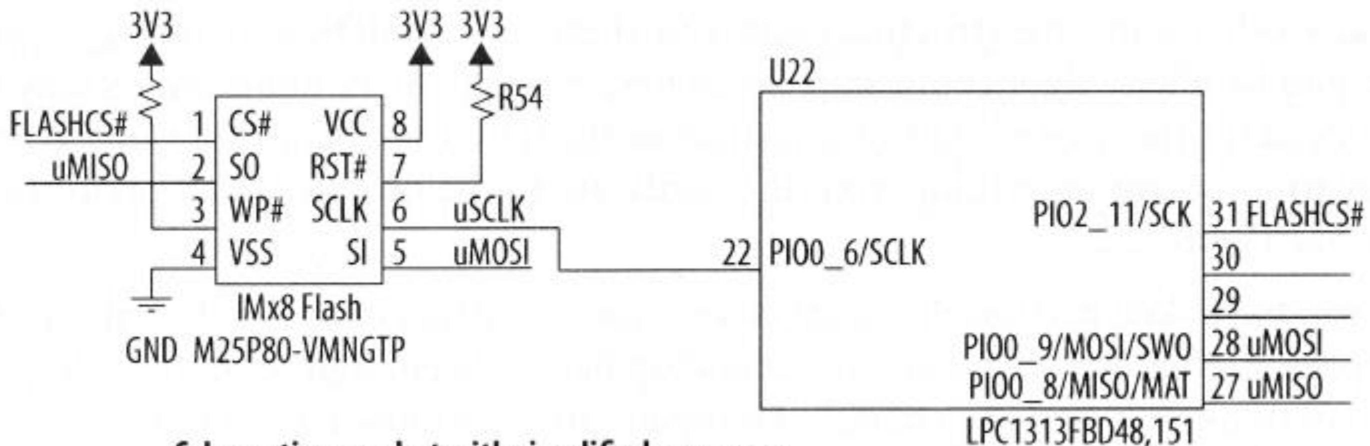
Föreläsning 3 – Hårdvara

- Kopplingsschema
- Blockschema
- Digitala och Analoga interface
- Kommunikation
- STM32

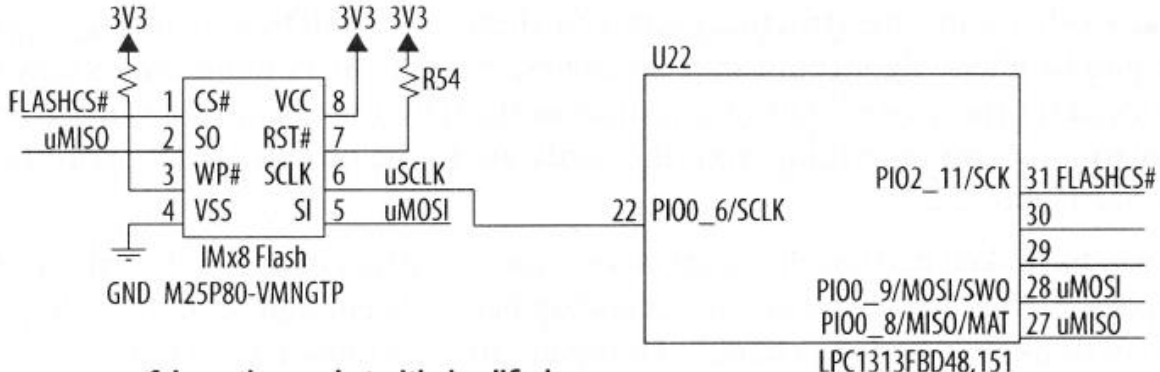
Creating a System Architecture*



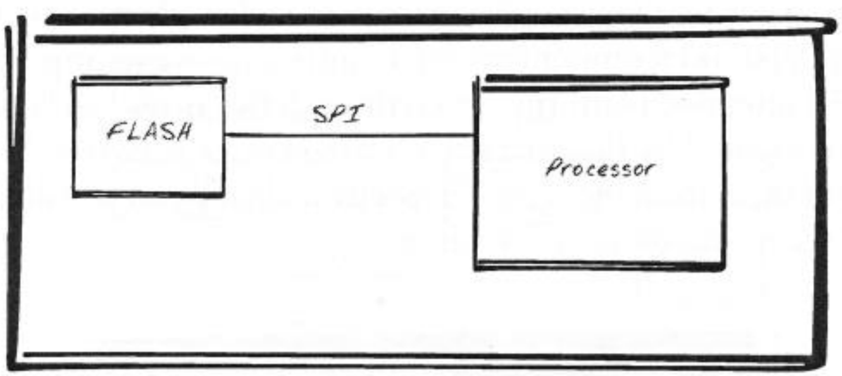
Schematic snapshot with simplified processor

* Source: White, Making Embedded Systems

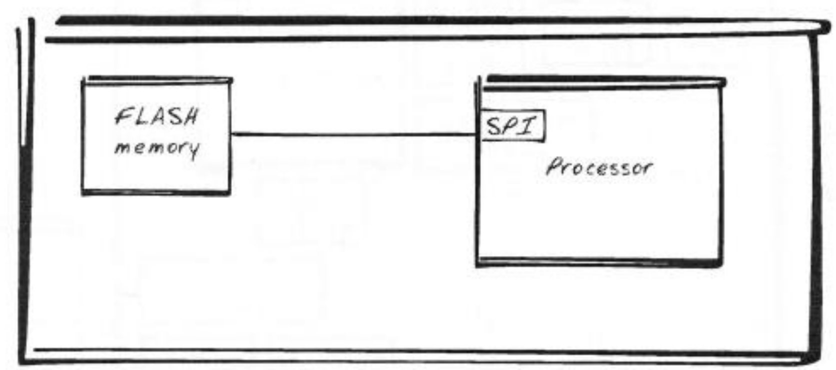
Block Diagram



Schematic snapshot with simplified processor



Hardware block diagram



Software architecture block diagram

Hardware Block Diagram **Software Architecture Block Diagram**

Matningsspänningar, några frågeställningar

- Hur ska vi hantera matningsspänningar?
- Batteridrift?
- Kretsar med olika matningsspänning 3V, 5V, går det att kombinera?
- Vad betyder etta och nolla i spänning?
- Hur ser det ut med matningsspänning på Discoverykortet?

I/O pins, GPIO

- Output
  - Push-pull
  - Open drain
  - Disabled
- Input
  - Floating
  - Pull up
  - Pull down

Check maximum allowable current in datasheet

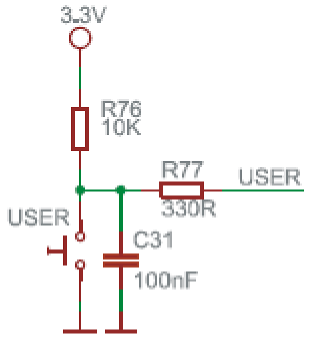
Input voltage usually limited by supply voltage
STM32: some of the inputs are 5V tolerant, check in data sheet!

Debouncing

- Mechanical contacts bounces
- Hardware debouncing
  - SR-latch
  - RC-net
- Software debouncing
  - Read and check if stable
  - Do not use for interrupt

Link: http://www.ganssle.com/debouncing.pdf

IAR Development kit



Sensors

- Light sensor
- Temperature sensor
- Pressure sensor
- Magnetic field sensor
- MEMS (Microelectromechanical system)
  - Accelerometer
  - Gyroscope
- ...

Analog input

- Analog to digital converter
  - Sample frequency, how fast (throughput, bandwidth)
  - Number of bits (8 bits=256 levels)
- Signal processing
  - Sampling, Nyquist theorem, filtering
  - Fourier transform – understand the math
  - Fast Fourier transform – real world problem

Link: The Scientist and Engineer's Guide to Digital Signal Processing http://www.dspguide.com/

Actuators

- Motor
  - Stepper motor
    - No feedback needed
    - Digitally controlled
    - Expensive
  - DC motor
    - Voltage determine speed
    - PWM control
    - Precise position needs feedback
    - Cheap

How do you know the start position? Detection of start point needed.

Timer in µC can be used for PWM control

Drive Circuitry

- Drive high current, external transistor switch or driver circuit
  - Inductive load, watch up for spikes!

- H-bridge, reverse current direction

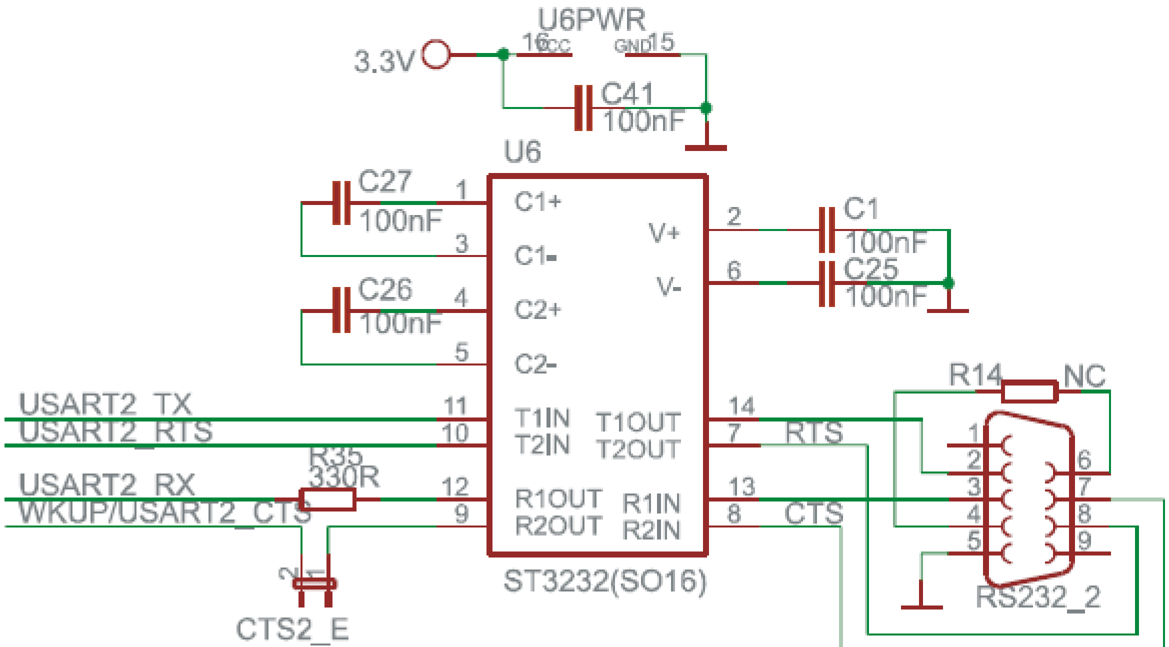
Kommunikation

– Mellan kretsar: I^2C, SPI, USART/UART
– Kabel: RS232, UART, USB
– Trådlös kommunikation:
  - ZigBee
  - WiFi
  - 6LowPAN (Internet of Things)
  - Bluetooth

Asyncronous serial interface

- Serial port, RS232, USART, UART (Universal asynchronous receiver/transmitter)
- Asyncronous, no clock
- Baudrate (bits/sec) need to be known to identify bytes transmitted
- 1 startbit, 7-8 bits, 1-1.5 stopbits
- Voltage level 3 to 15 V and -3 to -15 V
- VDD= 3V for STM32 microcontroller, how to drive voltage level?

RS232 Voltage level conversion



Many circuits available for conversion

Seriell port - RS232

- UART
  - I vår microcontroller används UART (USART) för asynkron seriell kommunikation
  - Två kretsar som har 3 V matningsspänning kan direkt kopplas ihop för seriekommunikation. Rx, Tx
  - Fel i kommunikationen? Troligen fel hastighet. Mät med oscilloskop!
  - Vad betyder 9600 baud? Hur lång är en bit i sekunder?

SPI – Serial Peripheral Interface bus

- Synchronous serial communication
- 4 wire interface + ground
  - Serial Data Out (SDO, MOSI= Master Out Slave In)
  - Serial Data In (SDI, MISO= Master In Slave Out)
  - Clock (SCLK)
  - Select signal (SS = Slave Select)
- Simple interface, can easily be programmed by Bit-Banging

I^2C – Inter Integrated Circuit

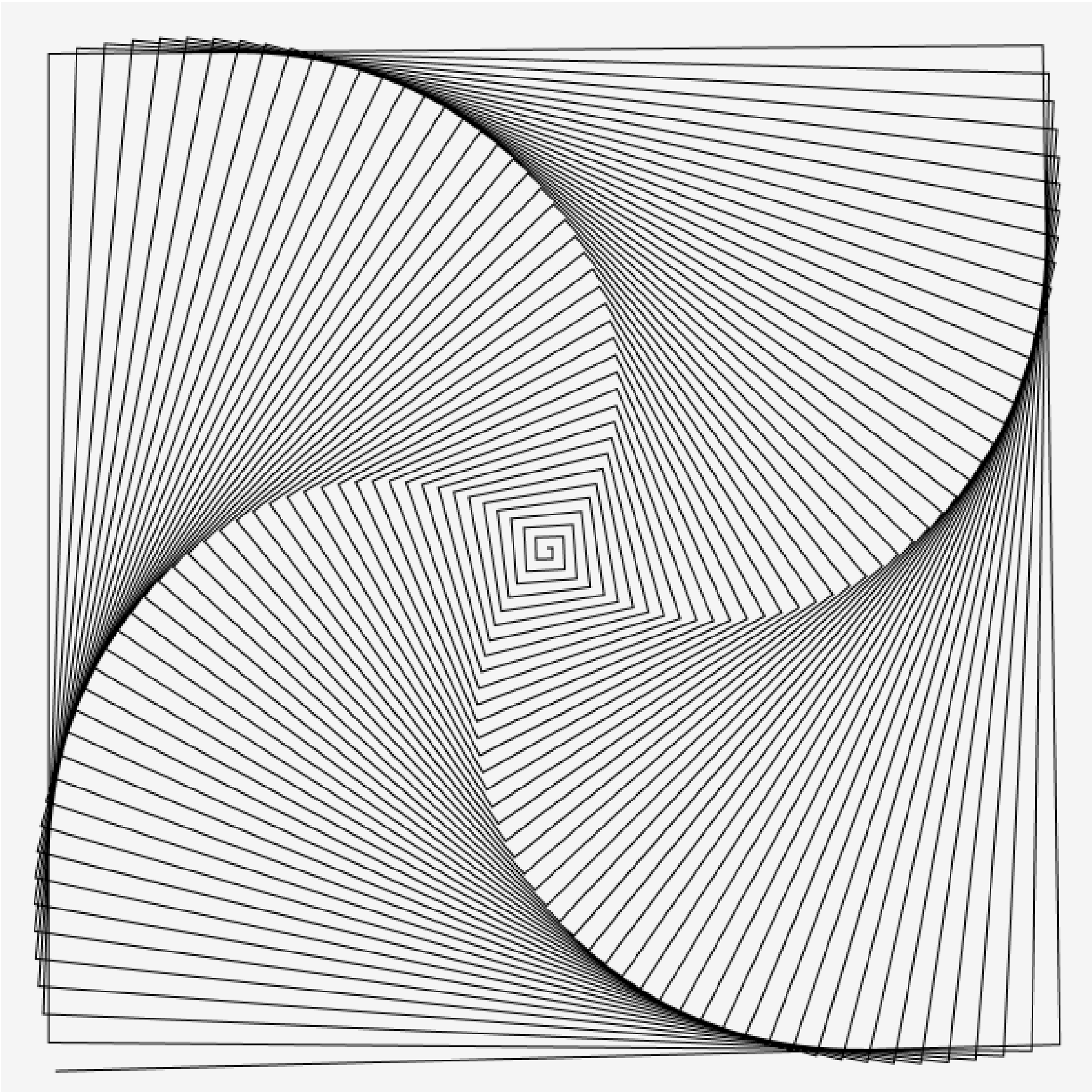
- Two wire interface + ground
  - SCL Serial clock
  - SDA Serial data

Pull up needed, several circuits connected to same line with several outputs

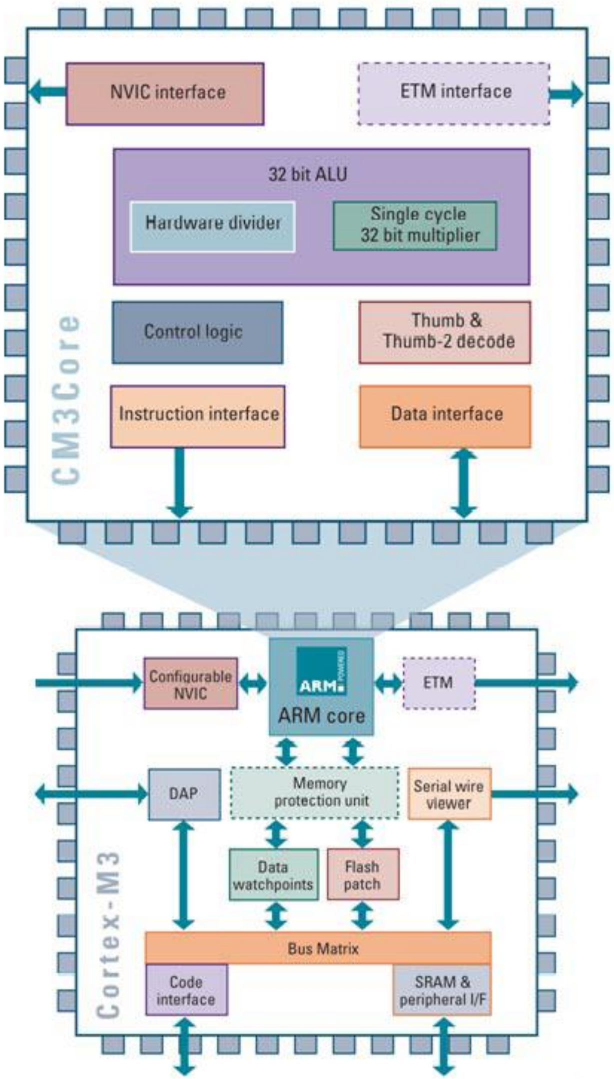
- Master: clock, start communication
- Can have more than one master
- Easy hardware, complex driver
- 100 kb/s, SPI faster

Other serial buses

- 1-wire
  - Low speed , implicit clock, unique id-code for each chip
- USB
  - Universal Serial Bus



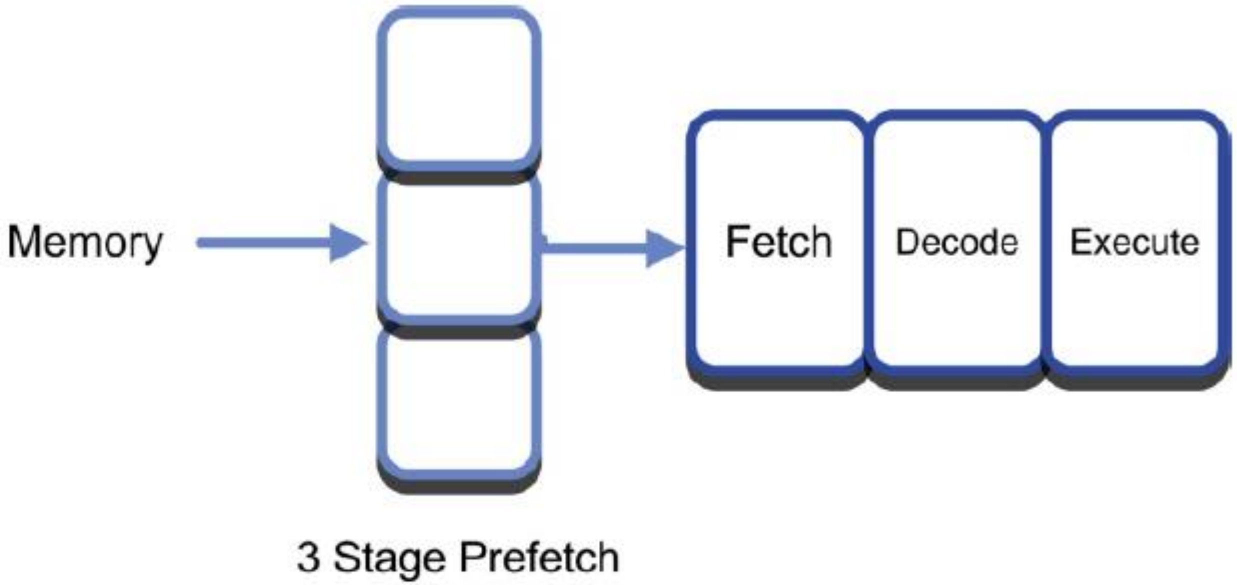
ARM STM32 Cortex-M3



The heart of the STM32 is the Cortex-M3 processor.
The Cortex M3 processor is a standardised microcontroller including 32 bit CPU, bus structure, nested interrupt unit, debug system and standard memory layout.

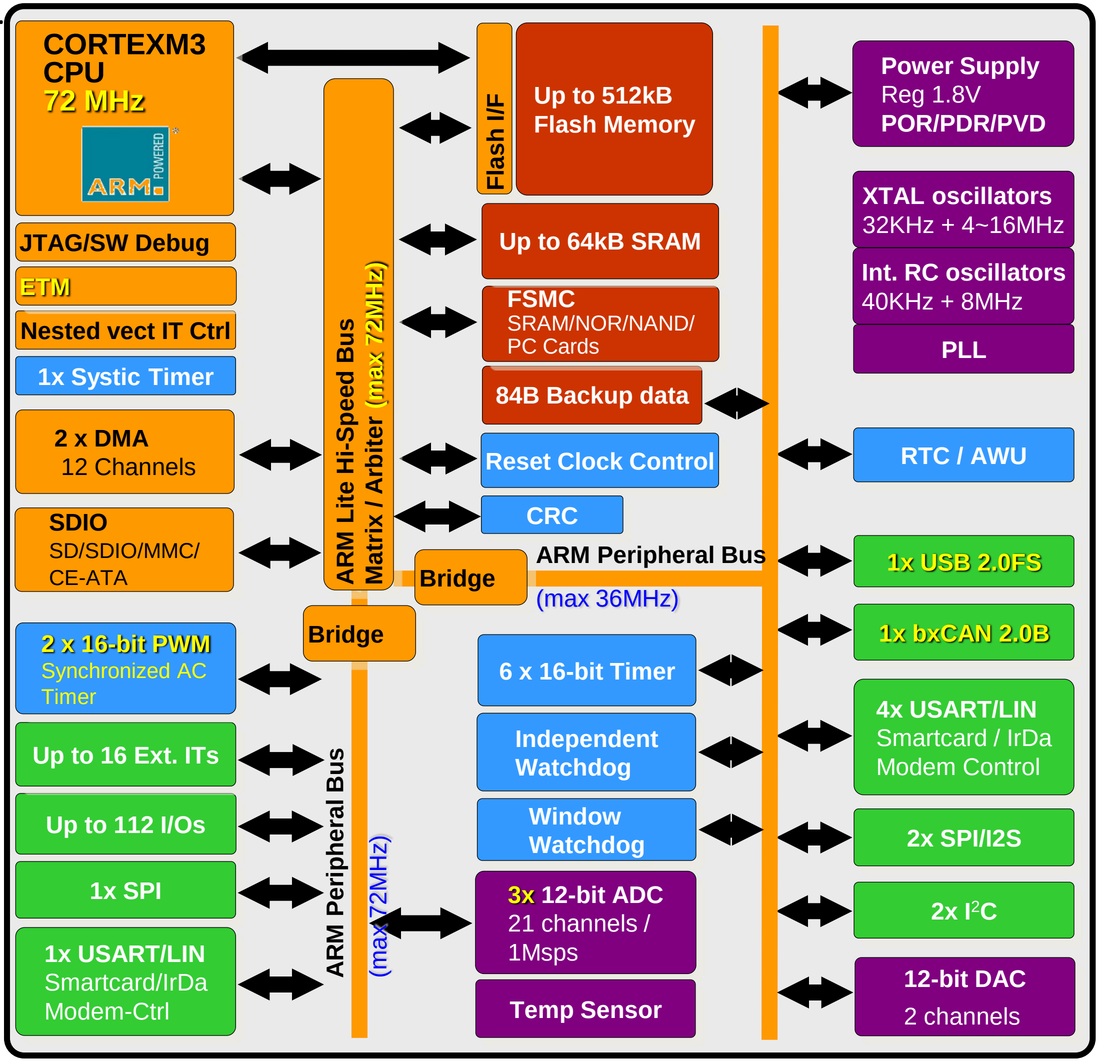
Cortex CPU Pipeline

Three stage pipeline with prefetch



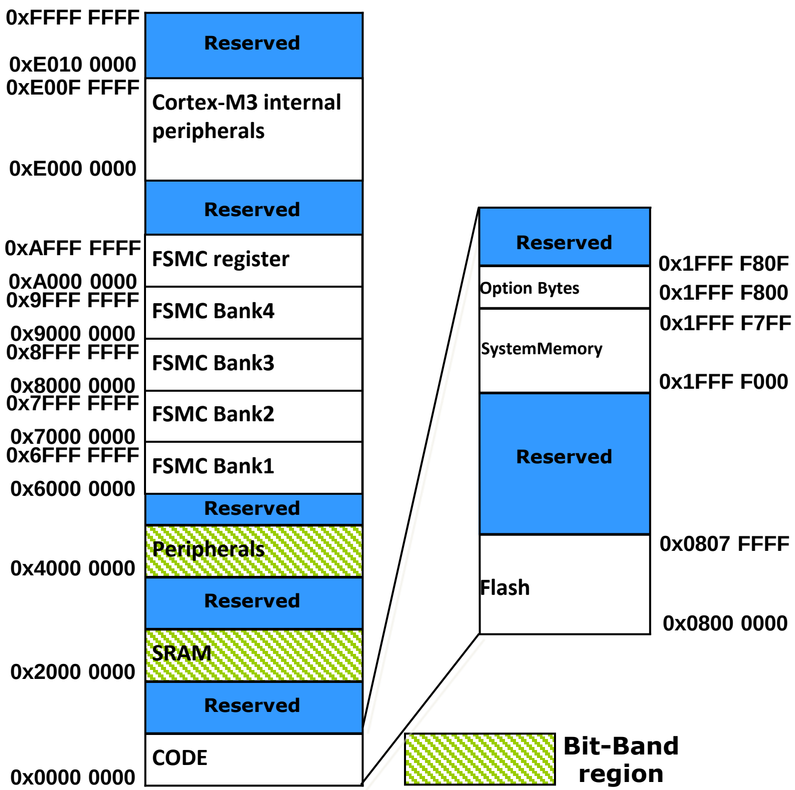
STM32F10x High-density Series Block Diagram

- **ARM 32-bit Cortex-M3 CPU**
- **Nested Vectored Interrupt Controller (NVIC) w/ 60 maskable IT + 16 prog. priority levels**
- **Embedded Memories :**
  - FLASH: up to 512kB
  - SRAM: up to 64kB
- **External memory interface FSMC:** support NAND, SRAM, NOR, PC Cards and others memory devices
- **2 x DMA w/ 12 channels**
- **SDIO:** support SD, SDIO, MMC and CE-ATA cards
- **Power Supply with internal regulator and low power modes :**
  - 2V to 3V6 supply
  - 4 Low Power Modes with Auto Wake-up
- **Integrated Power On Reset (POR)/Power Down Reset (PDR) + Programmable voltage detector (PVD)**
- **Backup domain w/ 84B user data**
- **Up to 72 MHz frequency managed & monitored by the Clock Control w/ Clock Security System**
- **Rich set of peripherals & IOs**
  - Embedded low power RTC with V_{BAT} capability
  - Dual Watchdog Architecture
  - 9 Timers w/ advanced control features (including Cortex SysTick)
  - 12 communications Interfaces
  - Up to 112 I/Os (144 pin package) w/ 16 external interrupts/event
  - Up to 3x12-bits 1Msps ADC w/ up to 21 channels and Embedded temperature sensor w/ +/-1.5° linearity with T°
  - 12-bits DAC w/ 2 channels



Memory Mapping and Boot Modes

- Addressable memory space of 4 GBytes
- RAM : up to 64 kBytes
- FLASH : up to 512 kBytes



- **Boot modes**

  Depending on the Boot configuration, Embedded Flash Memory, System Memory or Embedded SRAM Memory is aliased at @0x00

BOOT Mode Selection Pins		Boot Mode	Aliasing
BOOT1	BOOT0		
x	0	User Flash	User Flash is selected as boot space
0	1	SystemMemory	SystemMemory is selected as boot space
1	1	Embedded SRAM	Embedded SRAM is selected as boot space

- **SystemMemory**: contains the Bootloader used to re-program the FLASH through USART1.

For more details refer to AN2606 & UM0462

A PC Windows Demonstrator is available as well.

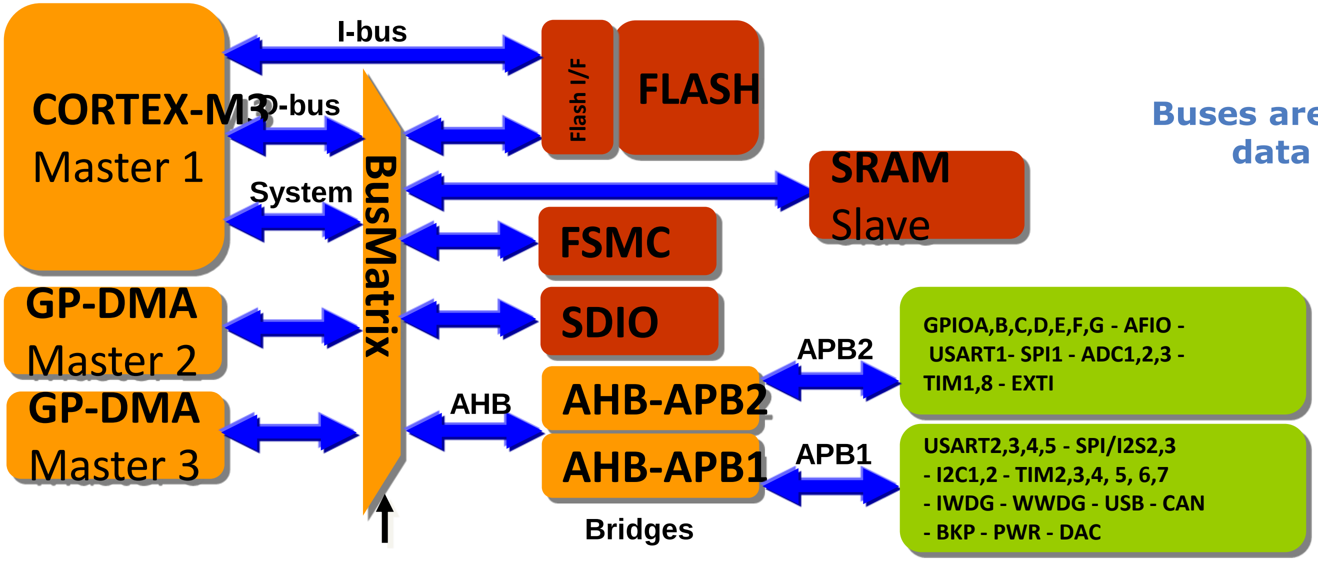
- **Boot from Embedded SRAM :**

  In the application initialization code you have to Relocate the Vector Table in SRAM using the NVIC Exception Table and Offset register

System Architecture

- **Multiply possibilities of bus accesses to SRAM, Flash, Peripherals, DMA**
  - BusMatrix added to Harvard architecture allows parallel access
- **Efficient DMA and Rapid data flow**
  - Direct path to SRAM through arbiter, guarantees alternating access
  - Harvard architecture + BusMatrix allows Flash execution in parallel with DMA transfer
- **Increase Peripherals Speed for better performance**
  - Dual Advanced Peripheral buses (APB) architecture w/ High Speed APB (APB2) up to 72MHz and Low Speed APB (APB1) up to 36MHz

➔ Allows to optimize use of peripherals (18MHz SPI, 4.5Mbps USART, 72MHz PWM Timer, 18MHz toggling I/Os)



Buses are not overloaded with data movement tasks

System Timer

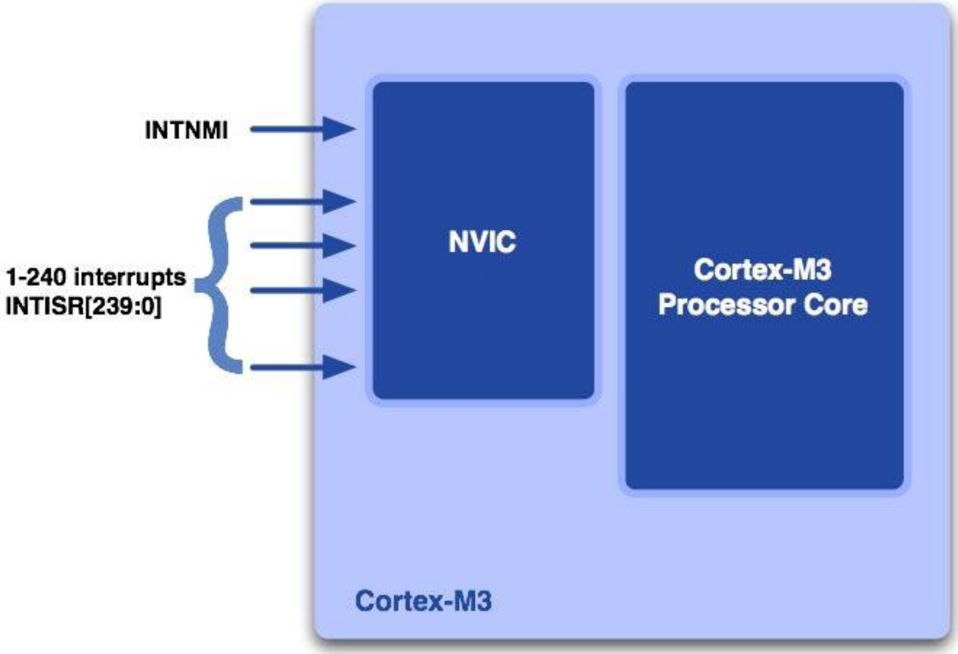
The SysTick Timer is a 24-bit auto-reload timer located within the Cortex-M3 processor.

24-bit down counter, with auto reload and end of count interrupt
Generate periodic interrupt for scheduled tasks

- Flexible system timer
- 24-bit self-reloading down counter with end of count interrupt generation
- 2 configurable Clock sources
- Suitable for Real Time OS or other scheduled tasks

In STM32F10x the SysTick clock can be: CPU clock or CPU clock/8
(provided externally by the Reset Clock Control)

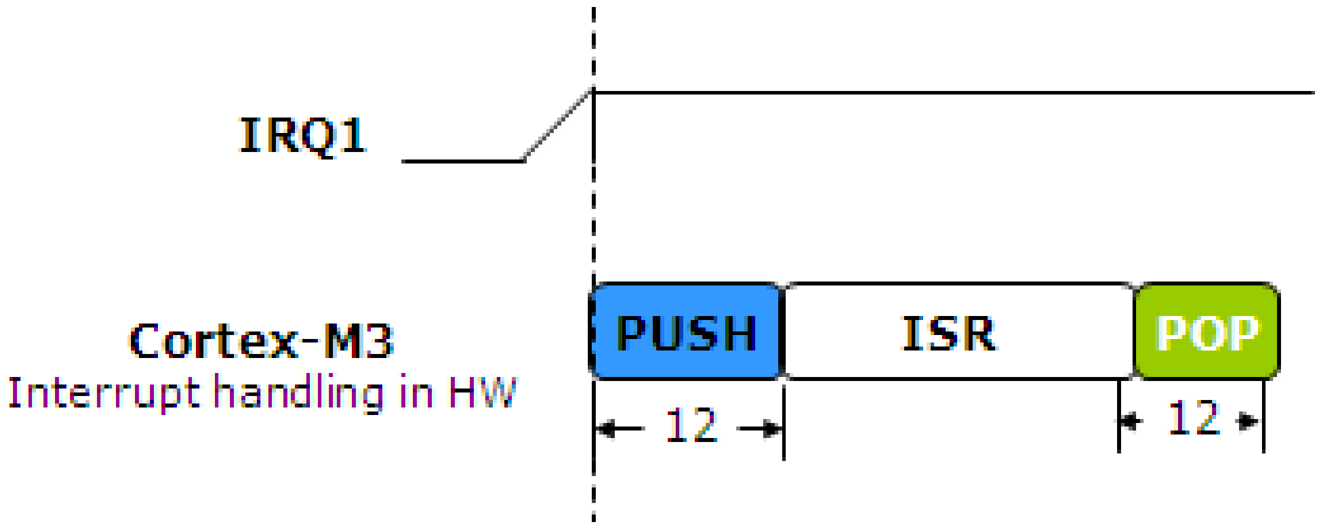
NVIC - Nested Vector Interrupt Controller



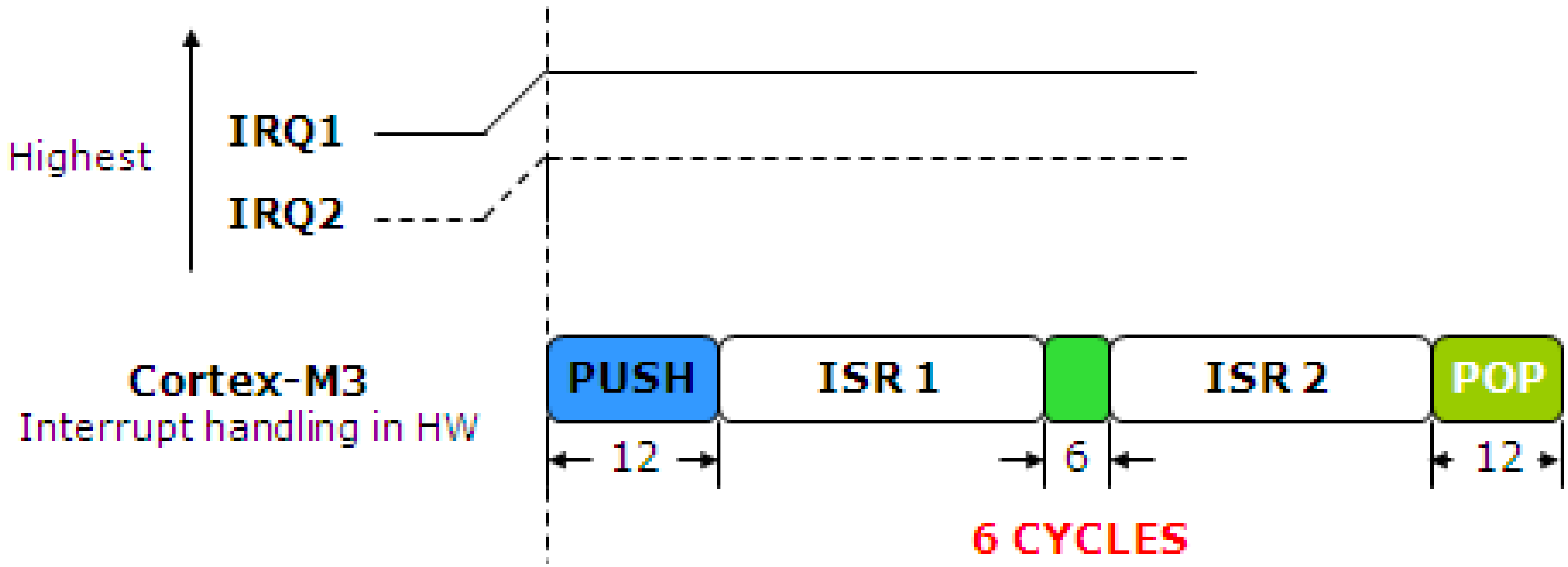
The Nested Vector Interrupt Controller is a standard unit within the Cortex core. This means that all Cortex-based microcontrollers will have the same interrupt structure, regardless of manufacturer.

NVIC

The NVIC will respond to an interrupt with a latency of twelve cycles. This includes a microcoded routine to automatically push a set of registers onto the stack.

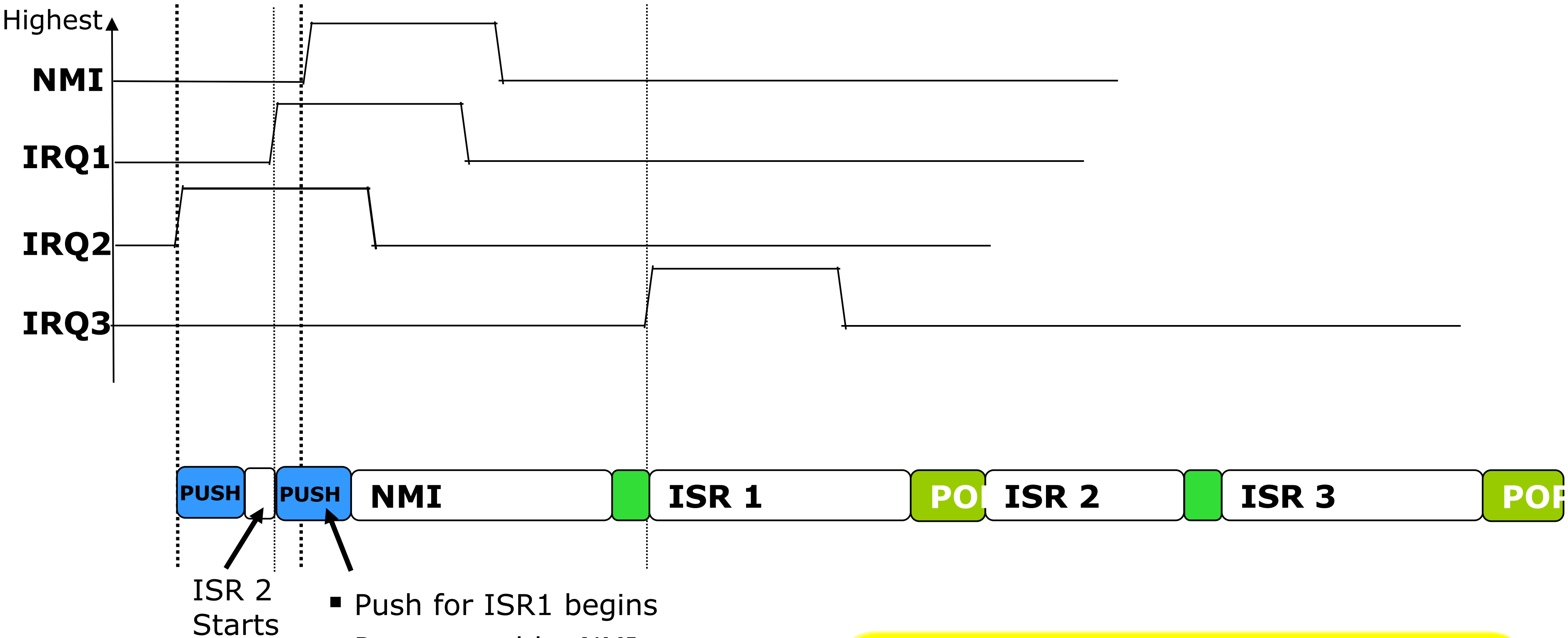


Tail-chaining



Multiple interrupts will be 'tail chained' so there is a minimum delay of six cycles from the end of one interrupt to the start of the next.

Interrupt response - Example



- Push for ISR1 begins
- Pre-empted by NMI
- New instruction fetch in parallel minimises time to NMI

Cortex-M3

- Following NMI processor tail-chains into ISR1
- ISR2 Completed
- Pop only occurs on return to "Main"

Exception Vector Types and Priority

No.	Exception Type	Priority	Type of Priority	Descriptions
1	Reset	-3 (Highest)	fixed	Reset
2	NMI	-2	fixed	Non-Maskable Interrupt
3	Hard Fault	-1	fixed	Default fault if other hander not implemented
4	MemManage Fault	0	settable	MPU violation or access to illegal locations
5	Bus Fault	1	settable	Fault if AHB interface receives error
6	Usage Fault	2	settable	Exceptions due to program errors
7-10	Reserved	N.A.	N.A.	
11	SVCall	3	settable	System Service call
12	Debug Monitor	4	settable	Break points, watch points, external debug
13	Reserved	N.A.	N.A.	
14	PendSV	5	settable	Pendable request for System Device
15	SYSTICK	6	settable	System Tick Timer
16	Interrupt #0	7	settable	External Interrupt #0
......	....	....	settable	....
256	Interrupt#240	247	settable	External Interrupt #240

The Cortex exception table contains the start address or an ISR which is loaded into the Program counter as the CPU enters the exception.

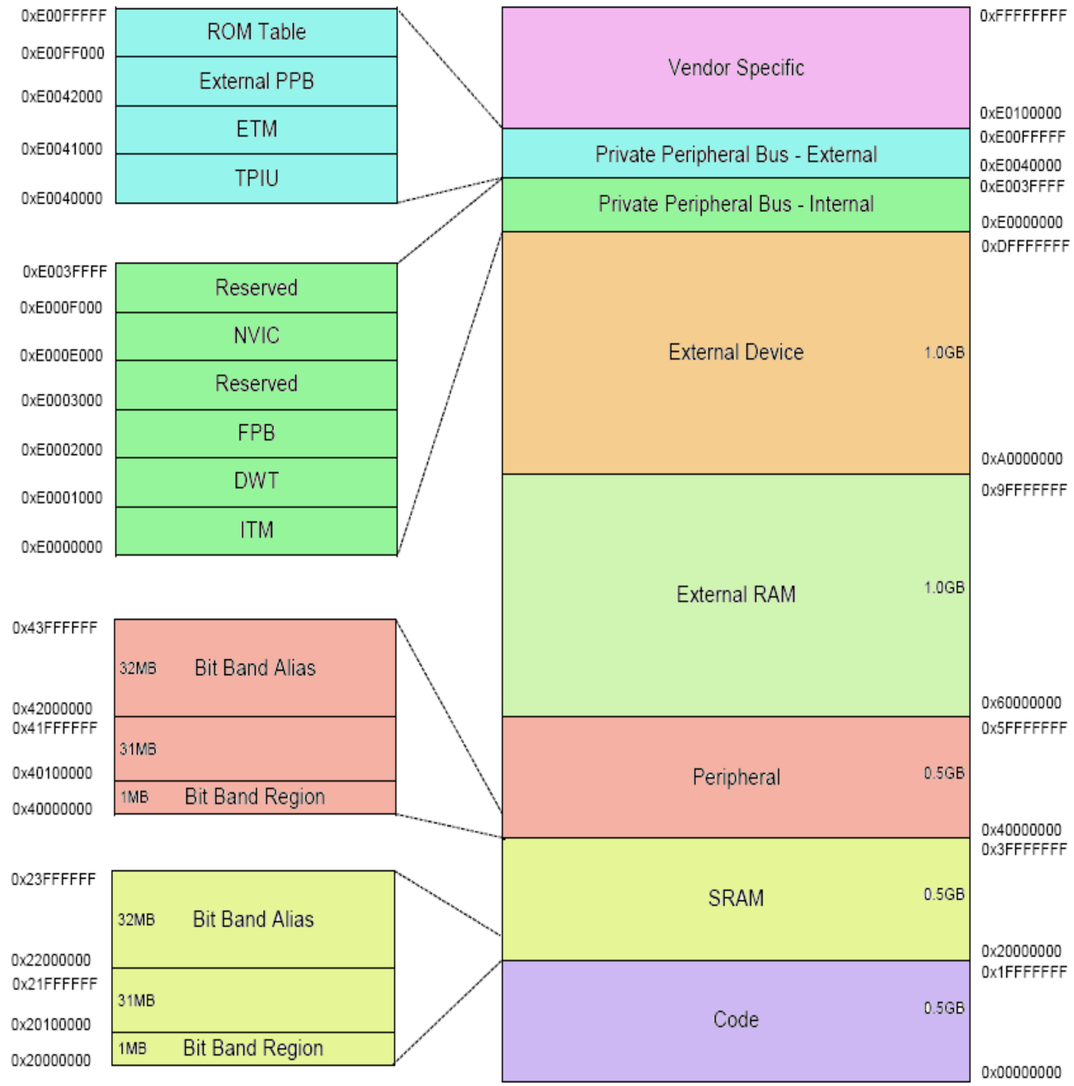
Vector Table

- Vector Table starts at location 0
  - In the code section of the memory map
- Vector Table contains addresses (vectors) of exception handlers and ISRs
  - Not instructions like other ARM processors
- Table size (in words) is = number of IRQ inputs + 16
  - Minimum size (case of 1 IRQ) : 17 words
  - Maximum size (case of 240 IRQs) 256 words
- Main stack pointer initial value in location 0
  - Set up by hardware during Reset
- Vector Table can be relocated (to SRAM)
  - Software configurable through dedicated register in SCB

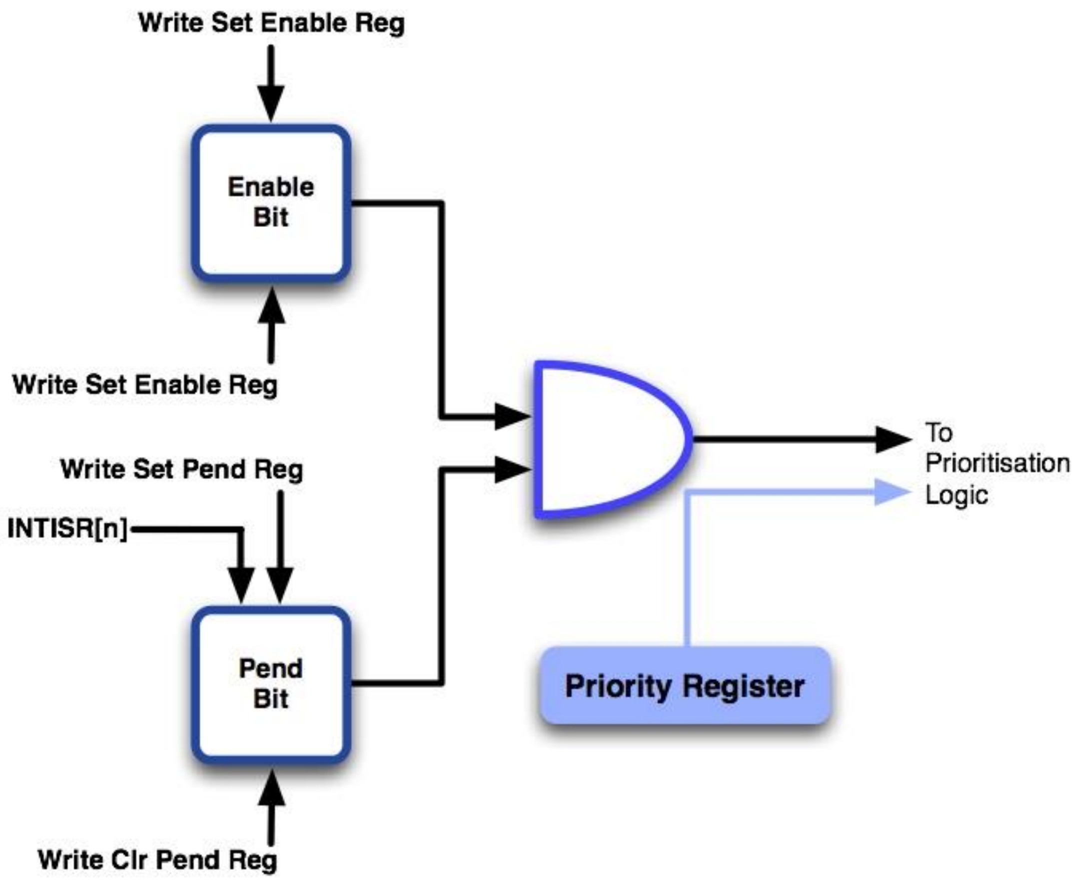
Address	Vector
0x00	Initial Main SP
0x04	Reset
0x08	NMI
0x0C	Hard Fault
0x10	Memory Manage
0x14	Bus Fault
0x18	Usage Fault
0x1C-0x28	Reserved
0x2C	SVCall
0x30	Debug Monitor
0x34	Reserved
0x38	PendSV
0x3C	Systick
40	IRQ0
...	More IRQs

Cortex-M3 Memory Map

- **Vendor Specific (0.5GB)**
  - **Set aside to enable vendors to implement peripheral compatibility with previous systems**
- **Private Peripheral Bus (1M)**
  - **Address space for system components (CoreSight, NVIC etc.)**
- **External Device (1GB).**
  - **Intended for external devices and/or shared memory that needs ordering/non-buffered**
- **External RAM (1GB)**
  - **Intended for off chip memory**
- **Peripheral (0.5G)**
  - **Intended for normal peripherals. The bottom 1MB of the 32MB peripheral address space (0x40000000 – 0x400FFFFF) is reserved for bit-band accesses. Accesses to the peripheral 32MB bit band alias region (0x42000000 – 0x43FFFFFF) are remapped to this 1MB**
- **SRAM (0.5GB)**
  - **Intended for on-chip SRAM. The bottom 1MB of the SRAM address space (0x20000000 - 0x200FFFFF) is reserved for bit-band accesses. Accesses to the SRAM 32MB bit band alias region (0x22000000 – 0x23FFFFFF) are remapped to this 1MB address space.**
- **Code(0.5GB)**
  - **Reserved for code memory (flash, SRAM). This region is accessed via the Cortex-M3 ICode and DCode busses.**



Interrupt enable bits



Each interrupt source has an enable bit in the NVIC and in the peripheral. In the STM32 there are sixteen levels of priority.

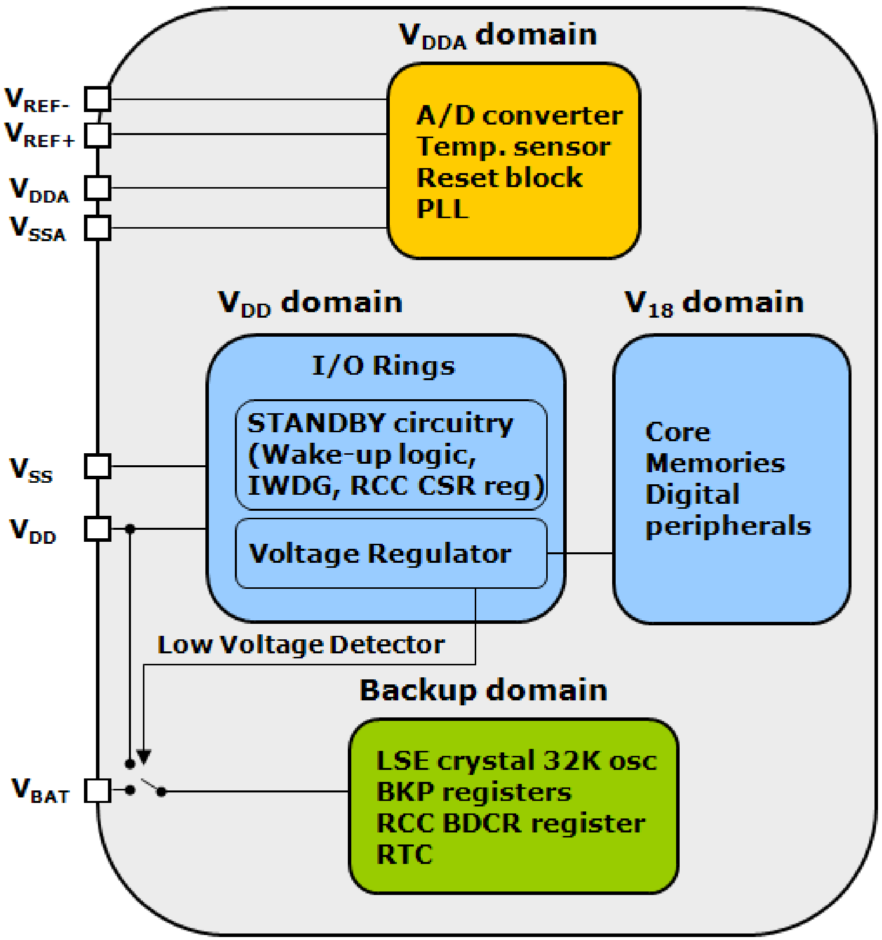
NVIC* Configuration And Use

To use the NVIC we need to do three things.

1. Configure the vector table for the interrupt sources we want to use
2. Configure the NVIC registers to enable and set the priorities of the NVIC interrupts
3. Configure the peripheral and enable its interrupt support.

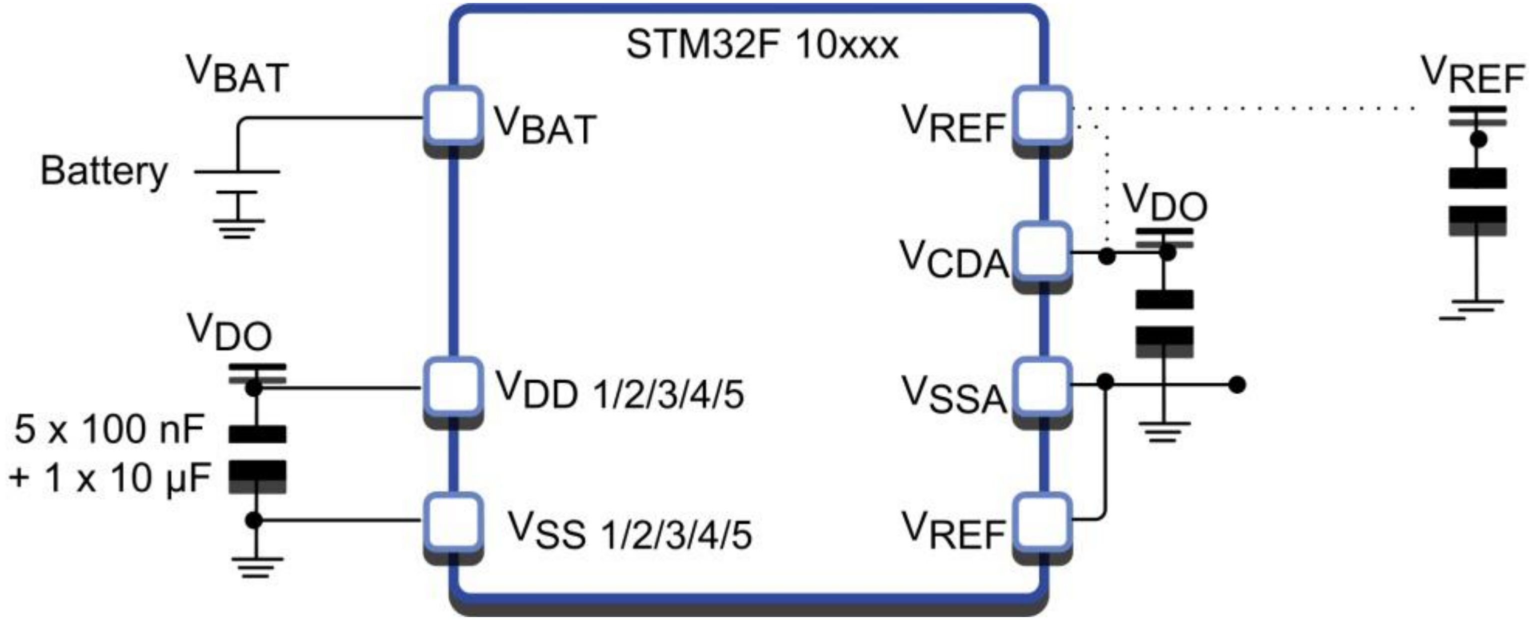
*NVIC = Nested Vector Interupt Controller

Power supply



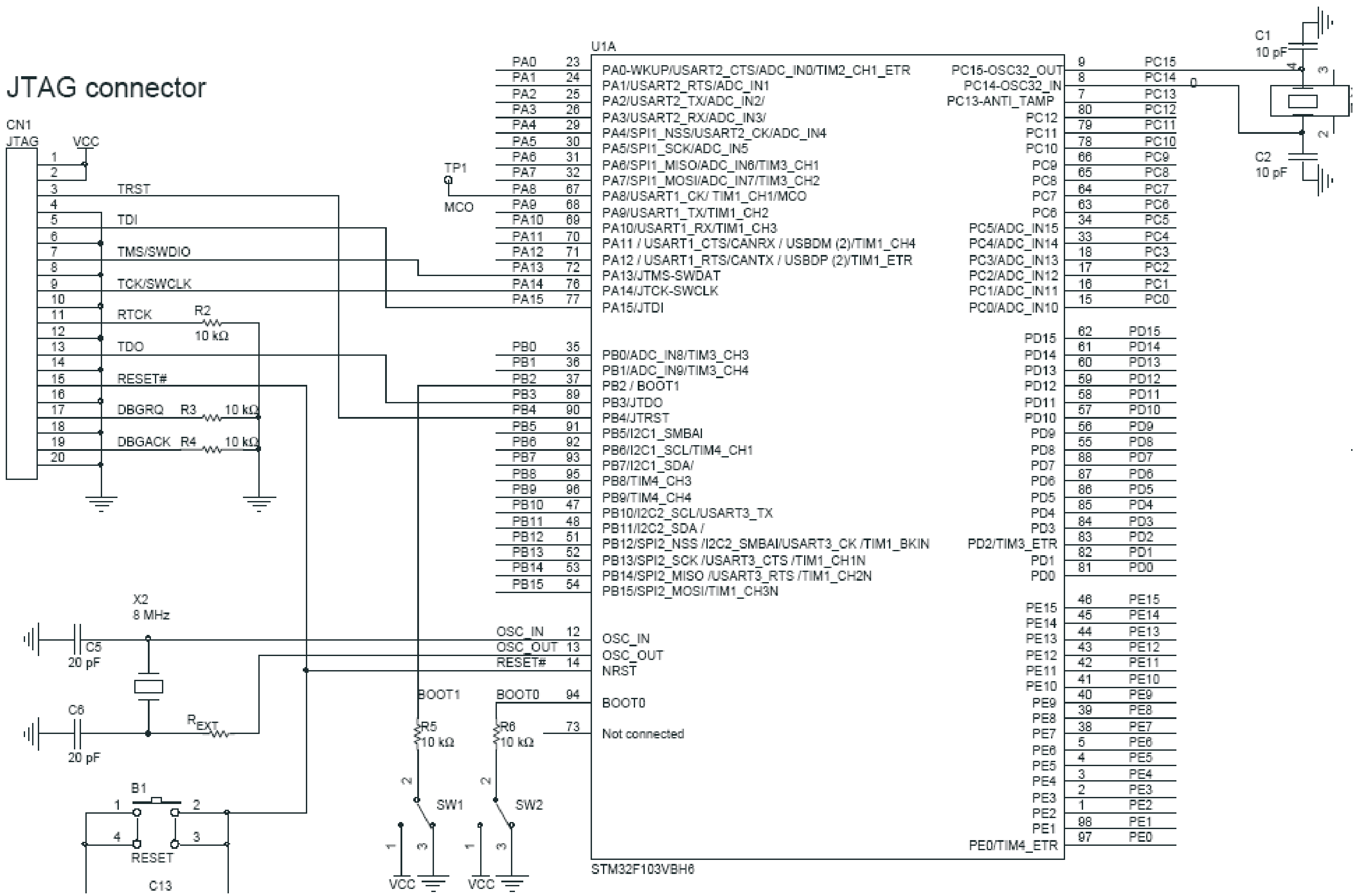
The STM32 runs from a single 2.0V-3.6V supply. There is an additional backup power domain and a separate supply for the ADC converter (144 pin package only).

Supply stabilisation capacitors



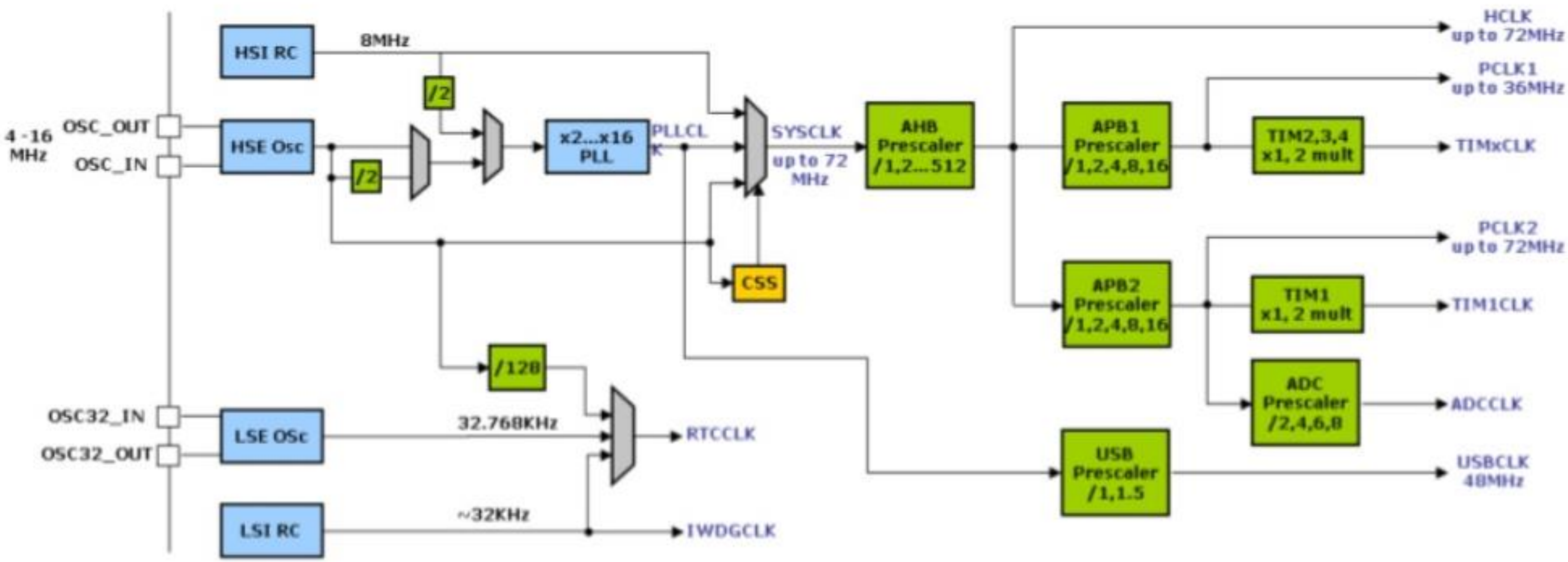
With an internal reset and an internal voltage regulator, the STM32 only needs seven external capacitors.

Basic Hardware Schematic



Clock Schemes

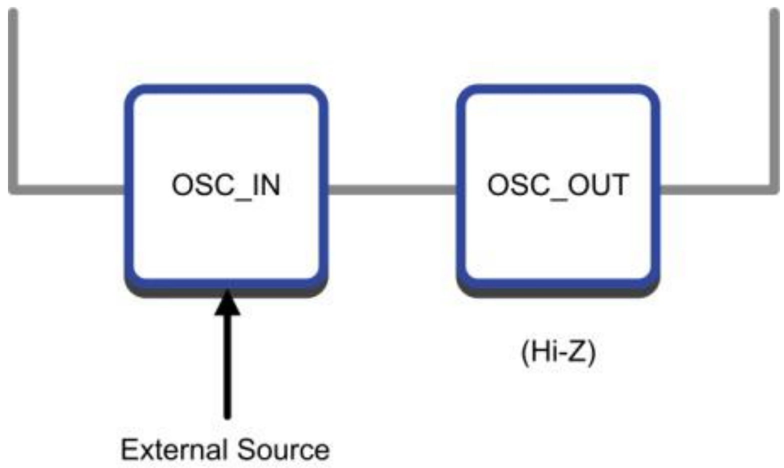
2 external clocks
2 internal clocks
1 PLL

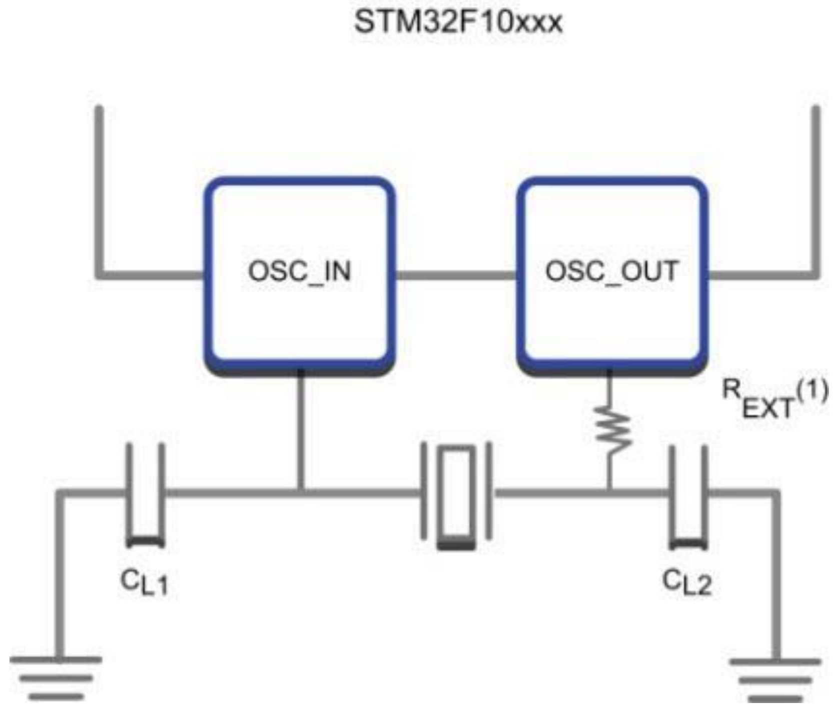


Internal Oscillator

The STM32 has internal RC oscillators which are capable of supplying a clock to the internal PLL. This will allow the microcontroller to run at its maximum 72 MHz clock frequency. The internal oscillators are not as accurate or stable as an external crystal; consequently for most designs you will need at least one clock source.

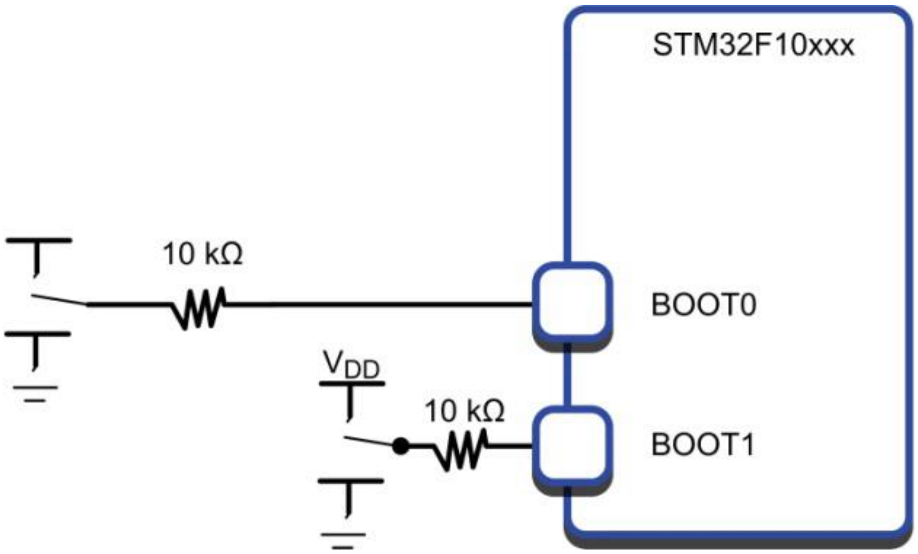
External Oscillator





The External Oscillator can be run from a crystal or external clock source.

Boot Modes



The external boot pins are used to select which region of memory aliased to the first 2k of memory. This can be user Flash, the internal bootloader or the first 2k of SRAM