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	CANbus, LINbusSCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	32
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 11x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f561j9tae

SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.2 Auxiliary Voltage Detector (AVD)

The Voltage Detector function (AVD) is based on an analog comparison between a $V_{IT-(AVD)}$ and $V_{IT+(AVD)}$ reference value and the V_{DD} main supply. The $V_{IT-(AVD)}$ reference value for falling voltage is lower than the $V_{IT+(AVD)}$ reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real time status bit (AVDF) in the SICSR register. This bit is read only.

Caution: The AVD function is active only if the LVD is enabled through the option byte.

6.4.2.1 Monitoring the V_{DD} Main Supply

If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the $V_{IT+(AVD)}$ or $V_{IT-(AVD)}$ threshold (AVDF bit toggles).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See Figure 16.

The interrupt on the rising edge is used to inform the application that the V_{DD} warning state is over.

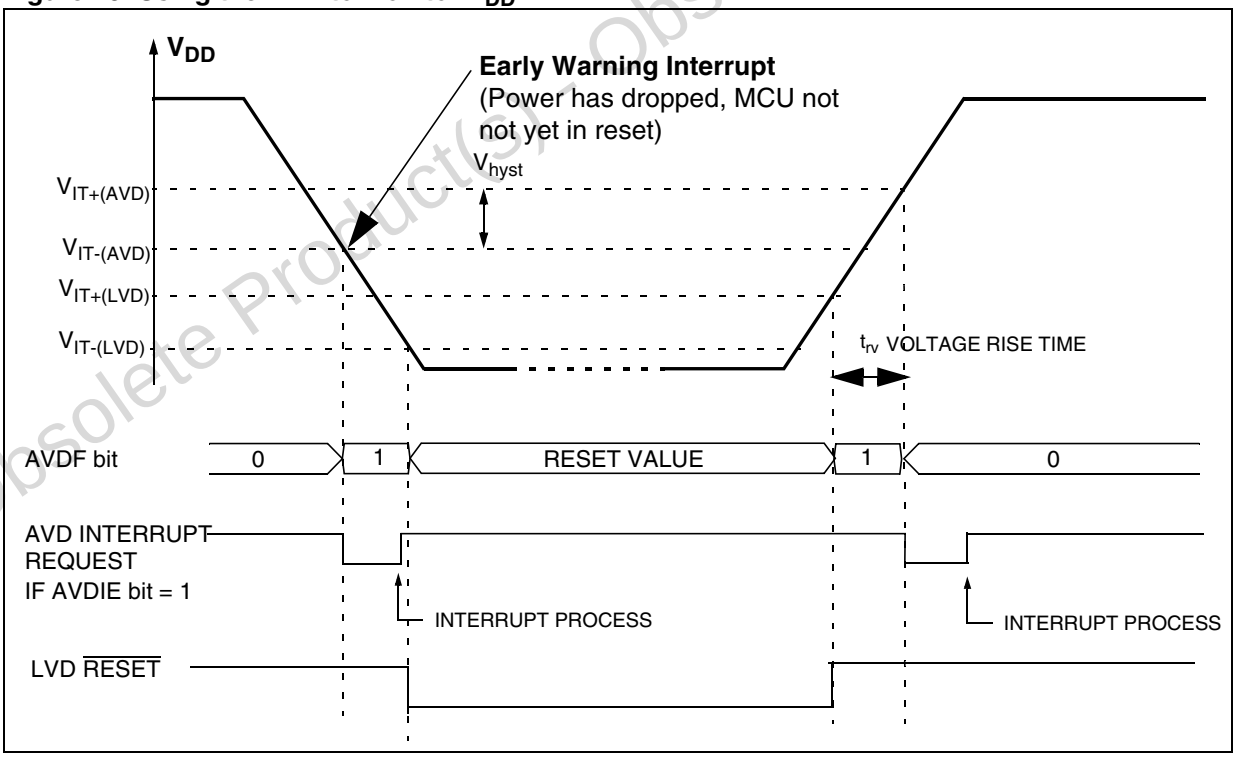
If the voltage rise time t_{rv} is less than 256 or 4096 CPU cycles (depending on the reset delay selected by option byte), no AVD interrupt will be generated when $V_{IT+(AVD)}$ is reached.

If t_{rv} is greater than 256 or 4096 cycles then:

– If the AVD interrupt is enabled before the $V_{IT+(AVD)}$ threshold is reached, then two AVD interrupts will be received: The first when the AVDIE bit is set and the second when the threshold is reached.

– If the AVD interrupt is enabled after the $V_{IT+(AVD)}$ threshold is reached, then only one AVD interrupt occurs.

Figure 16. Using the AVD to Monitor V_{DD}



Obsolete Product(s) - Obsolete Product(s)

WINDOW WATCHDOG (Cont'd)

The application program must write in the WDGCR register at regular intervals during normal operation to prevent an MCU reset. This operation must occur only when the counter value is lower than the window register value. The value to be stored in the WDGCR register must be between FFh and C0h (see Figure 2):

– Enabling the watchdog:
When Software Watchdog is selected (by option byte), the watchdog is disabled after a reset. It is enabled by setting the WDGA bit in the WDGCR register, then it cannot be disabled again except by a reset.

When Hardware Watchdog is selected (by option byte), the watchdog is always active and the WDGA bit is not used.

– Controlling the downcounter:
This downcounter is free-running: It counts down even if the watchdog is disabled. When the watchdog is enabled, the T6 bit must be 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 (see Figure 2. Approximate Timeout Duration). The timing varies between a minimum and a maximum value due to the unknown status of the prescaler when writing to the WDGCR register (see Figure 3).

The window register (WDGWR) contains the high limit of the window: To prevent a reset, the downcounter must be reloaded when its value is lower than the window register value and greater than 3Fh. Figure 4 describes the window watchdog process.

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

– Watchdog Reset on Halt option
If the watchdog is activated and the watchdog reset on halt option is selected, then the HALT instruction will generate a Reset.

10.1.4 Using Halt Mode with the WDG

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.

Obsolete Product(s) - Obsolete Product(s)

WINDOW WATCHDOG (Cont'd)

Figure 36. Exact Timeout Duration (t_{min} and t_{max})

WHERE:

$t_{min0} = (LSB + 128) \times 64 \times t_{OSC2}$

$t_{max0} = 16384 \times t_{OSC2}$

$t_{OSC2} = 125ns$ if $f_{OSC2} = 8$ MHz

CNT = Value of T[5:0] bits in the WDGCR register (6 bits)

MSB and LSB are values from the table below depending on the timebase selected by the TB[1:0] bits in the MCCSR register

TB1 Bit (MCCSR Reg.)	TB0 Bit (MCCSR Reg.)	Selected MCCSR Timebase	MSB	LSB
0	0	2ms	4	59
0	1	4ms	8	53
1	0	10ms	20	35
1	1	25ms	49	54

To calculate the minimum Watchdog Timeout (t_{min}):

IF $CNT < \left\lceil \frac{MSB}{4} \right\rceil$ **THEN** $t_{min} = t_{min0} + 16384 \times CNT \times t_{osc2}$

ELSE $t_{min} = t_{min0} + \left[16384 \times \left(CNT - \left\lceil \frac{4CNT}{MSB} \right\rceil \right) + (192 + LSB) \times 64 \times \left\lceil \frac{4CNT}{MSB} \right\rceil \right] \times t_{osc2}$

To calculate the maximum Watchdog Timeout (t_{max}):

IF $CNT \leq \left\lceil \frac{MSB}{4} \right\rceil$ **THEN** $t_{max} = t_{max0} + 16384 \times CNT \times t_{osc2}$

ELSE $t_{max} = t_{max0} + \left[16384 \times \left(CNT - \left\lceil \frac{4CNT}{MSB} \right\rceil \right) + (192 + LSB) \times 64 \times \left\lceil \frac{4CNT}{MSB} \right\rceil \right] \times t_{osc2}$

Note: In the above formulae, division results must be rounded down to the next integer value.

Example:

With 2ms timeout selected in MCCSR register

Value of T[5:0] Bits in WDGCR Register (Hex.)	Min. Watchdog Timeout (ms) t_{min}	Max. Watchdog Timeout (ms) t_{max}
00	1.496	2.048
3F	128	128.552

Obsolete Product(s) - Obsolete Product(s)

ON-CHIP PERIPHERALS (Cont'd)

INPUT CAPTURE CONTROL / STATUS REGISTER (ARTICCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	CS2	CS1	CIE2	CIE1	CF2	CF1

Bit 7:6 = Reserved, always read as 0.

Bit 5:4 = **CS[2:1]** *Capture Sensitivity*
These bits are set and cleared by software. They determine the trigger event polarity on the corresponding input capture channel.
0: Falling edge triggers capture on channel x.
1: Rising edge triggers capture on channel x.

Bit 3:2 = **CIE[2:1]** *Capture Interrupt Enable*
These bits are set and cleared by software. They enable or disable the Input capture channel interrupts independently.
0: Input capture channel x interrupt disabled.
1: Input capture channel x interrupt enabled.

Bit 1:0 = **CF[2:1]** *Capture Flag*
These bits are set by hardware and cleared by software reading the corresponding ARTICRx register. Each CFx bit indicates that an input capture x has occurred.
0: No input capture on channel x.
1: An input capture has occurred on channel x.

INPUT CAPTURE REGISTERS (ARTICRx)

Read only

Reset Value: 0000 0000 (00h)

7							0
IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0

Bit 7:0 = **IC[7:0]** *Input Capture Data*

These read only bits are set and cleared by hardware. An ARTICRx register contains the 8-bit auto-reload counter value transferred by the input capture channel x event.

Obsolete Product(s) - Obsolete Product(s)

16-BIT TIMER (Cont'd)

OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB

COUNTER HIGH REGISTER (CHR)

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

COUNTER LOW REGISTER (CLR)

Read Only
Reset Value: 1111 1100 (FCh)

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 accessing the CSR register clears the TOF bit.

7							0
MSB							LSB

ALTERNATE COUNTER HIGH REGISTER (ACHR)

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

ALTERNATE COUNTER LOW REGISTER (ACLR)

Read Only
Reset Value: 1111 1100 (FCh)

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.

7							0
MSB							LSB

INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

INPUT CAPTURE 2 LOW REGISTER (IC2LR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

Obsolete Product(s) - Obsolete Product(s)

10.5 8-BIT TIMER (TIM8)

10.5.1 Introduction

The timer consists of a 8-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the clock prescaler.

10.5.2 Main Features

- Programmable prescaler: f_{CPU} divided by 2, 4 , 8 or f_{OSC2} divided by 8000.
- Overflow status flag and maskable interrupt
- Output compare functions with
  - 2 dedicated 8-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Input capture functions with
  - 2 dedicated 8-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One pulse mode
- Reduced Power Mode
- 4 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2)*

The Block Diagram is shown in Figure 59.

***Note:** Some timer pins may not be available (not bonded) in some ST7 devices. Refer to the device pin out description.

When reading an input signal on a non-bonded pin, the value will always be '1'.

10.5.3 Functional Description

10.5.3.1 Counter

The main block of the Programmable Timer is a 8-bit free running upcounter and its associated 8-bit registers.

These two read-only 8-bit registers contain the same value but with the difference that reading the ACTR register does not clear the TOF bit (Timer overflow flag), located in the Status register, (SR).

Writing in the CTR register or ACTR register resets the free running counter to the FCh value.
Both counters have a reset value of FCh (this is the only value which is reloaded in the 8-bit timer). The reset value of both counters is also FCh in One Pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as shown in Table 19 Clock Control Bits. The value in the counter register repeats every 512, 1024, 2048 or 20480000 f_{CPU} clock cycles depending on the CC[1:0] bits.
The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or $f_{OSC2}/8000$.

For example, if $f_{OSC2}/8000$ is selected, and f_{OSC2} = 8 MHz, the timer frequency will be 1 ms. Refer to Table 19 on page 105.

Obsolete Product(s) - Obsolete Information(s)

8-BIT TIMER (Cont'd)

10.5.8 8-bit Timer Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
3C	**CR2**	OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	0
3D	**CR1**	ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
3E	**CSR**	ICF1	OCF1	TOF	ICF2	OCF2	TIMD		
3F	**IC1R**	MSB							LSB
40	**OC1R**	MSB							LSB
41	**CTR**	MSB							LSB
42	**ACTR**	MSB							LSB
43	**IC2R**	MSB							LSB
44	**OC2R**	MSB							LSB

Obsolete Product(s) - Obsolete Product(s)

SERIAL PERIPHERAL INTERFACE (Cont'd)

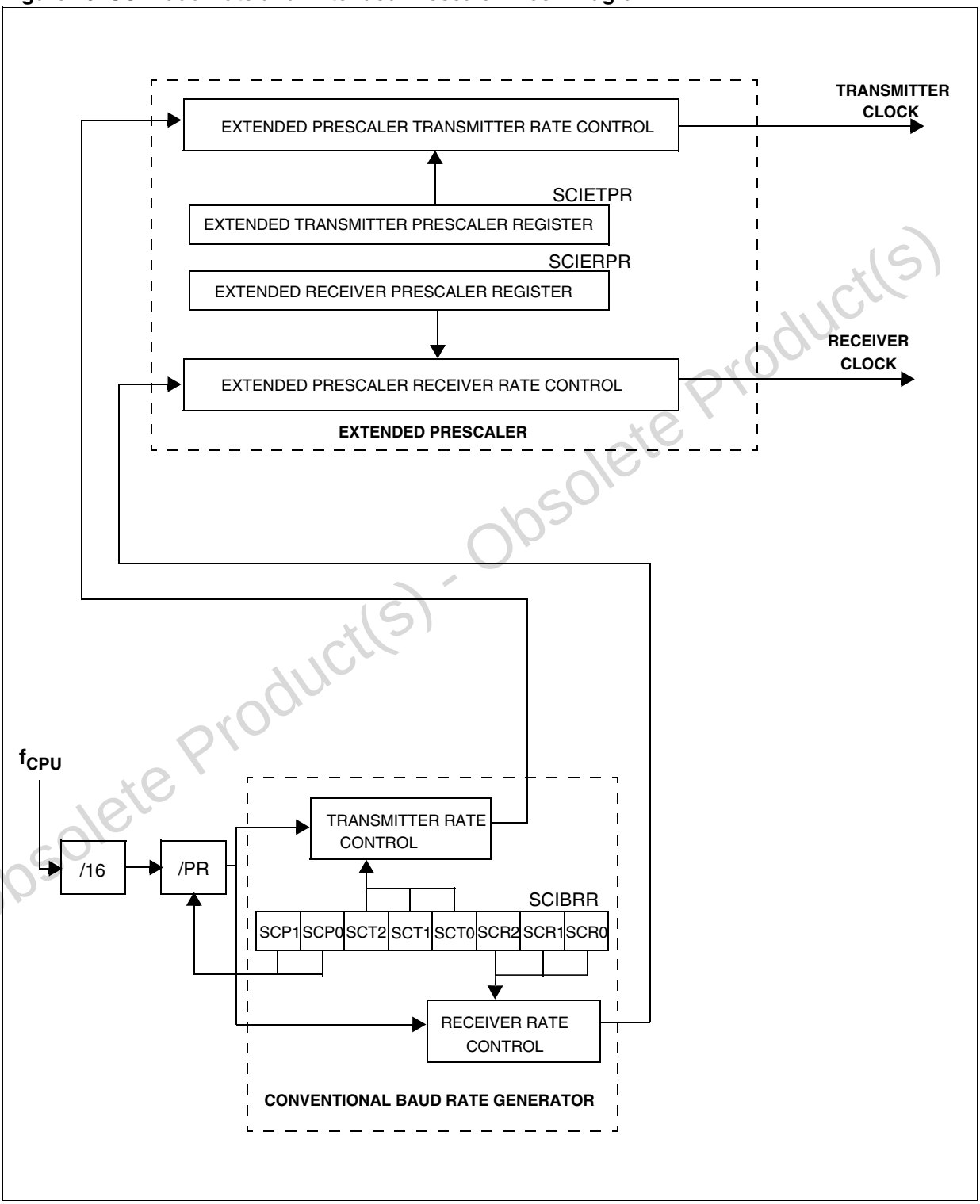
Table 22. SPI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
21	**SPIDR** Reset Value	MSB x	x	x	x	x	x	x	LSB x
22	**SPICR** Reset Value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
23	**SPICSR** Reset Value	SPIF 0	WCOL 0	OVR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

Obsolete Product(s) - Obsolete Product(s)

LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)

Figure 79. SCI Baud Rate and Extended Prescaler Block Diagram



Obsolete Product(s) - Obsolete Product(s)

LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)

10.7.6 Low Power Modes

Mode	Description
WAIT	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
HALT	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

10.7.7 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE		
Received Data Ready to be Read	RDRF	RIE		
Overrun Error or LIN Synch Error Detected	OR/ LHE			
Idle Line Detected	IDLE	ILIE		
Parity Error	PE	PIE		
LIN Header Detection	LHDF	LHIE		

The SCI interrupt events are connected to the same interrupt vector (see Interrupts chapter).

These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Obsolete Product(s) - Obsolete Product(s)

LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode)

10.7.9 LIN Mode - Functional Description.

The block diagram of the Serial Control Interface, in LIN slave mode is shown in Figure 5.

It uses six registers:

– 3 control registers: SCICR1, SCICR2 and SCICR3

– 2 status registers: the SCISR register and the LHLR register mapped at the SCIERPR address

– A baud rate register: LPR mapped at the SCIBRR address and an associated fraction register LPFR mapped at the SCIETPR address

The bits dedicated to LIN are located in the SCICR3. Refer to the register descriptions in Section 0.1.10 for the definitions of each bit.

10.7.9.1 Entering LIN Mode

To use the LINSCI in LIN mode the following configuration must be set in SCICR3 register:

– Clear the M bit to configure 8-bit word length.

– Set the LINE bit.

Master

To enter master mode the LSLV bit must be reset In this case, setting the SBK bit will send 13 low bits.

Then the baud rate can programmed using the SCIBRR, SCIERPR and SCIETPR registers.

In LIN master mode, the Conventional and / or Extended Prescaler define the baud rate (as in standard SCI mode)

Slave

Set the LSLV bit in the SCICR3 register to enter LIN slave mode. In this case, setting the SBK bit will have no effect.

In LIN Slave mode the LIN baud rate generator is selected instead of the Conventional or Extended Prescaler. The LIN baud rate generator is common to the transmitter and the receiver.

Then the baud rate can be programmed using LPR and LPRF registers.

Note: It is mandatory to set the LIN configuration first before programming LPR and LPRF, because the LIN configuration uses a different baud rate generator from the standard one.

10.7.9.2 LIN Transmission

In LIN mode the same procedure as in SCI mode has to be applied for a LIN transmission.

To transmit the LIN Header the proceed as follows:

– First set the SBK bit in the SCICR2 register to start transmitting a 13-bit LIN Synch Break

– reset the SBK bit

– Load the LIN Synch Field (0x55) in the SCIDR register to request Synch Field transmission

– Wait until the SCIDR is empty (TDRE bit set in the SCISR register)

– Load the LIN message Identifier in the SCIDR register to request Identifier transmission.

Obsolete Product(s) - Obsolete Product(s)

LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode) (cont'd)

10.7.9.9 Error due to LIN Synch measurement

The LIN Synch Field is measured over eight bit times.

This measurement is performed using a counter clocked by the CPU clock. The edge detections are performed using the CPU clock cycle.

This leads to a precision of 2 CPU clock cycles for the measurement which lasts 16*8*LDIV clock cycles.

Consequently, this error (D_{MEAS}) is equal to:

$2 / (128*LDIV_{MIN})$.

$LDIV_{MIN}$ corresponds to the minimum LIN prescaler content, leading to the maximum baud rate, taking into account the maximum deviation of +/-15%.

10.7.9.10 Error due to Baud Rate Quantization

The baud rate can be adjusted in steps of 1 / (16 * LDIV). The worst case occurs when the "real" baud rate is in the middle of the step.

This leads to a quantization error (D_{QUANT}) equal to $1 / (2*16*LDIV_{MIN})$.

10.7.9.11 Impact of Clock Deviation on Maximum Baud Rate

The choice of the nominal baud rate ($LDIV_{NOM}$) will influence both the quantization error (D_{QUANT}) and the measurement error (D_{MEAS}). The worst case occurs for $LDIV_{MIN}$.

Consequently, at a given CPU frequency, the maximum possible nominal baud rate (LPR_{MIN}) should be chosen with respect to the maximum tolerated deviation given by the equation:

$D_{TRA} + 2 / (128*LDIV_{MIN}) + 1 / (2*16*LDIV_{MIN}) + D_{REC} + D_{TCL} < 3.75\%$

Example:

A nominal baud rate of 20Kbits/s at T_{CPU} = 125ns (8 MHz) leads to $LDIV_{NOM}$ = 25d.

$LDIV_{MIN} = 25 - 0.15*25 = 21.25$

$D_{MEAS} = 2 / (128*LDIV_{MIN}) * 100 = 0.00073\%$

$D_{QUANT} = 1 / (2*16*LDIV_{MIN}) * 100 = 0.0015\%$

LIN Slave systems

For LIN Slave systems (the LINE and LSLV bits are set), receivers wake up by LIN Synch Break or LIN Identifier detection (depending on the LHDM bit).

Hot Plugging Feature for LIN Slave Nodes

In LIN Slave Mute Mode (the LINE, LSLV and RWU bits are set) it is possible to hot plug to a network during an ongoing communication flow. In this case the SCI monitors the bus on the RDI line until 11 consecutive dominant bits have been detected and discards all the other bits received.

Obsolete Product(s) Obsolete Product(s)

LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Master Only) (Cont'd)

10.8.4.7 Parity control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in Table 24.

Table 25. Frame Formats

M bit	PCE bit	SCI frame
0	0	| SB | 8 bit data | STB |
	1	| SB | 7-bit data | PB | STB |
1	0	| SB | 9-bit data | STB |
	1	| SB | 8-bit data PB | STB |

Legend:

SB: Start Bit

STB: Stop Bit

PB: Parity Bit

Note: In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Even parity: The parity bit is calculated to obtain an even number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 0 if even parity is selected (PS bit = 0).

Odd parity: The parity bit is calculated to obtain an odd number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an even number of "1s" if even parity is selected (PS = 0) or an odd number of "1s" if odd parity is selected (PS = 1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PIE is set in the SCICR1 register.

10.8.5 Low Power Modes

Mode	Description
WAIT	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
HALT	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

10.8.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE		
Received Data Ready to be Read	RDRF	RIE		
Overrun Error Detected	OR			
Idle Line Detected	IDLE	ILIE		
Parity Error	PE	PIE		

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Obsolete Product(s) - Obsolete Product(s)

LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Master Only) (Cont'd)

10.8.7 SCI Synchronous Transmission

The SCI transmitter allows the user to control a one way synchronous serial transmission. The SCLK pin is the output of the SCI transmitter clock. No clock pulses are sent to the SCLK pin during start bit and stop bit. Depending on the state of the LBCL bit in the SCICR3 register, clock pulses are or are not be generated during the last valid data bit (address mark). The CPOL bit in the SCICR3 register allows the user to select the clock polarity, and the CPHA bit in the SCICR3 register allows the user to select the phase of the external clock (see Figure 91, Figure 92 and Figure 93).

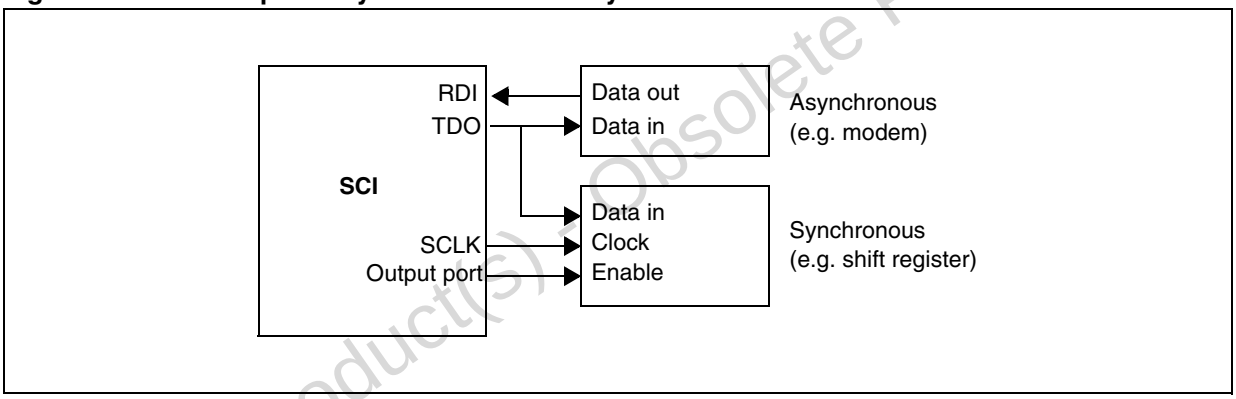
During idle, preamble and send break, the external SCLK clock is not activated.

These options allow the user to serially control peripherals which consist of shift registers, without losing any functions of the SCI transmitter which can still talk to other SCI receivers. These options do not affect the SCI receiver which is independent from the transmitter.

Note: The SCLK pin works in conjunction with the TDO pin. When the SCI transmitter is disabled (TE and RE = 0), the SCLK and TDO pins go into high impedance state.

Note: The LBCL, CPOL and CPHA bits have to be selected before enabling the transmitter to ensure that the clock pulses function correctly. These bits should not be changed while the transmitter is enabled.

Figure 91. SCI Example of Synchronous and Asynchronous Transmission



Obsolete Product(s) - Obsolete Product(s)

Figure 114. Workaround 2

```
Ld          a, CRFR
And         a,#3
Cp          a,#2            ; test FMP=2 ?
Jrne        _RELEASE        ; if not release


Btjf        CMSR,#5,_RELEASE ; test if reception on going.
                             ; if not release

Btjf        CDGR,#3,_RELEASE ; sample RX pin for 8 CAN bit time
Btjf        CDGR,#3,_RELEASE
Btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE
btjf        CDGR,#3,_RELEASE

_RELEASE:
bset        CRFR,#5
```

Obsolete Product(s) - Obsolete Product(s)

beCAN CONTROLLER (Cont'd)

CAN FILTER CONFIGURATION REG.1 (CFCR2)

All bits of this register are set and cleared by software.
Read / Write
Reset Value: 0000 0000 (00h)

7							0
0	FSC51	FSC50	FACT5	0	FSC41	FSC40	FACT4

Bit 7 = Reserved. Forced to 0 by hardware.

Bits 6:5 = **FSC5[1:0]** *Filter Scale Configuration*
These bits define the scale configuration of Filter 5.

Bit 4 = **FACT5** *Filter Active*
The software sets this bit to activate filter 5. To modify the Filter 5 registers (CF5R[0:7]) the FACT5 bit must be cleared.
0: Filter 5 is not active
1: Filter 5 is active

Bit 3 = Reserved. Forced to 0 by hardware.

Bits 2:1 = **FSC4[1:0]** *Filter Scale Configuration*
These bits define the scale configuration of Filter 4.

Bit 0 = **FACT4** *Filter Active*
The software sets this bit to activate Filter 4. To modify the Filter 4 registers (CF4R[0:7]), the FACT4 bit must be cleared.
0: Filter 4 is not active
1: Filter 4 is active

CAN FILTER MODE REGISTER (CFMR0)

All bits of this register are set and cleared by software.
Read / Write
Reset Value: 0000 0000 (00h)

7							0
FMH3	FML3	FMH2	FML2	FMH1	FML1	FMH0	FML0

Bit 7 = **FMH3** *Filter Mode High*
Mode of the high registers of Filter 3.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 6 = **FML3** *Filter Mode Low*
Mode of the low registers of Filter 3.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

Bit 5 = **FMH2** *Filter Mode High*
Mode of the high registers of Filter 2.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 4 = **FML2** *Filter Mode Low*
Mode of the low registers of Filter 2.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

Bit 3 = **FMH1** *Filter Mode High*
Mode of the high registers of Filter 1.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 2 = **FML1** *Filter Mode Low*
Mode of the low registers of filter 1.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

Bit 1 = **FMH0** *Filter Mode High*
Mode of the high registers of filter 0.
0: High registers are in mask mode
1: High registers are in identifier list mode

Bit 0 = **FML0** *Filter Mode Low*
Mode of the low registers of filter 0.
0: Low registers are in mask mode
1: Low registers are in identifier list mode

Obsolete Product(s) - Obsolete Product(s)

12.3.2 Operating Conditions with Low Voltage Detector (LVD)

Subject to general operating conditions for T_A.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)		4.0[1)]	4.2	4.5	V
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)		3.8	4.0	4.25[1)]	
$V_{hys(LVD)}$	LVD voltage threshold hysteresis[1)]	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$	150	200	250	mV
Vt_{POR}	V_{DD} rise time rate[1)]		6			µs/V
					100	ms/V
$t_{g(VDD)}$	V_{DD} glitches filtered (not detected) by LVD[1)]	Measured at $V_{IT-(LVD)}$			40	ns

Notes:

1. Data based on characterization results, not tested in production.

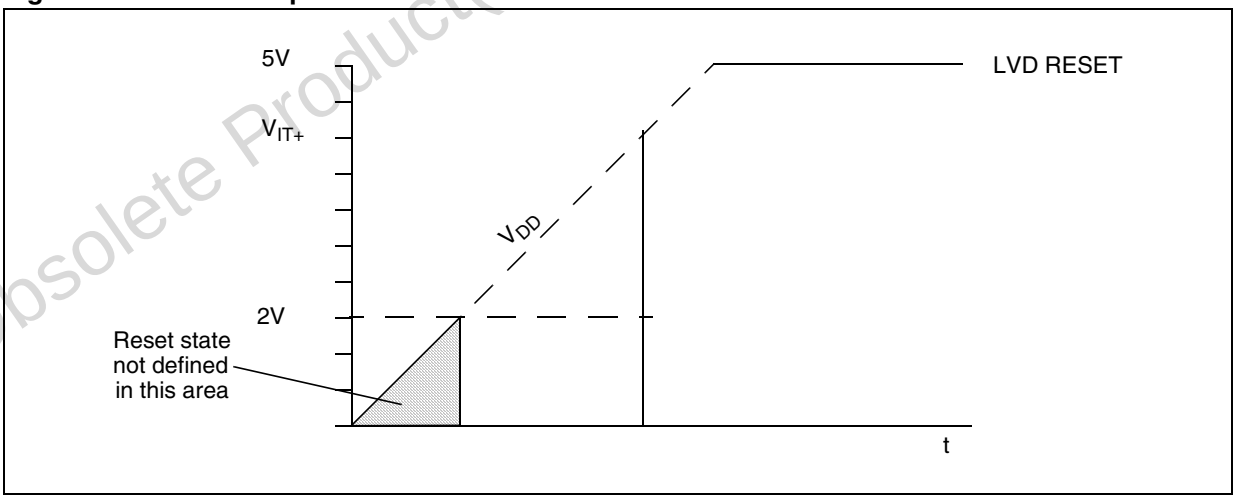
12.3.3 Auxiliary Voltage Detector (AVD) Thresholds

Subject to general operating conditions for T_A.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(AVD)}$	1⇒0 AVDF flag toggle threshold (V_{DD} rise)		4.4[1)]	4.6	4.9	V
$V_{IT-(AVD)}$	0⇒1 AVDF flag toggle threshold (V_{DD} fall)		4.2	4.4	4.65[1)]	
$V_{hys(AVD)}$	AVD voltage threshold hysteresis	$V_{IT+(AVD)}$-$V_{IT-(AVD)}$		250		mV
ΔV_{IT-}	Voltage drop between AVD flag set and LVD reset activated	$V_{IT-(AVD)}$-$V_{IT-(LVD)}$		450		

1. Data based on characterization results, not tested in production.

Figure 120. LVD Startup Behavior



Note: When the LVD is enabled, the MCU reaches its authorized operating voltage from a reset state. However, in some devices, the reset signal may be undefined until V_{DD} is approximately 2V. As a consequence, the I/Os may toggle when V_{DD} is below this voltage.

Because Flash write access is impossible below this voltage, the Flash memory contents will not be corrupted.

Obsolete Product(s) - Obsolete Product(s)

12.10 CONTROL PIN CHARACTERISTICS

12.10.1 Asynchronous $\overline{\text{RESET}}$ Pin

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage[1)]					0.3 x V_{DD}	V
V_{IH}	Input high level voltage[1)]			0.7 x V_{DD}			
V_{hys}	Schmitt trigger voltage hysteresis[2)]	V_{DD} = 5V			1.5		
V_{OL}	Output low level voltage[3)]	V_{DD} = 5V	I_{IO} = +5mA		0.68	0.95	
			I_{IO} = +2mA		0.28	0.45	
R_{ON}	Weak pull-up equivalent resistor[4)]	$V_{IN} = V_{SS}$		20	40	80	kΩ
$t_{w(RSTL)out}$	Generated reset pulse duration	Internal reset source			30		µs
$t_{h(RSTL)in}$	External reset pulse hold time[5)]			2.5			
$t_{g(RSTL)in}$	Filtered glitch duration[6)]				200		ns

Notes:

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels.
3. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
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 the RESET pin with a duration below $t_{h(RSTL)in}$ can be ignored.
5. The reset network (the resistor and two capacitors) protects the device against parasitic resets, especially in noisy environments.
6. Data guaranteed by design, not tested in production.

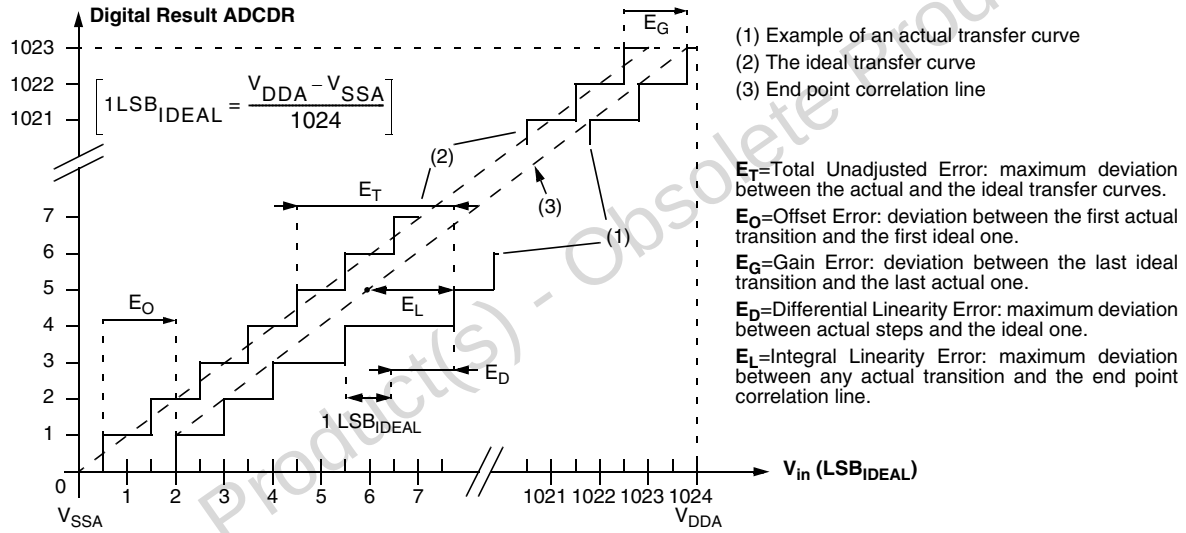
Obsolete Product(s) - Obsolete Product(s)

ADC CHARACTERISTICS (Cont'd)

ADC Accuracy with f_{CPU} = 8 MHz, f_{ADC} = 4 MHz R_{AIN} < 10kΩ, V_{DD} = 5V

Symbol	Parameter	Conditions	Typ	Max	Unit
$\|E_T\|$	Total unadjusted error[1)]		3.2	5	LSB
$\|E_O\|$	Offset error[1)]		1	4	
$\|E_G\|$	Gain Error[1)]		0.7		
$\|E_D\|$	Differential linearity error[1)]		1.5	2.3	
$\|E_L\|$	Integral linearity error[1)]		1.2	3.6	

Figure 145. ADC Accuracy Characteristics



Notes:

1. Data based on characterization results, not tested in production. ADC Accuracy vs. Negative Injection Current: Injecting negative current on any of the standard (non-robust) analog input pins should be avoided as this significantly reduces the accuracy of the conversion being performed on another analog input. It is recommended to add a Schottky diode (pin to ground) to standard analog pins which may potentially inject negative current. The effect of negative injection current on robust pins is specified in Section 12.9.

Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in Section 12.9 does not affect the ADC accuracy.

Obsolete Product(s) - Obsolete Product(s)

DEVICE CONFIGURATION AND ORDER INFORMATION (cont'd)

14.2 TRANSFER OF CUSTOMER CODE

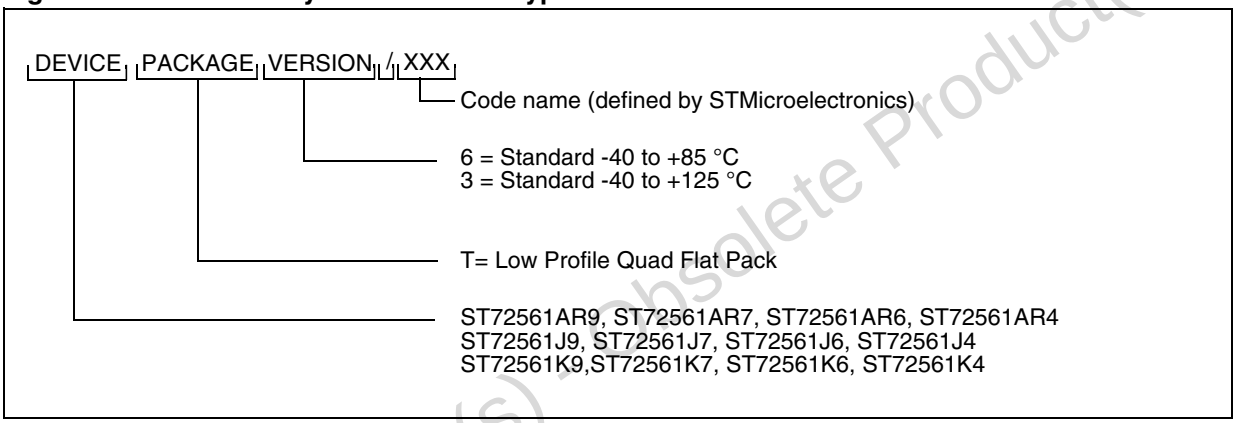
Customer code is made up of the ROM/FASTROM contents and the list of the selected options (if any). The ROM/FASTROM contents are to be sent on diskette, or by electronic means, with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended.

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Figure 150. ROM Factory Coded Device Types



Obsolete Product(s) - Obsolete Product(s)