STMicroelectronics - ST72F321AR9TA Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321ar9ta

6.6 System integrity management (SI)

The System Integrity Management block contains the Low Voltage Detector (LVD) and Auxiliary Voltage Detector (AVD) functions. It is managed by the SICSR register.

6.6.1 Low voltage detector (LVD)

The low voltage detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a V_{IT-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD reset circuitry generates a reset when V_{DD} is below:

- V_{IT+} when V_{DD} is rising
- V_{IT-} when V_{DD} is falling

The LVD function is illustrated in *Figure 14*.

The voltage threshold can be configured by option byte to be low, medium or high.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above V_{IT-}, the MCU can only be in two modes:

- under full software control
- in static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

During a low voltage detector reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

Note: *The LVD allows the device to be used without any external RESET circuitry.*

If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range. Below 3.8V, device operation is not guaranteed.

The LVD is an optional function which can be selected by option byte.

It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from Reset, to ensure the application functions properly.

6.6.5 System Integrity (SI) Control/Status register (SICSR)

SICSR Reset value: 000x 000x (00h)

7	6	5	4	3	2	1	0
AVDS	AVDIE	AVDF	LVDRF	Reserved			WDGRF
RW	RW	RW	RW	-			RW

Table 13. **SICSR description**

Bit	Name	Function
7	AVDS	*Voltage Detection selection* This bit is set and cleared by software. Voltage Detection is available only if the LVD is enabled by option byte. 0: Voltage detection on V_{DD} supply 1: Voltage detection on EVD pin
6	AVDIE	*Voltage Detector interrupt enable* This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag changes (toggles). The pending interrupt information is automatically cleared when software enters the AVD interrupt routine. 0: AVD interrupt disabled 1: AVD interrupt enabled
5	AVDF	*Voltage Detector flag* This read-only bit is set and cleared by hardware. If the AVDIE bit is set, an interrupt request is generated when the AVDF bit changes value. Refer to *Figure 15* and to *Monitoring the VDD main supply on page 44* for additional details. 0: V_{DD} or V_{EVD} over $V_{IT+(AVD)}$ threshold 1: V_{DD} or V_{EVD} under $V_{IT-(AVD)}$ threshold
4	LVDRF	*LVD reset flag* This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (writing zero). See *Table 14: Reset source flags* for more details. When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.
3:1	-	Reserved, must be kept cleared.
0	WDGRF	*Watchdog reset flag* This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (writing zero) or an LVD Reset (to ensure a stable cleared state of the WDGRF flag when CPU starts). Combined with the LVDRF flag information, the flag description is given in *Table 14*.

Table 14. **Reset source flags**

Reset sources	LVDRF	WDGRF
External $\overline{\text{RESET}}$ pin	0	0
Watchdog	0	1
LVD	1	X

The TLI, RESET, and TRAP vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

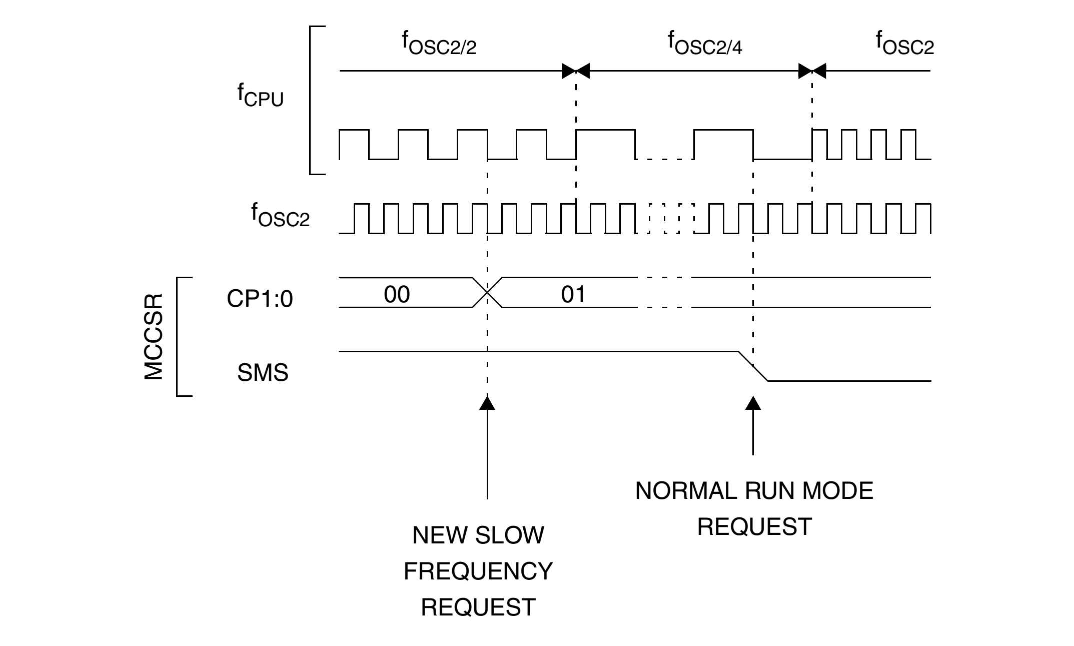
Caution: If the I1_x and I0_x bits are modified while the interrupt x is executed the following behavior has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

Table 19. Interrupt dedicated instruction set

Instruction	New description	Function/Example	I1	H	I0	N	Z	C
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC	I1	H	I0	N	Z	C
JRM	Jump if I1:0 = 11 (level 3)	I1:0 = 11 ?						
JRNM	Jump if I1:0 <> 11	I1:0 <> 11 ?						
POP CC	Pop CC from the Stack	Mem => CC	I1	H	I0	N	Z	C
RIM	Enable interrupt (level 0 set)	Load 10 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load 11 in I1:0 of CC	1		1			
TRAP	Software trap	Software NMI	1		1			
WFI	Wait for interrupt		1		0			

Note: During the execution of an interrupt routine, the HALT, POPCC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

Figure 23. Slow mode clock transitions



8.3 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the ‘WFI’ instruction.

All peripherals remain active. During Wait mode, the I[1:0] bits of the CC register are forced to ‘10’, to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in Wait mode until an interrupt or RESET occurs, whereupon the Program Counter branches to the starting address of the interrupt or Reset service routine.

The MCU will remain in Wait mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to the following *Figure 24*.

10.9 Register description

10.9.1 Control register (WDGCR)

WDGCR Reset value: 0111 1111 (7Fh)

7	6	5	4	3	2	1	0
WDGA	T[6:0]						
RW	RW						

Table 36. WDGCR register description

Bit	Name	Function
7	WDGA	*Activation bit* This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset. 0: Watchdog disabled 1: Watchdog enabled *Note: This bit is not used if the hardware watchdog option is enabled by option byte.*
6:0	T[6:0]	*7-bit counter (MSB to LSB)* These bits contain the value of the watchdog counter. It is decremented every 16384 f_{OSC2} cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

Table 37. Watchdog timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
002Ah	WDGCR Reset Value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

12 PWM auto-reload timer (ART)

12.1 Introduction

The Pulse Width Modulated Auto-Reload Timer on-chip peripheral consists of an 8-bit auto-reload counter with compare/capture capabilities and of a 7-bit prescaler clock source.

These resources allow five possible operating modes:

- Generation of up to 4 independent PWM signals
- Output compare and Time base interrupt
- Up to 2 input capture functions
- External event detector
- Up to 2 external interrupt sources

The three first modes can be used together with a single counter frequency.

The timer can be used to wake up the MCU from Wait and Halt modes.

Figure 35. PWM auto-reload timer block diagram

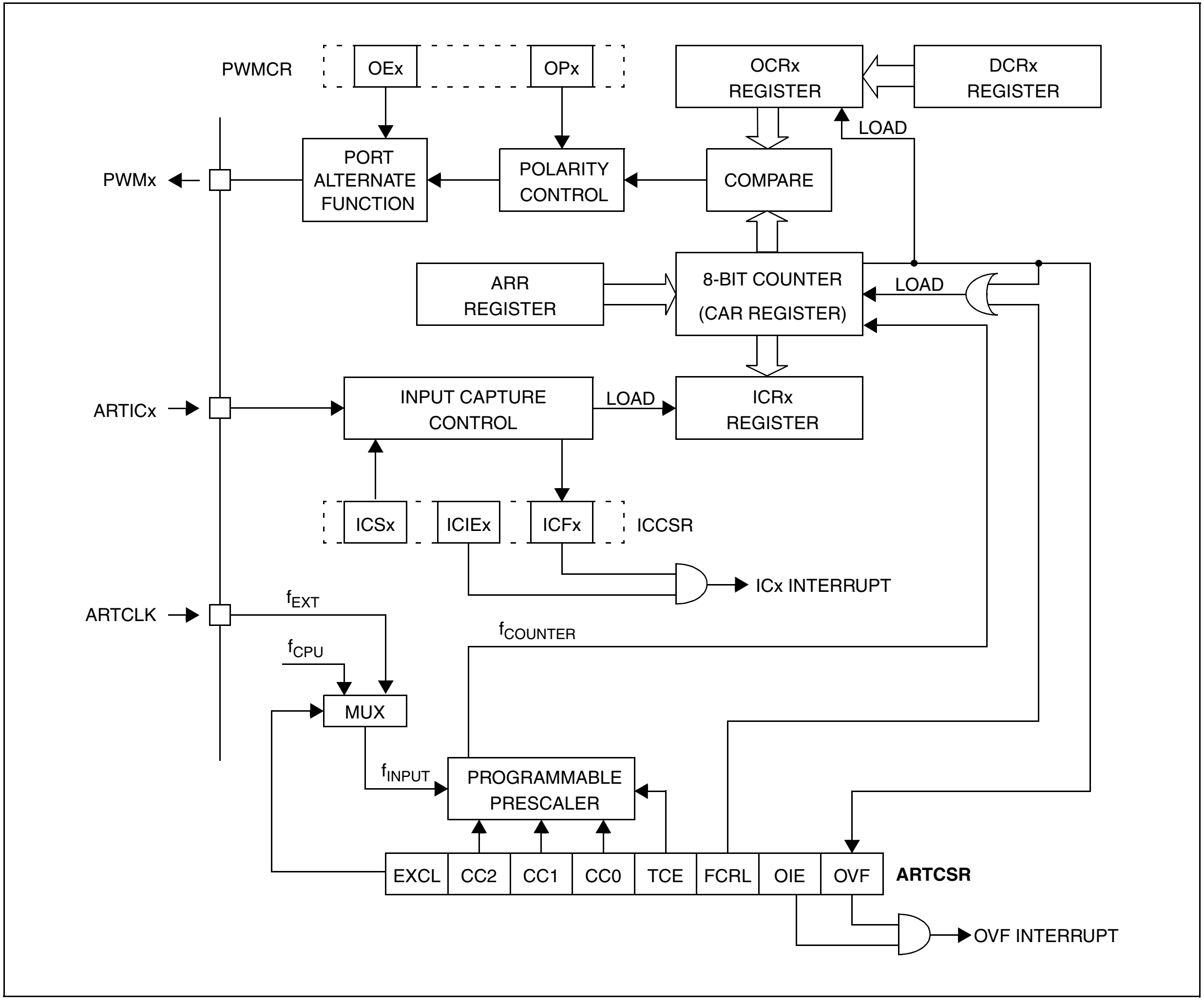
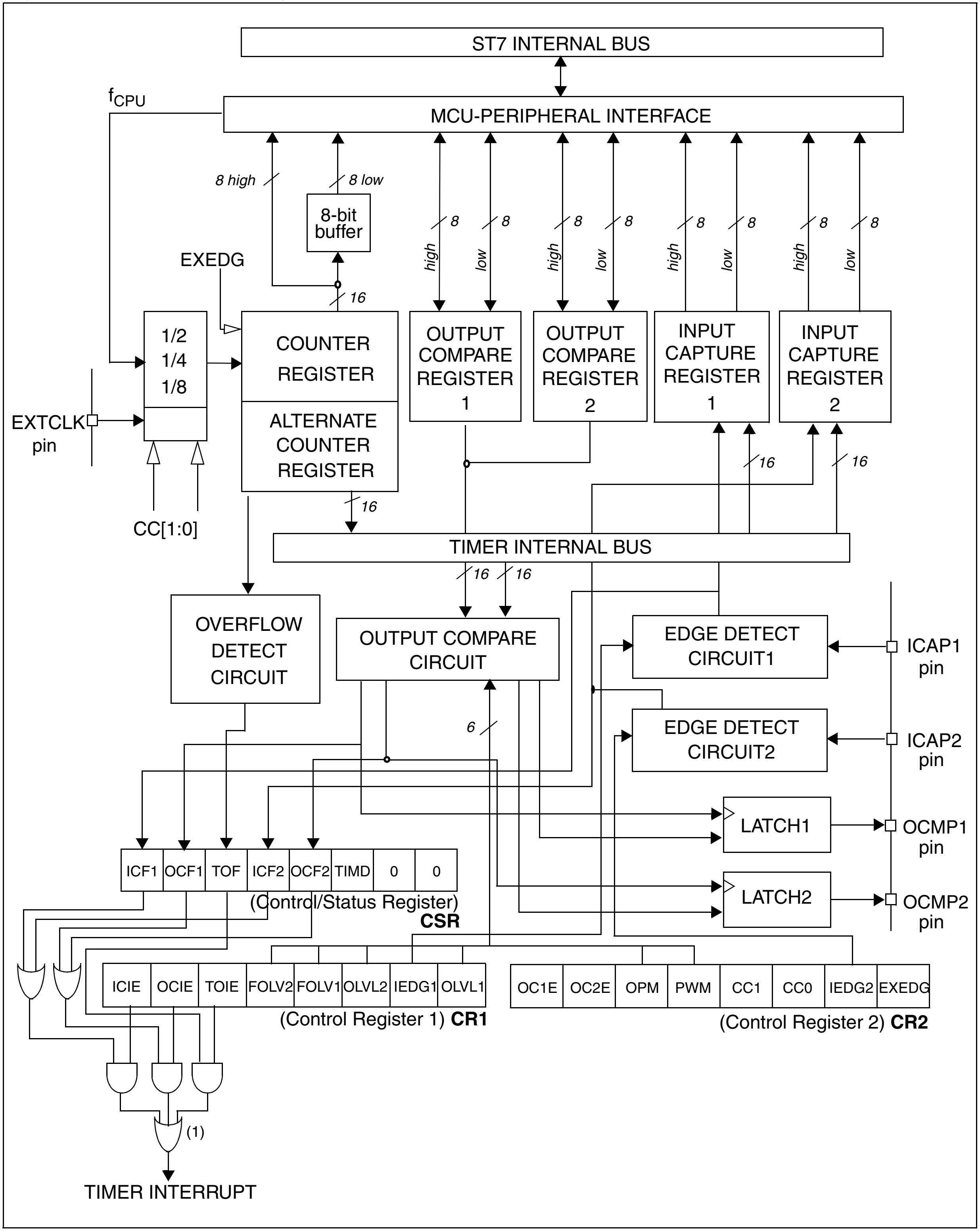


Figure 41. Timer block diagram



1. If IC, OC and TO interrupt request have separate vectors, then the last OR is not present (see device interrupt vector table).

13.7.13 Alternate counter low register (ACLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

ACLR Reset value: 1111 1100 (FCh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

13.7.14 Input capture 2 high register (IC2HR)

This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).

IC2HR Reset value: Undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

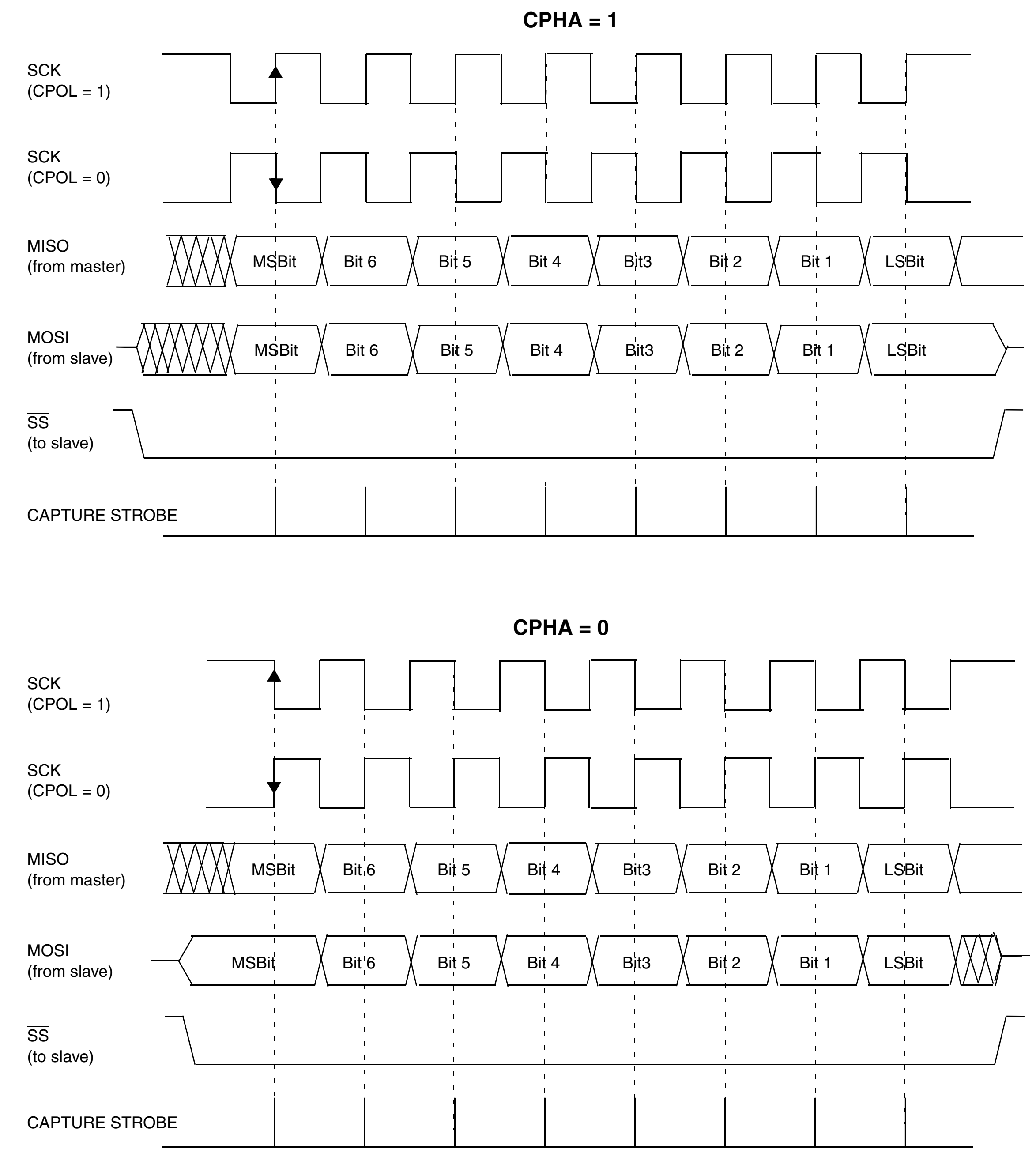
13.7.15 Input capture 2 low register (IC2LR)

This is an 8-bit read only register that contains the low part of the counter value (transferred by the Input Capture 2 event).

IC2LR Reset value: Undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Figure 59. Data clock timing diagram



Note: This figure should not be used as a replacement for parametric information. Refer to Section 19: Electrical characteristics.

Table 80. SCI register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0050h	SCISR Reset value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
0051h	SCIDR Reset value	MSB x	x	x	x	x	x	x	LSB x
0052h	SCIBRR Reset value	SCP1 0	SCP0 0	SCT2 0	SCT1 0	SCT0 0	SCR2 0	SCR1 0	SCR0 0
0053h	SCICR1 Reset value	R8 x	T8 0	SCID 0	M 0	WAKE 0	PCE 0	PS 0	PIE 0
0054h	SCICR2 Reset value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0
0055h	SCIERPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	SCIPETPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0

16.3.3 SDA/SCL line control

Transmitter mode

The interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the data register.

Receiver mode

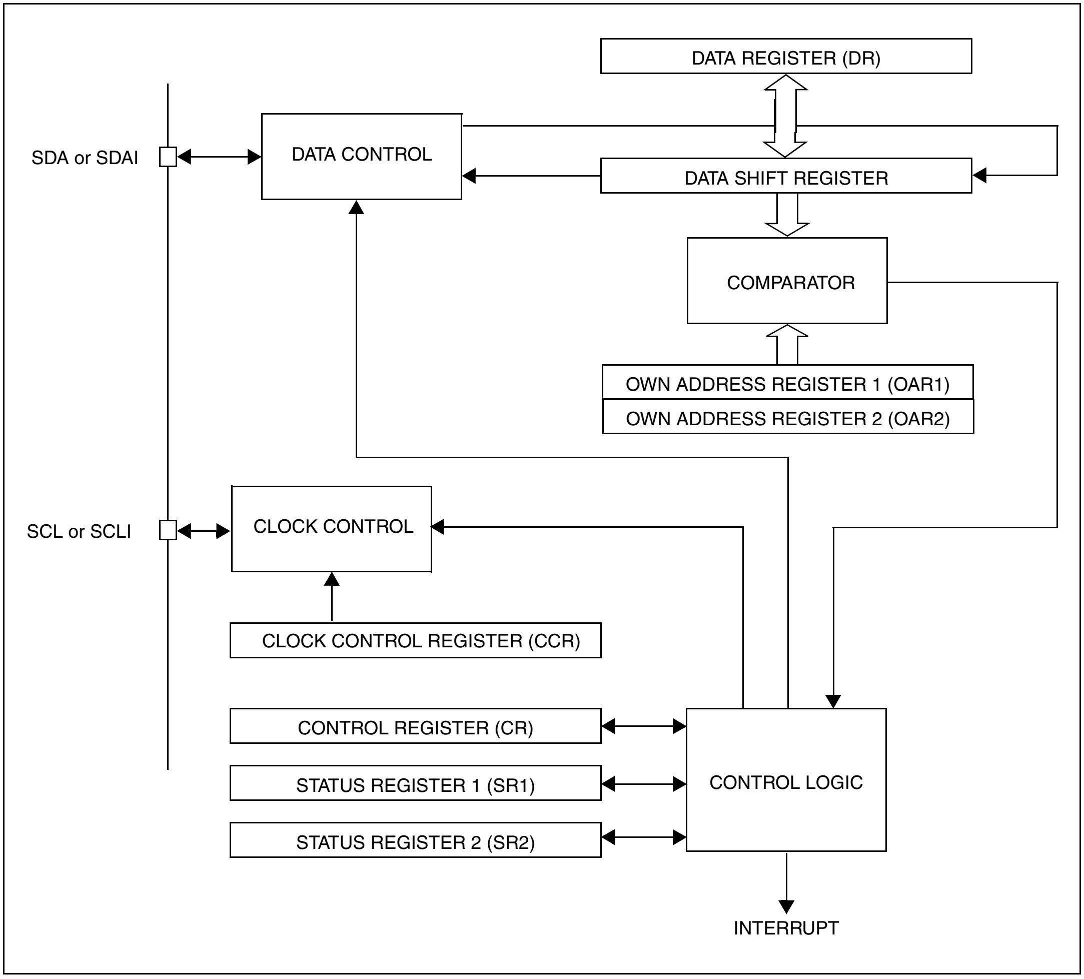
The interface holds the clock line low after reception to wait for the microcontroller to read the byte in the data register.

The SCL frequency (f_{SCL}) is controlled by a programmable clock divider which depends on the I^2C bus mode.

When the I^2C cell is enabled, the SDA and SCL ports must be configured as floating inputs. In this case, the value of the external pull-up resistor used depends on the application.

When the I^2C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 67. I^2C interface block diagram



16.7 Register description

16.7.1 I^{2}C control register (CR)

CR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved		PE	ENGC	START	ACK	STOP	ITE
-		RW	RW	RW	RW	RW	RW

Table 83. CR register description

Bit	Name	Function
7:6	-	Reserved. Forced to 0 by hardware.
5	PE	*Peripheral enable* This bit is set and cleared by software. 0: Peripheral disabled 1: Master/Slave capability *Notes:* *- When PE = 0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released while PE = 0* *- When PE = 1, the corresponding I/O pins are selected by hardware as alternate functions.* *To enable the I^{2}C interface, write the CR register* ***TWICE*** *with PE = 1 as the first write only activates the interface (only PE is set).*
4	ENGC	*Enable General Call* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0). The 00h General Call address is acknowledged (01h ignored). 0: General Call disabled 1: General Call enabled *Note: In accordance with the I2C standard, when GCAL addressing is enabled, an I2C slave can only receive data. It will not transmit data to the master.*
3	START	*Generation of a Start condition* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0) or when the Start condition is sent (with interrupt generation if ITE = 1). **In Master mode** 0: No start generation 1: Repeated start generation **In Slave mode** 0: No start generation 1: Start generation when the bus is free
2	ACK	*Acknowledge enable* This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE = 0). 0: No acknowledge returned 1: Acknowledge returned after an address byte or a data byte is received

17.3 Functional description

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than V_{AREF} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in *Section 19: Electrical characteristics*.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the allotted time.

17.3.1 A/D converter configuration

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the *Chapter 9: I/O ports*. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register:

- Select the CS[3:0] bits to assign the analog channel to convert.

17.3.2 Starting the conversion

In the ADCCSR register:

- Set the ADON bit to enable the A/D converter and to start the conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

- The EOC bit is set by hardware.
- The result is in the ADCDR registers.

A read to the ADCDRH or a write to any bit of the ADCCSR register resets the EOC bit.

To read the 10 bits, perform the following steps:

1. Poll the EOC bit.
2. Read the ADCDRL register.
3. Read the ADCDRH register. This clears EOC automatically.

Note: The data is not latched, so both the low and the high data register must be read before the next conversion is complete, so it is recommended to disable interrupts while reading the conversion result.

To read only 8 bits, perform the following steps:

1. Poll the EOC bit.
2. Read the ADCDRH register. This clears EOC automatically.

17.3.3 Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[3:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

17.4 Low power modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

Table 91. Effect of low power modes on ADC

Mode	Effect
Wait	No effect on A/D converter
Halt	A/D converter disabled. After wake-up from Halt mode, the A/D converter requires a stabilization time t_{STAB} (see *Section 19: Electrical characteristics*) before accurate conversions can be performed.

17.5 Interrupts

None.

17.6 ADC registers

17.6.1 Control/status register (ADCCSR)

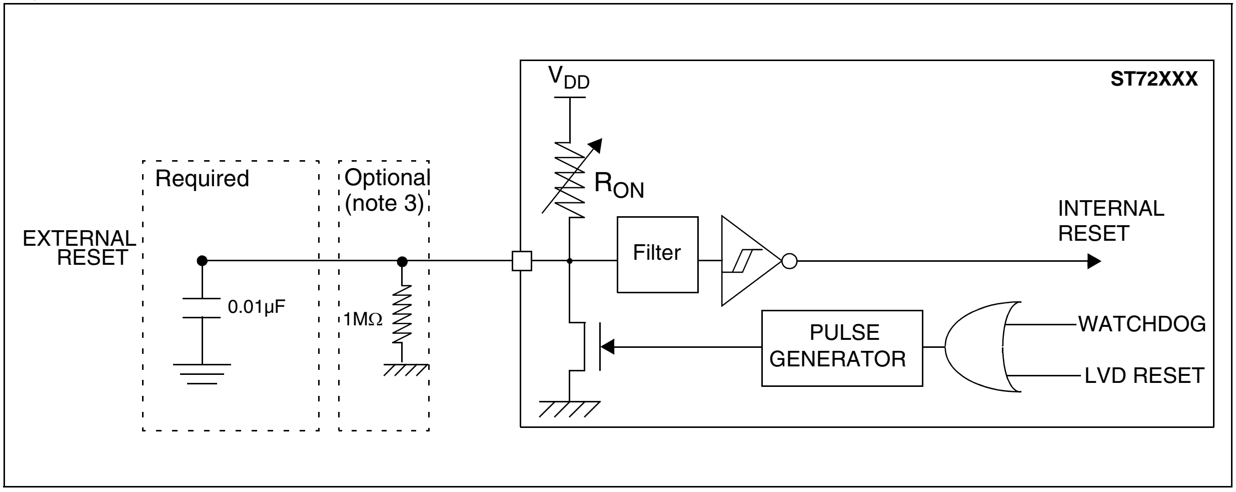
ADCCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
EOC	SPEED	ADON	Reserved	CH[3:0]			
RO	RW	RW	-	RW			

Table 92. ADCCSR register description

Bit	Name	Function
7	EOC	*End of Conversion* This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register. 0: Conversion is not complete 1: Conversion complete
6	SPEED	*ADC clock selection* This bit is set and cleared by software. 0: $f_{ADC} = f_{CPU}/4$ 1: $f_{ADC} = f_{CPU}/2$

Figure 90. $\overline{\text{RESET}}$ pin protection when LVD is enabled



Note: 1 *The reset network protects the device against parasitic resets.*

The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).

Whether the reset source is internal or external, the user must ensure that the level on the $\overline{\text{RESET}}$ *pin can go below the* V_{IL} *maximum level specified in Section 19.9.1 on page 207. Otherwise the reset will not be taken into account internally.*

Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ *pin, the user must ensure that the current sunk on the* $\overline{\text{RESET}}$ *pin is less than the absolute maximum value specified for* $I_{INJ(RESET)}$ *in Section 19.2.2 on page 187.*

2 *When the LVD is enabled, it is recommended not to connect a pull-up resistor or capacitor. A 10nF pull-down capacitor is required to filter noise on the reset line.*

3 *In case a capacitive power supply is used, it is recommended to connect a 1MΩ pull-down resistor to the* $\overline{\text{RESET}}$ *pin to discharge any residual voltage induced by the capacitive effect of the power supply (this will add 5µA to the power consumption of the MCU).*

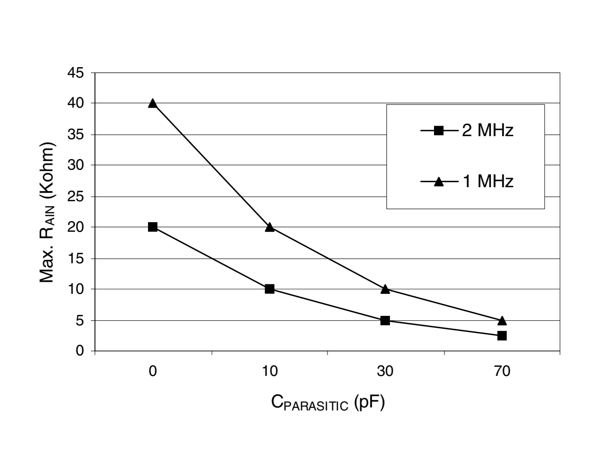
4 *Tips when using the LVD:*

A. Check that all recommendations related to reset circuit have been applied (see notes above).

B. Check that the power supply is properly decoupled (100nF + 10µF close to the MCU). Refer to AN1709 and AN2017. If this cannot be done, it is recommended to put a 100nF + 1MΩ pull-down on the $\overline{\text{RESET}}$ *pin.*

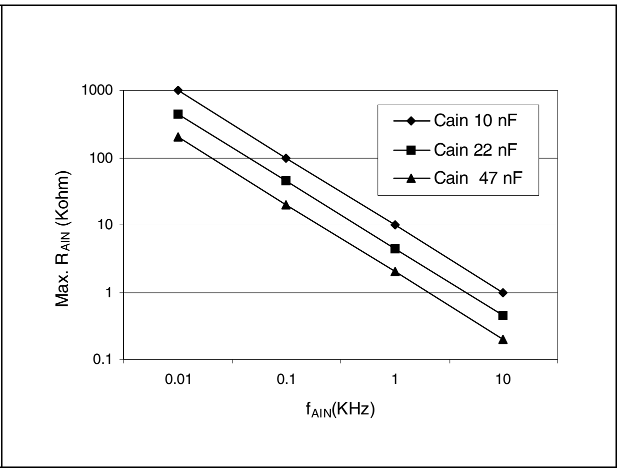
C. The capacitors connected on the $\overline{\text{RESET}}$ *pin and also the power supply are key to avoid any start-up marginality. In most cases, steps A and B above are sufficient for a robust solution. Otherwise, replace 10nF pull-down on the* $\overline{\text{RESET}}$ *pin with a 5µF to 20µF capacitor.*

Figure 97. R_{AIN} maximum versus f_{ADC} with C_{AIN} = 0pF[1]



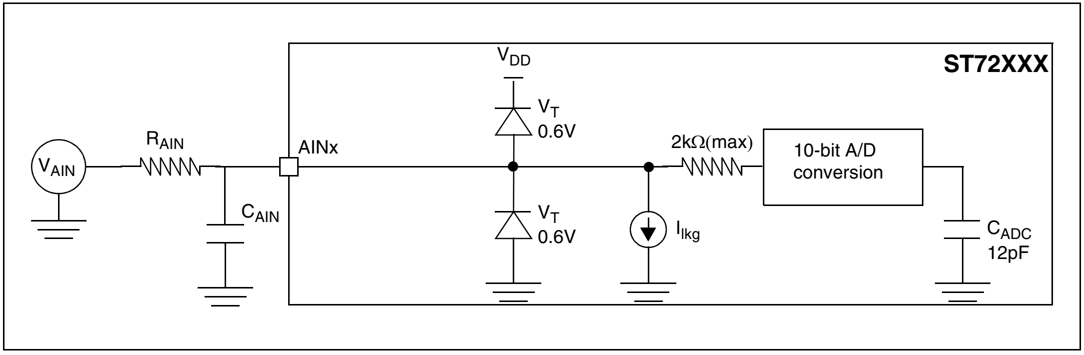
[1] $C_{PARASITIC}$ represents the capacitance of the PCB (dependent on soldering and PCB layout quality) plus the pad capacitance (3pF). A high $C_{PARASITIC}$ value will downgrade conversion accuracy. To remedy this, f_{ADC} should be reduced.

Figure 98. Recommended C_{AIN} and R_{AIN} values[1]



[1] This graph shows that, depending on the input signal variation (f_{AIN}), C_{AIN} can be increased for stabilization time and decreased to allow the use of a larger serial resistor (R_{AIN}).

Figure 99. Typical A/D converter application



19.12.1 Analog power supply and reference pins

Depending on the MCU pin count, the package may feature separate V_{AREF} and V_{SSA} analog power supply pins. These pins supply power to the A/D converter cell and function as the high and low reference voltages for the conversion.

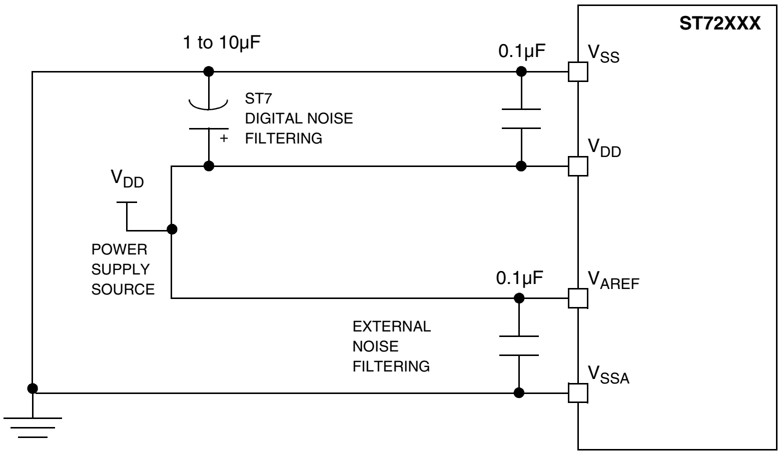
Separation of the digital and analog power pins allow board designers to improve A/D performance. Conversion accuracy can be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines (see *Section 19.12.2: General PCB design guidelines*).

19.12.2 General PCB design guidelines

To obtain best results, some general design and layout rules should be followed when designing the application PCB to shield the noise-sensitive, analog physical interface from noise-generating CMOS logic signals.

- Use separate digital and analog planes. The analog ground plane should be connected to the digital ground plane via a single point on the PCB.
- Filter power to the analog power planes. It is recommended to connect capacitors, with good high frequency characteristics, between the power and ground lines, placing 0.1µF and optionally, if needed 10pF capacitors as close as possible to the ST7 power supply pins and a 1 to 10µF capacitor close to the power source (see *Figure 100*).
- The analog and digital power supplies should be connected in a star network. Do not use a resistor, as V_{AREF} is used as a reference voltage by the A/D converter and any resistance would cause a voltage drop and a loss of accuracy.
- Properly place components and route the signal traces on the PCB to shield the analog inputs. Analog signals paths should run over the analog ground plane and be as short as possible. Isolate analog signals from digital signals that may switch while the analog inputs are being sampled by the A/D converter. Do not toggle digital outputs on the same I/O port as the A/D input being converted.

Figure 100. Power supply filtering



To avoid this, a semaphore is set to ‘1’ before checking the level change. The semaphore is changed to level '0' inside the interrupt routine. When a level change is detected, the semaphore status is checked. If it is ‘1’, it means that the last interrupt has been missed. In this case, the interrupt routine is invoked with the call instruction.

There is another possible case, that is, if PxOR or PxDDR are written to with global interrupts disabled (interrupt mask bit set). In this case, the semaphore is changed to ‘1’ when the level change is detected. Detecting a missed interrupt is done after the global interrupts are enabled (interrupt mask bit reset) and by checking the status of the semaphore. If it is ‘1’, it means that the last interrupt was missed and the interrupt routine is invoked with the call instruction.

To implement the workaround, the following software sequence is to be followed for writing into the PxOR/PxDDR registers. The example is for Port PF1 with falling edge interrupt sensitivity. The software sequence is given for both cases (global interrupts disabled / global interrupts enabled):

Case 1: Writing to PxOR or PxDDR with global interrupts enabled:

```
LD A,#01

LD sema,A
; set the semaphore to '1'
LD A,PFDR

AND A,#02

LD  X,A
; store the level before writing to PxOR/PxDDR
LD A,#$90

LD PFDDR,A
; Write to PFDDR
LD A,#$ff

LD  PFOR,A
 ; Write to PFOR
LD A,PFDR

AND A,#02

LD Y,A
; store the level after writing to PxOR/PxDDR
LD A,X
; check for falling edge
cp A,#02

jrne OUT

TNZ Y

jrne OUT

LD A,sema
```

```
; check for falling edge
cp A,#$02

jrne OUT

TNZ Y

jrne OUT

LD A,#$01

LD sema,A
; set the semaphore to '1' if edge is detected
RIM
 ; reset the interrupt mask
LD A,sema
; check the semaphore status
CP A,#$01

jrne OUT

call call_routine
; call the interrupt routine
RIM

OUT:
RIM
JP while_loop

.call_routine
; entry to call_routine
PUSH A

PUSH X

PUSH CC

.ext1_rt
; entry to interrupt routine
LD A,#$00

LD sema,A

IRET
```