



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	16MHz
Connectivity	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	22
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	32-SDIP (0.400", 10.16mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f262g2b6

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

5 CENTRAL PROCESSING UNIT

5.1 INTRODUCTION

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

5.2 MAIN FEATURES

- Enable executing 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- Low power HALT and WAIT modes
- Priority maskable hardware interrupts
- Non-maskable software/hardware interrupts

5.3 CPU REGISTERS

The 6 CPU registers shown in Figure 7 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures.

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).



Figure 7. CPU Registers

5/

CENTRAL PROCESSING UNIT (Cont'd)

Condition Code Register (CC)

Read/Write

Reset Value: 111x1xxx



The 8-bit Condition Code register contains the interrupt masks and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Arithmetic Management Bits

Bit $4 = \mathbf{H}$ Half carry.

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 2 = N Negative.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It's a copy of the result 7^{th} bit.

0: The result of the last operation is positive or null.

1: The result of the last operation is negative

(i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** Zero.

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

0: The result of the last operation is different from zero.

1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = C Carry/borrow.

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

Interrupt Management Bits

Bit 5,3 = 11, 10 Interrupt

The combination of the I1 and I0 bits gives the current interrupt software priority.

Interrupt Software Priority	11	10
Level 0 (main)	1	0
Level 1	0	1
Level 2	0	0
Level 3 (= interrupt disable)	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (IxSPR). They can be also set/ cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See the interrupt management chapter for more details.



CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 7Fh



The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 8).

Since the stack is 128 bytes deep, the 8 most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

Figure 8. Stack Manipulation Example

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 8

- When an interrupt is received, the SP is decremented and the context is pushed on the stack.
- On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.



6 SUPPLY, RESET AND CLOCK MANAGEMENT

The device includes a range of utility features for securing the application in critical situations (for example in case of a power brown-out), and reducing the number of external components. An overview is shown in Figure 10.

For more details, refer to dedicated parametric section.

Main Features

- Optional PLL for multiplying the frequency by 2 (not to be used with internal RC oscillator)
- Reset Sequence Manager (RSM)
- Multi-Oscillator Clock Management (MO)
 - 4 Crystal/Ceramic resonator oscillators
 - 1 Internal RC oscillator
- System Integrity Management (SI)
 - Main supply Low Voltage Detector (LVD)
 - Auxiliary Voltage Detector (AVD) with interrupt capability for monitoring the main supply

6.1 PHASE LOCKED LOOP

If the clock frequency input to the PLL is in the 2 to 4 MHz range, the PLL can be used to multiply the frequency by two to obtain an f_{OSC2} of 4 to 8 MHz.

The PLL is enabled by option byte. If the PLL is disabled, then $f_{OSC2} = f_{OSC}/2$.

Caution: The PLL is not recommended for applications where timing accuracy is required. See "PLL Characteristics" on page 139.

Figure 9. PLL Block Diagram



Figure 10. Clock, Reset and Supply Block Diagram



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} and V_{IT} reference value and the V_{DD} main supply. The V_{IT} reference value for falling voltage is lower than the V_{IT} 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 (VDF) in the SICSR register. This bit is read only.

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

6.4.2.1 Monitoring the V_{DD} Main Supply

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see Section 15.1 on page 162).

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

Figure 15. Using the AVD to Monitor V_{DD}

57/

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 15.

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 $t_{\rm rv}$ is less than 256 or 4096 CPU cycles (depending on the reset delay selected by option byte), no AVD interrupt will be generated when $V_{\rm IT+(AVD)}$ 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 2 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 $V_{\text{IT}+(\text{AVD})}$ threshold is reached then only one AVD interrupt will occur.



I/O PORTS (Cont'd)

Table 8. I/O Configurations



Notes:

- 1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.
- 2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.
- 3. For true open drain, these elements are not implemented.



MISCELLANEOUS REGISTERS (Cont'd)

10.3 MISCELLANEOUS REGISTER DESCRIPTION

MISCELLANEOUS REGISTER 1 (MISCR1)

Read/Write

Reset Value: 0000 0000 (00h)

7

0

	l.		l.				l.
IS11	IS10	мсо	IS01	IS00	CP1	CP0	SMS

Bits 7:6 = IS1[1:0] ei1 sensitivity

The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the ei1 external interrupts. These two bits can be written only when the I[1:0] bits in the CC register are set to 1 (interrupt masked).

ei1: Port B (C optional)

External Interrupt Sensitivity	IS11	IS10
Falling edge & low level	0	0
Rising edge only	0	1
Falling edge only	1	0
Rising and falling edge	1	1

Bit 5 = MCO Main clock out selection

This bit enables the MCO alternate function on the PC2 I/O port. It is set and cleared by software. 0: MCO alternate function disabled (I/O pin free for

- general-purpose I/O)
- 1: MCO alternate function enabled (f_{CPU} on I/O port)

Bits 4:3 = ISO[1:0] ei0 sensitivity

The interrupt sensitivity, defined using the IS0[1:0] bits, is applied to the ei0 external interrupts. These two bits can be written only when the I[1:0] bits in-the CC register are set to 1 (interrupt masked).

ei0: Port A (C optional)

External Interrupt Sensitivity	IS01	IS00
Falling edge & low level	0	0
Rising edge only	0	1
Falling edge only	1	0
Rising and falling edge	1	1

Bits 2:1 = CP[1:0] CPU clock prescaler

These bits select the CPU clock prescaler which is applied in the various slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software

f _{CPU} in SLOW mode	CP1	CP0
f _{OSC2} / 2	0	0
f _{OSC2} / 4	1	0
f _{OSC2} / 8	0	1
f _{OSC2} / 16	1	1

Bit 0 = **SMS** *Slow mode select*

This bit is set and cleared by software.

0: Normal mode. $f_{CPU} = f_{OSC2}$

1: Slow mode. f_{CPU} is given by CP1, CP0

See low power consumption mode and MCC chapters for more details.



WATCHDOG TIMER (Cont'd)

11.1.5 Low Power Modes

Mode	Description		
SLOW	No effect on	Watchdog.	
WAIT	No effect on	Watchdog.	
	OIE bit in	WDGHALT bit	
	MCCSR	in Option	
	register	Byte	
	0	0	No Watchdog reset is generated. The MCU enters Halt mode. The Watch- dog counter is decremented once and then stops counting and is no longer able to generate a watchdog reset until the MCU receives an external inter- rupt or a reset.
HALT			If an external interrupt is received, the Watchdog restarts counting after 256 or 4096 CPU clocks. If a reset is generated, the Watchdog is disabled (reset state) unless Hardware Watchdog is selected by option byte. For application recommendations see Section 11.1.7 below.
	0	1	A reset is generated.
	1	x	No reset is generated. The MCU enters Active Halt mode. The Watchdog counter is not decremented. It stop counting. When the MCU receives an oscillator interrupt or external interrupt, the Watchdog restarts counting immediately. When the MCU receives a reset the Watchdog restarts counting after 256 or 4096 CPU clocks.

11.1.6 Hardware Watchdog Option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the WDGCR is not used. Refer to the Option Byte description.

11.1.7 Using Halt Mode with the WDG (WDGHALT option)

The following recommendation applies if Halt mode is used when the watchdog is enabled.

 Before executing the HALT instruction, refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.

11.1.8 Interrupts

None.

57/

11.1.9 Register Description CONTROL REGISTER (WDGCR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	Т6	T5	T4	ТЗ	T2	T1	ТО

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.

Bit 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).

16-BIT TIMER (Cont'd)

Figure 35. Timer Block Diagram



16-BIT TIMER (Cont'd)

11.3.3.5 One Pulse Mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure:

To use one pulse mode:

- 1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in the opposite column).
- 2. Select the following in the CR1 register:
 - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
 - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
 - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
- 3. Select the following in the CR2 register:
 - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
 - Set the OPM bit.
 - Select the timer clock CC[1:0] (see Table 14 Clock Control Bits).



When a valid event occurs on the ICAP1 pin, the counter value is loaded in the ICR1 register. The counter is then initialized to FFFCh, the OLVL2 bit is output on the OCMP1 pin and the ICF1 bit is set.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (i.e. clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.

2. An access (read or write) to the ICiLR register.

The OC1R register value required for a specific timing application can be calculated using the following formula:

Where:

t = Pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits, see Table 14 Clock Control Bits)

If the timer clock is an external clock the formula is:

Where:

t = Pulse period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin, (See Figure 44).

Notes:

- 1. The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.
- 2. When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.
- 3. If OLVL1=OLVL2 a continuous signal will be seen on the OCMP1 pin.
- 4. The ICAP1 pin can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.
- 5. When one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

47/

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



SERIAL PERIPHERAL INTERFACE (Cont'd)

11.4.3.2 Slave Select Management

As an alternative to using the \overline{SS} pin to control the Slave Select signal, the application can choose to manage the Slave Select signal by software. This is configured by the SSM bit in the SPICSR register (see Figure 49)

In software management, the external SS pin is free for other application uses and the internal SS signal level is driven by writing to the SSI bit in the SPICSR register.

In Master mode:

- SS internal must be held high continuously

In Slave Mode:

There are two cases depending on the data/clock timing relationship (see Figure 48):

- If CPHA=1 (data latched on 2nd clock edge):
 - \overline{SS} internal must be held low during the entire transmission. This implies that in single slave applications the \overline{SS} pin either can be tied to V_{SS} , or made free for standard I/O by managing the \overline{SS} function by software (SSM= 1 and SSI=0 in the in the SPICSR register)

If CPHA=0 (data latched on 1st clock edge):

 - SS internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If SS is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see Section 11.4.5.3).



Figure 49. Hardware/Software Slave Select Management



SERIAL PERIPHERAL INTERFACE (Cont'd)

11.4.3.3 Master Mode Operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL=1 or pulling down SCK if CPOL=0).

To operate the SPI in master mode, perform the following steps in order (if the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account):

1. Write to the SPICR register:

- Select the clock frequency by configuring the SPR[2:0] bits.
- Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. Figure 50 shows the four possible configurations.
 Note: The slave must have the same CPOL and CPHA settings as the master.
- 2. Write to the SPICSR register:
 - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the SS pin high for the complete byte transmit sequence.
- 3. Write to the SPICR register:
 - Set the MSTR and SPE bits
 <u>Note</u>: MSTR and SPE bits remain set only if SS is high).

The transmit sequence begins when software writes a byte in the SPIDR register.

11.4.3.4 Master Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

- 1. An access to the SPICSR register while the SPIF bit is set
- 2. A read to the SPIDR register.

5/

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

11.4.3.5 Slave Mode Operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

- 1. Write to the SPICSR register to perform the following actions:
 - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see Figure 50).
 Note: The slave must have the same CPOL and CPHA settings as the master.
 - Manage the SS pin as described in Section 11.4.3.2 and Figure 48. If CPHA=1 SS must be held low continuously. If CPHA=0 SS must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
- 2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

11.4.3.6 Slave Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set.

2. A write or a read to the SPIDR register.

Notes: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see Section 11.4.5.2).

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

11.5.4.7 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 18.

Table 18. Frame Form	nats
----------------------	------

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

Ex: data=00110101; 4 bits set => parity bit will be 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.

Ex: data=00110101; 4 bits set => parity bit will be 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.

11.5.4.8 SCI Clock Tolerance

During reception, each bit is sampled 16 times. The majority of the 8th, 9th and 10th samples is considered as the bit value. For a valid bit detection, all the three samples should have the same value otherwise the noise flag (NF) is set. For example: if the 8th, 9th and 10th samples are 0, 1 and 1 respectively, then the bit value will be "1", but the Noise Flag bit is be set because the three samples values are not the same.

Consequently, the bit length must be long enough so that the 8th, 9th and 10th samples have the desired bit value. This means the clock frequency should not vary more than 6/16 (37.5%) within one bit. The sampling clock is resynchronized at each start bit, so that when receiving 10 bits (one start bit, 1 data byte, 1 stop bit), the clock deviation must not exceed 3.75%.

Note: The internal sampling clock of the microcontroller samples the pin value on every falling edge. Therefore, the internal sampling clock and the time the application expects the sampling to take place may be out of sync. For example: If the baud rate is 15.625 kbaud (bit length is 64µs), then the 8th, 9th and 10th samples will be at 28µs, 32µs & 36µs respectively (the first sample starting ideally at 0µs). But if the falling edge of the internal clock occurs just before the pin value changes, the samples would then be out of sync by ~4us. This means the entire bit length must be at least 40µs (36µs for the 10th sample + 4µs for synchronization with the internal sampling clock).

I²C BUS INTERFACE (Cont'd) 11.6.7 Register Description

I²C CONTROL REGISTER (CR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	PE	ENGC	START	ACK	STOP	ITE

Bit 7:6 = Reserved. Forced to 0 by hardware.

Bit 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 I²C interface, write the CR register TWICE with PE=1 as the first write only activates the interface (only PE is set).

Bit 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.

Bit 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

Bit 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

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

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.

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 60 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 59) is detected.



11.7 10-BIT A/D CONVERTER (ADC)

11.7.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has 6 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from 6 different sources.

The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

11.7.2 Main Features

- 10-bit conversion
- 6 channels with multiplexed input
- Linear successive approximation

Figure 61. ADC Block Diagram

- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in Figure 61.

11.7.3 Functional Description

11.7.3.1 Analog Power Supply

 V_{DDA} and V_{SSA} are the high and low level reference voltage pins. In some devices (refer to device pin out description) they are internally connected to the V_{DD} and V_{SS} pins.

Conversion accuracy may therefore be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

457/



INSTRUCTION SET OVERVIEW (Cont'd)

12.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two sub-modes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table25. InstructionsSupportingDirect,Indexed,IndirectandIndirectIndexedAddressing ModesIndirectIndexedIndirect

Long and Short Instructions	Function
LD	Load
СР	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Additions/Sub- stractions operations
BCP	Bit Compare

Short Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Opera- tions
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Opera- tions
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

12.1.7 Relative mode (Direct, Indirect)

This addressing mode is used to modify the PC register value, by adding an 8-bit signed offset to it.

Available Relative Direct/Indirect Instructions	Function	
JRxx	Conditional Jump	
CALLR	Call Relative	

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

The offset is defined in memory, which address follows the opcode.

13.9 CONTROL PIN CHARACTERISTICS

13.9.1 Asynchronous RESET Pin

 $T_A = -40$ to $+85^{\circ}C$ unless otherwise specified

Symbol	Parameter	Conditions		Min	Тур	Max	Unit
V _{IL}	Input low level voltage			V _{ss} - 0.3		$0.3 \mathrm{xV}_{\mathrm{DD}}$	V
V _{IH}	Input high level voltage			$0.7 \mathrm{xV}_{\mathrm{DD}}$		V _{DD} + 0.3	v
V _{hys}	Schmitt trigger voltage hysteresis ¹⁾				2.5		V
V	Output low lovel veltage 2^{2}	V _5V	I _{IO} =+5mA		0.68	0.95	v
VOL	Output low level voltage	VDD-3V	I _{IO} =+2mA		0.28	0.45	
P	Pull-up oquivalant resistor	V _{DD} =5V V _{DD} =3V		20	40	80	kO
NON					85		N2 2
tw(RSTL)out	Generated reset pulse duration	Internal re	eset sources		30		μS
t _{h(RSTL)in}	External reset pulse hold time 3)			20			μS
t _{g(RSTL)in}	Filtered glitch duration				200		ns

Figure 90. Typical I_{PU} on RESET pin



Notes:

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 13.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.

3. To guarantee the reset of the device, a minimum pulse has to be applied to the $\overrightarrow{\text{RESET}}$ pin. All short pulses applied on RESET pin with a duration below $t_{h(\text{RSTL})in}$ can be ignored.



COMMUNICATION INTERFACE CHARACTERISTICS (Cont'd)



Figure 94. SPI Slave Timing Diagram with CPHA=1¹⁾

Figure 95. SPI Master Timing Diagram 1)



Notes:

1. Measurement points are done at CMOS levels: $0.3 x V_{\text{DD}}$ and $0.7 x V_{\text{DD}}.$

2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

لركم

15.3.2 PACKAGE/SOCKET FOOTPRINT PROPOSAL

Table 30. Suggested List of SDIP32 Socket Types

Package / Probe	Adaptor / Socket Reference		Same Footprint	Socket Type
SDIP32 EMU PROBE	TEXTOOL	232-1291-00	х	Textool

Table 31. Suggested List of SO28 Socket Types

Package / Probe	Adaptor / Socket Reference	Same Footprint	Socket Type
SO28	YAMAICHI IC51-0282-334-1		Clamshell
EMU PROBE	Adapter from SO28 to SDIP32 footprint (delivered with emulator)	Х	SMD to SDIP

Table 32. Suggested LFBGA Socket Type

57

Package	Socket Reference	
LFBGA 6 X6	ENPLAS OTB-36(144)-0.8-04	ENPLAS OTB-36(144)-0.8-04