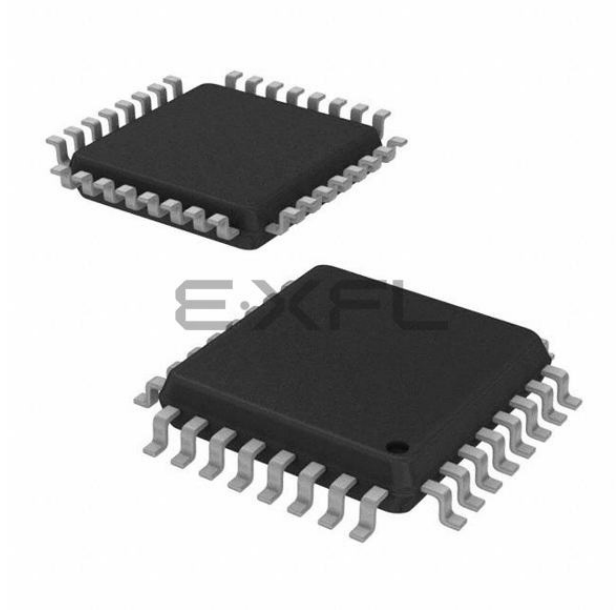


Welcome to E-XFL.COM

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
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fli49mk1t6tr

Table 3. Hardware register map[1] (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0032h	Reserved area (1 byte)				
0033h	WDG	WDGCR	Watchdog control register	7Fh	R/W
0034h	FLASH	FCSR	Flash control/status register	00h	R/W
0035h	EEPROM	EECSR	Data EEPROM control/status register	00h	R/W
0036h 0037h 0038h	ADC	ADCCSR ADCDRH ADCDRL	A/D control status register A/D data register high	00h xxh 0xh	R/W Read Only R/W
0039h	Reserved area (1 byte)				
003Ah	MCC	MCCSR	Main Clock Control/Status register	00h	R/W
003Bh 003Ch	Clock and reset	RCCR SICSR	RC oscillator control register System integrity control/status register	FFh 011x 0x00b	R/W R/W
003Dh		AVDTHCR	AVD threshold selection register / RC prescaler	00h	R/W
003Eh to 0047h	Reserved area (10 bytes)				
0048h 0049h	AWU	AWUCSR AWUPR	AWU control/status register AWU Preload register	FFh 00h	R/W R/W
004Ah 004Bh 004Ch 004Dh 004Eh 004Fh 0050h	DM[2]	DMCR DMSR DMBK1H DMBK1L DMBK2H DMBK2L DMCR2	DM control register DM status register DM breakpoint register 1 High DM breakpoint register 1 Low DM breakpoint register 2 High DM breakpoint register 2 Low DM control register 2	00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W R/W
0051h	Clock Controller	CKCNTCSR	Clock controller status register	09h	R/W
0052h to 0063h	Reserved area (18 bytes)				
0064h 0065h 0066h 0067h 0068h 0069h 006Ah	I2C	I2CCR I2CSR1 I2CSR2 I2CCCR I2COAR1 I2COAR2 I2CDR	I^2C control register I^2C status register 1 I^2C status register 2 I^2C clock control register I^2C own address register 1 I^2C own address register 2 I^2C data register	00h 00h 00h 00h 00h 40h 00h	R/W Read only Read only R/W R/W R/W R/W

1. Legend: x=undefined, R/W=read/write.
2. For a description of the debug module registers, see ICC protocol reference manual.

Depending on the ICP Driver code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection of the serial communication interface for downloading).

4.3.2 In-application programming (IAP)

This mode uses an IAP Driver program previously programmed in Sector 0 by the user (in ICP mode).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored etc.)
IAP mode can be used to program any memory areas except Sector 0, which is Write/Erase protected to allow recovery in case errors occur during the programming operation.

4.4 ICC interface

ICP needs a minimum of 4 and up to 6 pins to be connected to the programming tool. These pins are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- OSC1: main clock input for external source
- V_{DD}: application board power supply (optional, see Note 3)

Note: 1 If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.

2 During the ICP session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5 mA at high level (push pull output or pull-up resistor<1 kΩ). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R>1 kΩ or a reset management IC with open-drain output and pull-up resistor>1 kΩ, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.

3 The use of pin 7 of the ICC connector depends on the Programming Tool architecture. This pin must be connected when using most ST Programming Tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

4 In "enabled option byte" mode (38-pulse ICC mode), the internal RC oscillator is forced as a clock source, regardless of the selection in the option byte. In "disabled option byte" mode (35-pulse ICC mode), pin 9 has to be connected to the PB1/CLKIN pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte.

Caution: During normal operation the ICCCLK pin must be internally or externally pulled- up (external pull-up of 10 kΩ mandatory in noisy environment) to avoid entering ICC mode unexpectedly

4.5 Memory protection

There are two different types of memory protection: Read-out protection and Write/Erase Protection which can be applied individually.

4.5.1 Read-out protection

Read-out protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller. Both program and data EEPROM memory are protected.

In Flash devices, this protection is removed by reprogramming the option. In this case, both program and data EEPROM memory are automatically erased and the device can be reprogrammed.

Read-Out Protection selection depends on the device type:

- In Flash devices it is enabled and removed through the FMP_R bit in the option byte.
- In ROM devices it is enabled by mask option specified in the option list.

4.5.2 Flash write/erase protection

Write/erase protection, when set, makes it impossible to both overwrite and erase program memory. It does not apply to EEPROM data. Its purpose is to provide advanced security to applications and prevent any change being made to the memory content. Write/erase protection is enabled through the FMP_W bit in the option byte.

Caution: **Once set, write/erase protection can never be removed. A write-protected Flash device is no longer reprogrammable.**

4.6 Related documentation

For details on Flash programming and ICC protocol, refer to the ST7 Flash programming reference manual and to the ST7 ICC protocol reference manual.

4.7 Description of Flash control/status register (FCSR)

This register controls the XFlash erasing and programming using ICP, IAP or other programming methods.

1st RASS Key: 0101 0110 (56h)

2nd RASS Key: 1010 1110 (AEh)

When an EPB or another programming tool is used (in socket or ICP mode), the RASS keys are sent automatically.

Reset value: 000 0000 (00h)

7							0
0	0	0	0	0	OPT	LAT	PGM
Read/write							

5.3 Memory access

The data EEPROM memory read/write access modes are controlled by the E2LAT bit of the EEPROM Control/Status register (EECSR). The flowchart in *Figure 7* describes these different memory access modes.

5.3.1 Read operation (E2LAT=0)

The EEPROM can be read as a normal ROM location when the E2LAT bit of the EECSR register is cleared.

On this device, data EEPROM can also be used to execute machine code. Take care not to write to the data EEPROM while executing from it. This would result in an unexpected code being executed.

5.3.2 Write operation (E2LAT=1)

To access the write mode, the E2LAT bit has to be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 32 data latches according to its address.

When PGM bit is set by the software, all the previous bytes written in the data latches (up to 32) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must take care that all the bytes written between two programming sequences have the same high address: only the five Least Significant Bits of the address can change.

At the end of the programming cycle, the PGM and LAT bits are cleared simultaneously.

Note: Care should be taken during the programming cycle. Writing to the same memory location will over-program the memory (logical AND between the two write access data result) because the data latches are only cleared at the end of the programming cycle and by the falling edge of the E2LAT bit. It is not possible to read the latched data (see Figure 9).

Figure 7. Data EEPROM programming flowchart

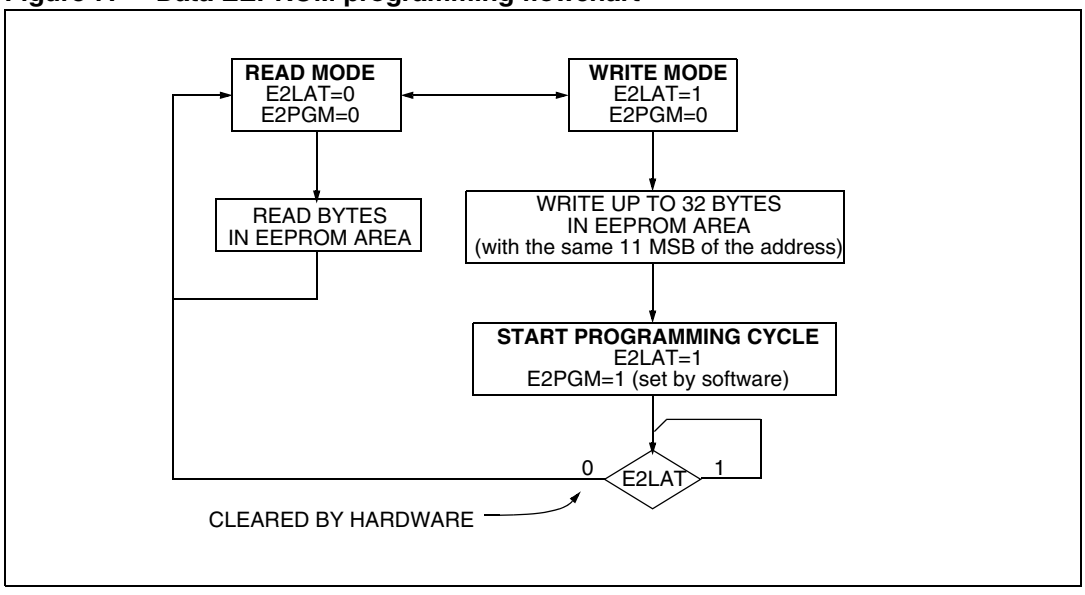
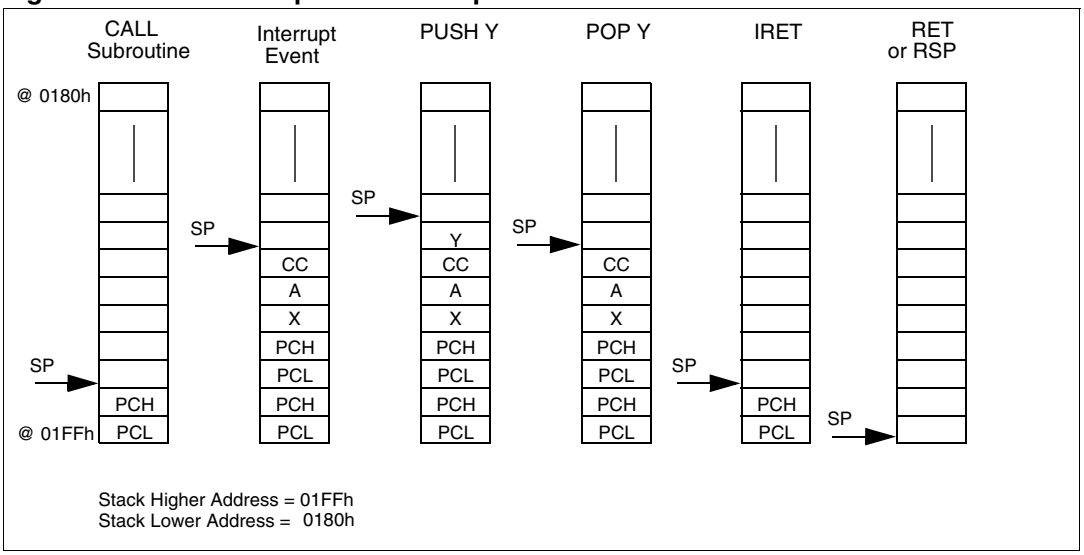


Figure 11. Stack manipulation example



7.5 Register description

7.5.1 Main clock control/status register (MCCSR)

Reset value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	MCO	SMS
Read/write							

Bits 7:2 = Reserved, must be kept cleared.

Bit 1 = **MCO** *Main clock out enable bit*

This bit is read/write by software and cleared by hardware after a reset. This bit allows to enable the MCO output clock.

0: MCO clock disabled, I/O port free for general purpose I/O.

1: MCO clock enabled.

Bit 0 = **SMS** *Slow mode selection bit*

This bit is read/write by software and cleared by hardware after a reset. This bit selects the input clock f_{OSC} or $f_{OSC}/32$.

0: Normal mode ($f_{CPU} = f_{OSC}$

1: Slow mode ($f_{CPU} = f_{OSC}/32$)

7.5.2 RC control register (RCCR)

Reset value: 1111 1111 (FFh)

7							0
CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2
Read/write							

Bits 7:0 = **CR[9:2]** *RC oscillator frequency adjustment bits*

These bits must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. The application can store the correct value for each voltage range in Flash memory and write it to this register at start-up.

00h = maximum available frequency

FFh = lowest available frequency

These bits are used with the CR[1:0] bits in the SICSR register. Refer to *Chapter 7.5.3*.

Note: To tune the oscillator, write a series of different values in the register until the correct frequency is reached. The fastest method is to use a dichotomy starting with 80h.

7.5.5 Clock controller control/status register (CKCNTCSR)

Reset value: 0000 1001 (09h)

7							0
0	0	0	0	AWU_FLAG	RC_FLAG	0	RC/AWU
Read/write							

Bits 7:4 = Reserved, must be kept cleared.

Bit 3 = **AWU_FLAG** *AWU selection bit*

This bit is set and cleared by hardware.

0: No switch from AWU to RC requested

1: AWU clock activated and temporization completed

Bit 2 = **RC_FLAG** *RC selection bit*

This bit is set and cleared by hardware.

0: No switch from RC to AWU requested

1: RC clock activated and temporization completed

Bit 1 = Reserved, must be kept cleared.

Bit 0 = **RC/AWU** *RC/AWU selection bit*

0: RC enabled

1: AWU enabled (default value)

Table 13. Clock register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
003Ah	MCCSR Reset value	- 0	- 0	- 0	- 0	- 0	- 0	MCO 0	SMS 0
003Bh	RCCR Reset value	CR9 1	CR8 1	CR7 1	CR6 1	CR5 1	CR4 1	CR3 1	CR2 1
003Ch	SICSR Reset value	- 0	CR1 1	CR0 1	WDGRF 0	- 0	LVDRF x	AVDF x	AVDIE x
003Dh	AVDTHCR Reset value	CK2 0	CK1 0	CK0 0	- 0	- 0	- 0	AVD1 0	AVD0 0
0051h	CKCNTCSR Reset value	- 0	- 0	- 0	- 0	AWU_ FLAG 1	RC_FLA G 0	- 0	RC/AWU 1

8.5.3 External interrupt control register (EICR)

Reset value: 0000 0000 (00h)

7							0
0	0	IS21	IS20	IS11	IS10	IS01	IS00
Read/write							

Bits 7:6 = Reserved, must be kept cleared.

Bits 5:4 = **IS2[1:0]** *ei2 sensitivity bits*

These bits define the interrupt sensitivity for ei2 (Port C) according to *Table 19*.

Bits 3:2 = **IS1[1:0]** *ei1 sensitivity bits*

These bits define the interrupt sensitivity for ei1 (Port B) according to *Table 19*.

Bits 1:0 = **IS0[1:0]** *ei0 sensitivity bits*

These bits define the interrupt sensitivity for ei0 (Port A) according to *Table 19*.

Note: 1 These 8 bits can be written only when the I bit in the CC register is set.

2 Changing the sensitivity of a particular external interrupt clears this pending interrupt. This can be used to clear unwanted pending interrupts. Refer to Section : External interrupt function.

Table 19. Interrupt sensitivity bits

ISx1	ISx0	External interrupt sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Table 30. PC3 pin (continued)

Mode	DDR	OR
open-drain output	1	0
push-pull output	1	1

Table 31. Port configuration

Port	Pin name	Input		Output	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA5:0	floating	pull-up interrupt	open-drain	push-pull
	PA7:6	floating	interrupt	true open-drain	
Port B	PB7:0	floating	pull-up interrupt	open-drain	push-pull
Port C	PC7:4, PC2:0	floating	pull-up interrupt	open-drain	push-pull
	PC3	floating	pull-up	open-drain	push-pull

Table 32. I/O port register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0000h	PADR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0001h	PADDR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0002h	PAOR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0003h	PBDR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0004h	PBDDR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0005h	PBOR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0006h	PCDR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0007h	PCDDR	MSB							LSB
	Reset value	0	0	0	0	0	0	0	0
0008h	PCOR	MSB							LSB
	Reset value	0	0	0	0	1	0	0	0

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is free-running: it counts down even if the watchdog is disabled. The value to be stored in the CR register must be between FFh and C0h (see *Table 33: Watchdog timing*):

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a reset.

Table 33. Watchdog timing (1)(2)

f_{CPU} = 8 MHz		
WDG counter code	**min [ms]**	**max [ms]**
C0h	1	2
FFh	127	128

1. The timing variation shown in *Table 33* is due to the unknown status of the prescaler when writing to the CR register.
2. The number of CPU clock cycles applied during the reset phase (256 or 4096) must be taken into account in addition to these timings.

11.1.4 Hardware watchdog option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the CR is not used.

Refer to the option byte description in *Section 14 on page 173*.

Using Halt mode with the WDG (WDGHALT option)

If Halt mode with Watchdog is enabled by option byte (No watchdog reset on HALT instruction), it is recommended before executing the HALT instruction to refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller. Same behavior in Active-halt mode.

11.1.5 Interrupts

None.

11.1.6 Register description

Control register (WDGCR)

Reset value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0
Read/Write							

Bit 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.

Bits 6:0 = **T[6:0]** *7-bit timer (MSB to LSB)*

These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

Table 34. Watchdog timer register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0033h	WDGCR Reset value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

PWMX control status register (PWMxCSR)

Reset value: 0000 0000 (00h)

7							0
0	0	0	0	OP_EN	OPEDGE	OPx	CMPFx
Read/write							

Bits 7:4= Reserved, must be kept cleared.

Bit 3 = OP_EN *One-pulse mode enable bit*

This bit is read/write by software and cleared by hardware after a reset. This bit enables the One-pulse feature for PWM2 and PWM3 **(only available for PWM3CSR)**

0: One-pulse mode disable for PWM2/3.

1: One-pulse mode enable for PWM2/3.

Bit 2 = OPEDGE *One-pulse edge selection bit*

This bit is read/write by software and cleared by hardware after a reset. This bit selects the polarity of the LTIC signal for One-pulse feature. This bit will be effective only if OP_EN bit is set **(only available for PWM3CSR)**

0: Falling edge of LTIC is selected.

1: Rising edge of LTIC is selected.

Bit 1 = **OPx** *PWMx output polarity bit*

This bit is read/write by software and cleared by hardware after a reset. This bit selects the polarity of the PWM signal.

0: The PWM signal is not inverted.

1: The PWM signal is inverted.

Bit 0 = **CMPFx** PWMx *compare flag*

This bit is set by hardware and cleared by software by reading the PWMxCSR register. It indicates that the upcounter value matches the Active DCRx register value.

0: Upcounter value does not match DCRx value.

1: Upcounter value matches DCRx value.

Break control register (BREAKCR)

Reset value: 0000 0000 (00h)

7							0
0	BREDGE	BA	BPEN	PWM3	PWM2	PWM1	PWM0
Read/write							

Bit 7 = Reserved

Master mode

To switch from default Slave mode to Master mode a Start condition generation is needed.

Start condition

Setting the START bit while the BUSY bit is cleared causes the interface to switch to Master mode (M/SL bit set) and generates a Start condition.

Once the Start condition is sent, the EVF and SB bits are set by hardware with an interrupt if the ITE bit is set.

The master then waits for a read of the SR1 register followed by a write in the DR register with the Slave address, **holding the SCL line low** (see *Figure 57* Transfer sequencing EV5).

Slave address transmission

1. The slave address is then sent to the SDA line via the internal shift register.
   - In 7-bit addressing mode, one address byte is sent.
   - In 10-bit addressing mode, sending the first byte including the header sequence causes the following event. The EVF bit is set by hardware with interrupt generation if the ITE bit is set.
2. The master then waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see *Figure 57* transfer sequencing EV9).
3. Then the second address byte is sent by the interface.
4. After completion of this transfer (and acknowledge from the slave if the ACK bit is set), the EVF bit is set by hardware with interrupt generation if the ITE bit is set.
5. The master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see *Figure 57* transfer sequencing EV6).
6. Next the master must enter receiver or transmitter mode.

Note: In 10-bit addressing mode, to switch the master to receiver mode, software must generate a repeated Start condition and resend the header sequence with the least significant bit set (11110xx1).

Master receiver

Following the address transmission and after SR1 and CR registers have been accessed, the **master receives bytes from the SDA line into the** DR register **via** the internal shift register. After each byte the interface generates in sequence:

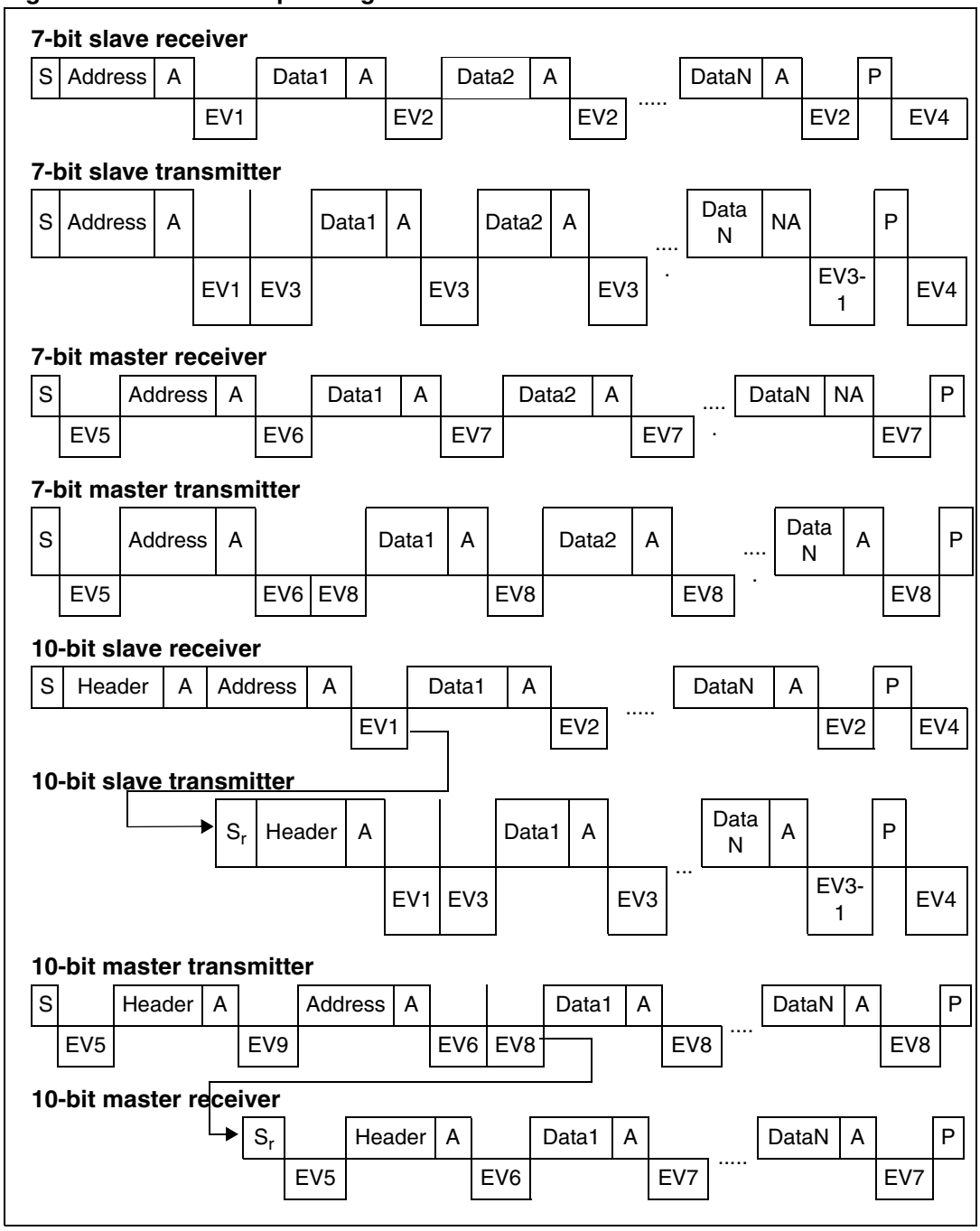
- Acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see *Figure 57* transfer sequencing EV7).

To close the communication: before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

Figure 57. Transfer sequencing



1. S=Start, S_r = Repeated Start, P=Stop, A=Acknowledge, NA=Non-acknowledge, EVx=Event (with interrupt if ITE=1).
2. **EV1:** EVF=1, ADSL=1, cleared by reading SR1 register.
3. **EV2:** EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.
4. **EV3:** EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.
5. **EV3-1:** EVF=1, AF=1, BTF=1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing DR register (DR=FFh). If lines are released by STOP=1, STOP=0, the

Bit 1 = **STOP** *Generation of a Stop condition bit*

This bit is set and cleared by software. It is also cleared by hardware in master mode. Note: This bit is not cleared when the interface is disabled (PE=0).

- In master mode:

  0: No stop generation

  1: Stop generation after the current byte transfer or after the current Start condition is sent. The STOP bit is cleared by hardware when the Stop condition is sent.

- In slave mode:

  0: No stop generation

  1: Release the SCL and SDA lines after the current byte transfer (BTF=1). In this mode the STOP bit has to be cleared by software.

Bit 0 = **ITE** *Interrupt enable bit*

This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE=0).

0: Interrupts disabled

1: Interrupts enabled

Refer to *Figure 58* for the relationship between the events and the interrupt.

SCL is held low when the ADD10, SB, BTF or ADSL flags or an EV6 event (See *Figure 57*) is detected.

Table 45. I²C register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0064h	**I2CCR** Reset value	0	0	PE 0	ENGC 0	START 0	ACK 0	STOP 0	ITE 0
0065h	**I2CSR1** Reset value	EVF 0	ADD10 0	TRA 0	BUSY 0	BTF 0	ADSL 0	M/SL 0	SB 0
0066h	**I2CSR2** Reset value	0	0	0	AF 0	STOPF 0	ARLO 0	BERR 0	GCAL 0
0067h	**I2CCCR** Reset value	FM/SM 0	CC6 0	CC5 0	CC4 0	CC3 0	CC2 0	CC1 0	CC0 0
0068h	**I2COAR1** Reset value	ADD7 0	ADD6 0	ADD5 0	ADD4 0	ADD3 0	ADD2 0	ADD1 0	ADD0 0
0069h	**I2COAR2** Reset value	FR1 0	FR0 1	0	0	0	ADD9 0	ADD8 0	0
006Ah	**I2CDR** Reset value	MSB 0	0	0	0	0	0	0	LSB 0

13.6.2 Crystal and ceramic resonator oscillators

The ST7 internal clock can be supplied with ten different crystal/ceramic resonator oscillators. All the information given in this paragraph are based on characterization results with specified typical external components. In the application, 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 start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...).

Table 74. Crystal/ceramic resonator oscillator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CrOSC}	Crystal oscillator frequency		2		16	MHz
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)		see table below			pF

Table 75. Typical ceramic resonators(1)

Supplier	f_{CrOSC} (MHz)	Typical ceramic resonators							Supply voltage range (V)	Temperature range (°C)
		Reference	Type	Oscillator modes	C1 (pF)	C2 (pF)	Rf	Rd (Ohm)		
Murata	2	CSTCC2M00G56Z-R0	SMD	LP or MP	(47)	(47)	open	0	2.4 to 5.5	-40 to 85
	4	CSTCR4M00G55Z-R0	SMD	MP or MS	(39)	(39)	open	0		
		CSTLS4M00G56Z-B0	LEAD	MP or MS	(47)	(47)	open	0		
	8	CSTCE8M00G52Z-R0	SMD	MS or HS	(10)	(10)	open	0		
		CSTLS8M00G53Z-B0	LEAD	MS or HS	(15)	(15)	open	0		
	16	CSTCE16M0V51Z-R0	SMD	HS	(5)	(5)	open	0	3.3 to 5.5	
		CSTLS16M0X53Z-B0	LEAD	HS	(15)	(15)	47k	0		

1. ( ) means load capacitor built in resonator.
Resonator characteristics given by the ceramic resonator manufacturer.
SMD = [-R0: plastic tape package Ø = 180 mm), -B0: Bulk]
LEAD = [-B0: bulk]
For more information on these resonators, please consult www.murata.com

Figure 74. Two typical applications with unused I/O pin

1. During normal operation the ICCCLK pin must be pulled-up, internally or externally (external pull-up of 10 kΩ mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset.
2. I/O can be left unconnected if it is configured as output (0 or 1) by the software. This has the advantage of greater EMC robustness and lower cost.

Figure 75. R_{pu} resistance versus voltage at four different temperatures

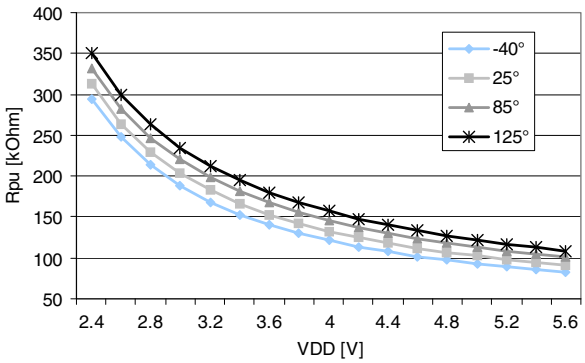
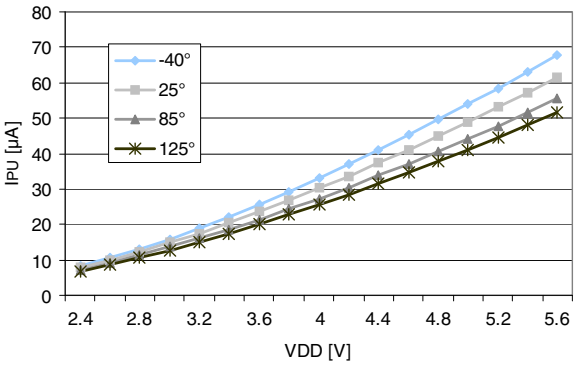


Figure 76. I_{pu} current versus voltage at four different temperatures



13.10 Control pin characteristics

13.10.1 Asynchronous $\overline{\text{RESET}}$ pin

T_A = -40 to 125 °C, unless otherwise specified.

Table 85. Asynchronous $\overline{\text{RESET}}$ pin characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage			V_{SS} - 0.3		$0.3V_{DD}$	V
V_{IH}	Input high level voltage			$0.7V_{DD}$		V_{DD}+0.3	
V_{hys}	Schmitt trigger voltage hysteresis(1)				2		V
V_{OL}	Output low level voltage (2)	V_{DD}= 5 V	I_{IO} = +2 mA		200		mV
R_{ON}	Pull-up equivalent resistor(3)	V_{IN}=V_{SS}	V_{DD} = 5 V	30	50	70	kΩ
			V_{DD} = 3 V		90(1)		
$t_{w(RSTL)out}$	Generated reset pulse duration	Internal reset sources			90(1)		µs
$t_{h(RSTL)in}$	External reset pulse hold time(4)			20			µs
$t_{g(RSTL)in}$	Filtered glitch duration				200		ns

1. Data based on characterization results, not tested in production
2. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section Table 59. on page 141* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
3. The R_{ON} pull-up equivalent resistor is based on a resistive transistor. Specified for voltages on $\overline{\text{RESET}}$ pin between V_{ILmax} and V_{DD}
4. To guarantee the reset of the device, a minimum pulse has to be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.

14.3.1 Starter kits

ST offers complete, affordable **starter kits**. Starter kits are complete hardware/software tool packages that include features and samples to help you quickly start developing your application.

14.3.2 Development and debugging tools

Application development for ST7 is supported by fully optimizing **C Compilers** and the **ST7 Assembler-Linker** toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16 Kbytes of code.

The range of hardware tools includes a full-featured **STice** Emulator**,** the low-cost **RLink** and the **ST7-STICK** in-circuit debugger/programmer. These tools are supported by the **ST7 Toolset** from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

14.3.3 Programming tools

During the development cycle, the **STice** emulator, the **ST7-STICK** and the **RLink** provide in-circuit programming capability for programming the Flash microcontroller on your application board.

ST also provides a low-cost dedicated in-circuit programmer and **ST7 Socket Boards,** which provide all the sockets required for programming any of the devices in a specific ST7 sub-family with any tool with in-circuit programming capability for ST7.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

14.3.4 Order codes for development and programming tools

Table 94 below lists the ordering codes for the ST7LITE49M development and programming tools. For additional ordering codes for spare parts and accessories, refer to the online product selector at www.st.com/mcu.

Table 94. Development tool order codes for the ST7LITE49M family

MCU	Debugging and programming tool	Starter kit with demo board	ST socket boards
ST7FLI49MK1T6 ST7FLI49MK1B6	STice emulator(1) STX-RLINK(2)(3) ST7-STICK(4)(5)	ST7FLITE-SK/RAIS(2)(3)	SBX-DIP32CD and SBX-QP32BC Socket boards(4)

1. Contact local ST sales office for sales types.
2. USB connection to PC.
3. Available from ST or from Raisonance, www.raisonance.com.
4. Add suffix /EU, /UK or /US for the power supply for your region.
5. Parallel port connection to PC.