

**MAKERERE UNIVERSITY**  
**COLLEGE OF COMPUTING & INFORMATION**  
**SCIENCES**  
**SCHOOL OF COMPUTING AND INFORMATICS**  
**TECHNOLOGY**  
**END OF SEMESTER I EXAMINATION 2022/2023**

PROGRAMME: COMPUTER SCIENCE

YEAR OF STUDY: II

COURSE NAME: EMBEDDED REAL-TIME SYSTEMS

COURSE CODE: CSC 2118

DATE: 27th January 2023 TIME: 08:00AM - 11:00AM

**EXAMINATION INSTRUCTIONS**

- a) ATTEMPT ALL QUESTIONS IN SECTION A (40 MARKS)
- b) ATTEMPT THREE (03) QUESTIONS IN SECTION B (60 MARKS)
- c) DO NOT OPEN THIS EXAM UNTIL YOU ARE TOLD TO DO SO
- d) ATTEMPT EACH QUESTION IN SECTION B ON A NEW PAGE
- e) ALL ROUGH WORK SHOULD BE IN YOUR ANSWER BOOKLET

## SECTION A

1. What is an embedded system, and what are components of an embedded system? **(2 marks)**
2. What is a micro-controller? **(2 marks)**
3. What is a fundamental difference between micro-controllers and microprocessors? **(2 marks)**
4. Give 3 reasons for using a microcontroller instead of general-purpose computer **(3 marks)**
5. What is security in an embedded system? **(2 marks)**
6. What is safety in an embedded system? **(2 marks)**
7. How does security affect safety of an embedded system? **(3 marks)**
8. What's the basic difference between RISC and CISC architecture? **(2 marks)**
9. State 3 challenges that are often faced in designing embedded systems **(8 marks)**
10. What are the 5 steps of the embedded design methodology? **(5 marks)**
11. Briefly describe the distinction between requirements and specification of an embedded system **(2 marks)**
12. Briefly describe the distinction between specification and architecture of an embedded system **(2 marks)**
13. What is quantisation and when is it used in embedded systems? **(2 marks)**
14. Given an analogue signal varying from 0 to 10V, how many bits of ADC would we need to achieve a 0.01V (2 decimal points) resolution? **(3 marks)**

## SECTION B

1. Analyze the code snippet below to answer the follow-on questions

```
1. int x;  
2. short *y;  
3. void functionX (void);  
4. void functionX (void)  
5. {  
6.     unsigned char i, j;  
7.     for (i = 0; i < 10; i++)  
8.     {  
9.         for (j = 0; j < 5; j++)  
10.        {  
11.            int k = 10;  
12.            x += k + j;  
13.        }  
14.        *y = x*j;  
15.    }  
16. }
```

- a. Explain what the code snippet does. **(3 marks)**
  - b. Use a table to make a trace of the values of the variables up to iteration i = 3. **(5 marks)**
  - c. What are pointers in C/C++ and what are they often used for in Embedded systems **(2 marks)**
  - d. Explain the scope of each variable **(2.5 marks)**
  - e. What is the difference between the variables of type **int** and **short**. How many bits are commonly used for each type and what is the maximum unsigned value each can represent **(2.5 marks)**
  - f. Re-write the code to use while loops instead of for loops **(5 marks)**
2. Push buttons are used in a variety of embedded applications to trigger or stop an action when someone or something presses the switch. However, due to the mechanical nature of the switches, a single push on the switch might be interpreted as multiple presses. Using the Atmega 328p and the Appendix:
    - a. Write a program in C/C++ that monitors and debounces an input switch. **(10 marks)**
    - b. Modify your code in 2.a such that a DC motor turns on when a user presses the switch the first time, and turns off when the user presses the switch a second time. **(10 marks)**

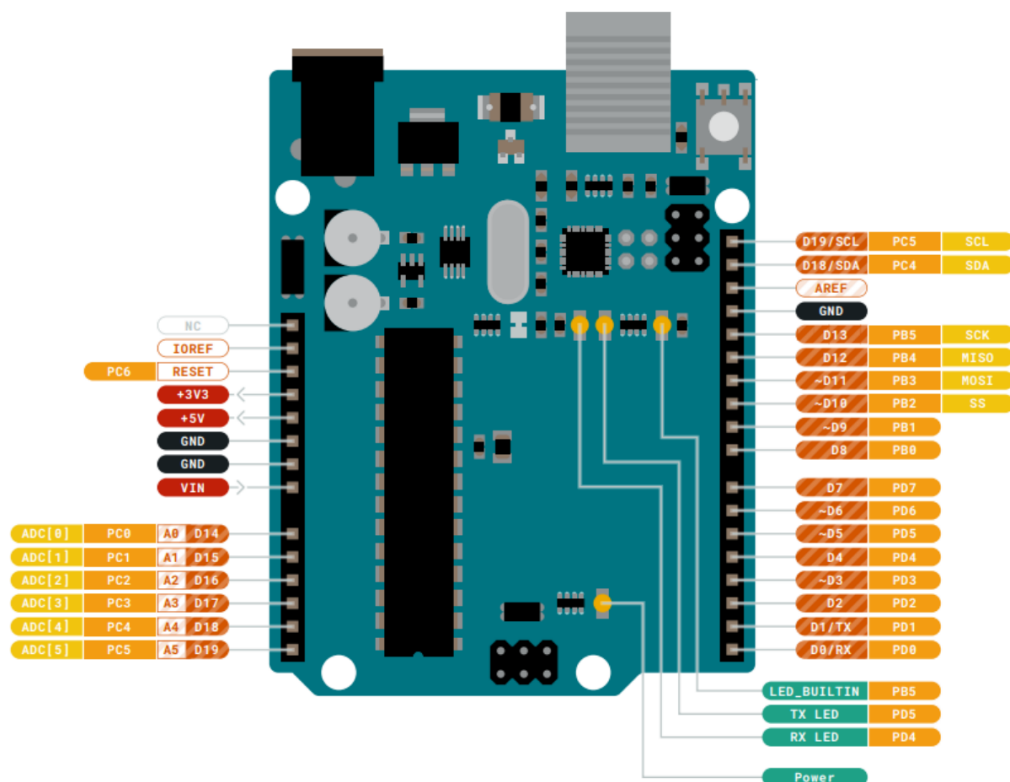
3. Time delays are regularly used in embedded systems to determine when a given operation starts or stops executing. Using the Atmega 328p and Appendix:
  - a. Use Timer interrupts and counter 1 to make an LED connected on a PIN of your choice on PORTB to blink on and off. Use the sample code below as a guide to help you write the complete code in C/C++ needed. **(10 marks)**

```
ISR(TIMER1_COMPA_vect)
{
    //code
}

int main()
{
    //code
    TIMSK1 = 0b00000010;
    sei(); //enable global interrupts
}
```

- b. What is the clock period when the clock select pins are set to 101 **(4 marks)**
  - c. Re-write this program to use an already existing time delay library `_delay_ms()` and a polling mechanism **(6 marks)**
4. As a growing embedded start-up company, you have been approached by a local government to help them develop an early warning system for wildfires. The client wants portable wireless smoke detectors that are the size of a smartphone. They want to deploy the smoke detectors across a forest conservation that spans 200 hectares. The devices need to be easily placed on tree trunks by forest rangers as they walk through the forest. Due to forest canopy cover, the devices cannot be solar powered, and neither can they have a GPS module to determine their location in real time, so it has been proposed that a sector identification number (SiD) representing different sections of the forest, shall be manually programmed into each device by the user (forest ranger) at time of deployment. Each unit should cost no more \$10 and will use LoRaWAN to transmit its data. Furthermore, each device will be battery powered.
  - a. Write up a requirements document for this embedded system **(5 marks)**
  - b. A hectare is approximately 100m length, and 100m width.
    - i. If each smoke detector can detect smoke in an area of 10000m<sup>2</sup> how many smoke detectors would be needed to cover the whole forest. **(2marks)**
    - ii. How many bits would be needed to store the SiD and in which memory would this value best be stored considering that batteries might have to be swapped occasionally. **(2 marks)**
  - c. Propose how the SiD can be input into a device, considering the process must be as user friendly as possible keeping in mind the device constraints. Augment your answer with pseudo code or flow diagram. **(5 marks)**
  - d. Present a high-level Architecture design using block diagrams **(6 marks)**

## Appendix A



### Atmega 328P pin out

## Appendix B

### 108 6. GENERAL-PURPOSE INPUT/OUTPUT

#### 6.4.1 PORTB - THE PORT B DATA REGISTER

Bit	7	6	5	4	3	2	1	0
0x25	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

- PORTB7-0: GPIO data value stored in bit  $n$ .

#### 6.4.2 DDRB - THE PORT B DATA DIRECTION REGISTER

Bit	7	6	5	4	3	2	1	0
0x24	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

- DDRB7-0: selects the direction of pin  $n$ . If  $DDRBn$  is written '1', then  $PORTBn$  is configured as an output pin. If  $DDRBn$  is written '0', then  $PORTBn$  is configured as an input pin.

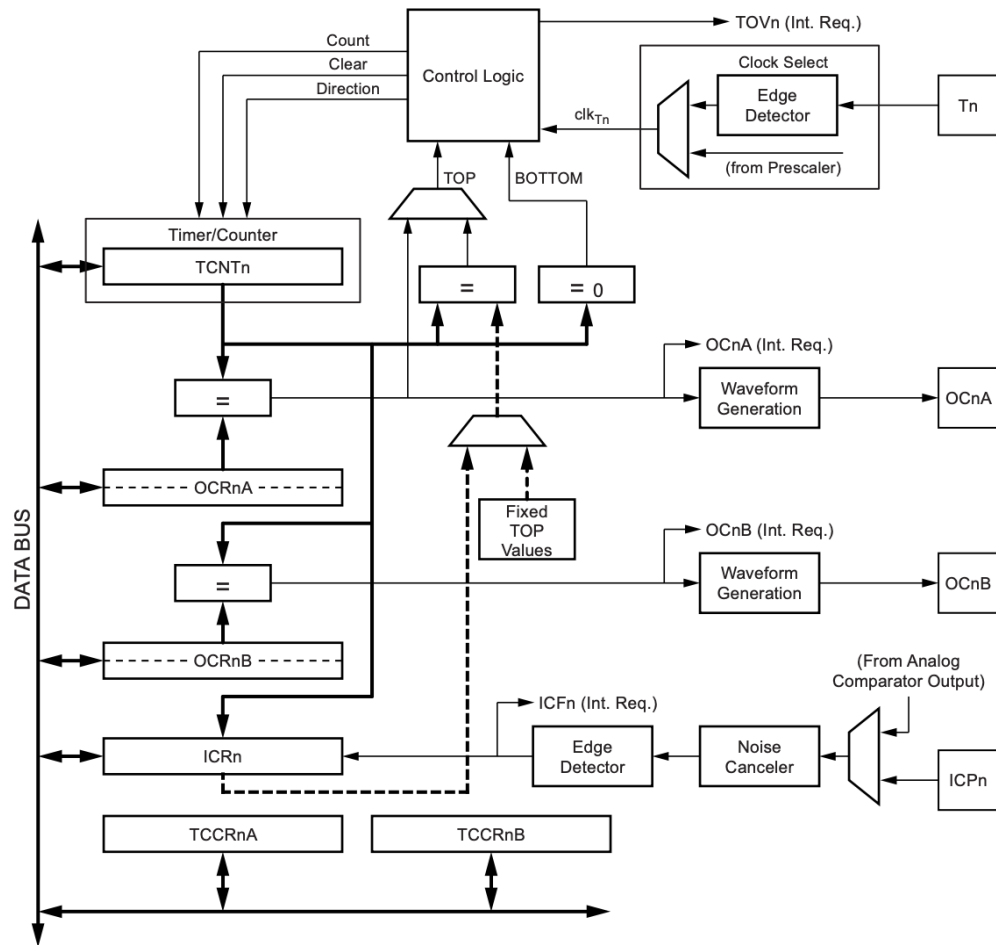
#### 6.4.3 PINB - THE PORT B INPUT PINS ADDRESS

Bit	7	6	5	4	3	2	1	0
0x23	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Read/Write	R	R	R	R	R	R	R	R
Default	-	-	-	-	-	-	-	-

- PINB7-0: logic value present on external pin  $n$ .

## Appendix C

Figure 15-1. 16-bit Timer/Counter Block Diagram<sup>(1)</sup>



## 15.11 Register Description

### 15.11.1 TCCR1A – Timer/Counter1 Control Register A

Bit (0x80)	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A**

- **Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B**

The COM1A1:0 and COM1B1:0 control the output compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. [Table 15-2](#) shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a normal or a CTC mode (non-PWM).

**Table 15-2. Compare Output Mode, non-PWM**

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on compare match.
1	0	Clear OC1A/OC1B on compare match (set output to low level).
1	1	Set OC1A/OC1B on compare match (set output to high level).

[Table 15-3](#) shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

**Table 15-3. Compare Output Mode, Fast PWM<sup>(1)</sup>**

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on compare match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on compare match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on compare match, clear OC1A/OC1B at BOTTOM (inverting mode)

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See [Section 15.9.3 "Fast PWM Mode" on page 101](#) for more details.



Table 15-4 shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the phase correct or the phase and frequency correct, PWM mode.

**Table 15-4. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM<sup>(1)</sup>**

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 9 or 11: Toggle OC1A on compare match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on compare match when up-counting. Set OC1A/OC1B on compare match when down counting.
1	1	Set OC1A/OC1B on compare match when up-counting. Clear OC1A/OC1B on compare match when down counting.

Note: 1. A special case occurs when OCR1A/OCR1B equals TOP and COM1A1/COM1B1 is set. See [Section 15.9.4 "Phase Correct PWM Mode" on page 103](#) for more details.

• **Bit 1:0 – WGM11:0: Waveform Generation Mode**

Combined with the WGM13:2 bits found in the TCCR1B register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see [Table 15-5](#). Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), clear timer on compare match (CTC) mode, and three types of pulse width modulation (PWM) modes. See ([Section 15.9 "Modes of Operation" on page 100](#)).

**Table 15-5. Waveform Generation Mode Bit Description<sup>(1)</sup>**

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, phase correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, phase correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, phase correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, phase and frequency correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, phase and frequency correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, phase correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, phase correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Note: 1. The CTC1 and PWM11:0 bit definition names are obsolete. Use the WGM12:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

### 15.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNC1: Input Capture Noise Canceler**

Setting this bit (to one) activates the input capture noise canceler. When the noise canceler is activated, the input from the input capture pin (ICP1) is filtered. The filter function requires four successive equal valued samples of the ICP1 pin for changing its output. The input capture is therefore delayed by four oscillator cycles when the noise canceler is enabled.

- **Bit 6 – ICES1: Input Capture Edge Select**

This bit selects which edge on the input capture pin (ICP1) that is used to trigger a capture event. When the ICES1 bit is written to zero, a falling (negative) edge is used as trigger, and when the ICES1 bit is written to one, a rising (positive) edge will trigger the capture.

When a capture is triggered according to the ICES1 setting, the counter value is copied into the input capture register (ICR1). The event will also set the input capture flag (ICF1), and this can be used to cause an input capture interrupt, if this interrupt is enabled.

When the ICR1 is used as TOP value (see description of the WGM13:0 bits located in the TCCR1A and the TCCR1B register), the ICP1 is disconnected and consequently the input capture function is disabled.

- **Bit 5 – Reserved Bit**

This bit is reserved for future use. For ensuring compatibility with future devices, this bit must be written to zero when TCCR1B is written.

- **Bit 4:3 – WGM13:2: Waveform Generation Mode**

See TCCR1A register description.

- **Bit 2:0 – CS12:0: Clock Select**

The three clock select bits select the clock source to be used by the Timer/Counter, see [Figure 15-10 on page 106](#) and [Figure 15-11 on page 106](#).

**Table 15-6. Clock Select Bit Description**

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{\text{I/O}}/1$ (no prescaling)
0	1	0	$\text{clk}_{\text{I/O}}/8$ (from prescaler)
0	1	1	$\text{clk}_{\text{I/O}}/64$ (from prescaler)
1	0	0	$\text{clk}_{\text{I/O}}/256$ (from prescaler)
1	0	1	$\text{clk}_{\text{I/O}}/1024$ (from prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter1, transitions on the T1 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

### 15.11.3 TCCR1C – Timer/Counter1 Control Register C

Bit	7	6	5	4	3	2	1	0	
(0x82)	FOC1A	FOC1B	–	–	–	–	–	–	TCCR1C
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC1A: Force Output Compare for Channel A**

- **Bit 6 – FOC1B: Force Output Compare for Channel B**

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the waveform generation unit. The OC1A/OC1B output is changed according to its COM1x1:0 bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1x1:0 bits that determine the effect of the forced compare.

A FOC1A/FOC1B strobe will not generate any interrupt nor will it clear the timer in clear timer on compare match (CTC) mode using OCR1A as TOP. The FOC1A/FOC1B bits are always read as zero.

### 15.11.4 TCNT1H and TCNT1L – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The two Timer/Counter I/O locations (TCNT1H and TCNT1L, combined TCNT1) give direct access, both for read and for write operations, to the Timer/Counter unit 16-bit counter. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See [Section 15.3 "Accessing 16-bit Registers" on page 91](#).

Modifying the counter (TCNT1) while the counter is running introduces a risk of missing a compare match between TCNT1 and one of the OCR1x registers.

Writing to the TCNT1 register blocks (removes) the compare match on the following timer clock for all compare units.

### 15.11.5 OCR1AH and OCR1AL – Output Compare Register 1 A

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

### 15.11.6 OCR1BH and OCR1BL – Output Compare Register 1 B

Bit	7	6	5	4	3	2	1	0	
(0x8B)	OCR1B[15:8]								OCR1BH
(0x8A)	OCR1B[7:0]								OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The output compare registers contain a 16-bit value that is continuously compared with the counter value (TCNT1). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC1x pin.

The output compare registers are 16-bit in size. To ensure that both the high and low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See [Section 15.3 "Accessing 16-bit Registers" on page 91](#).

### 15.11.7 ICR1H and ICR1L – Input Capture Register 1

Bit	7	6	5	4	3	2	1	0	
(0x87)	ICR1[15:8]								ICR1H
(0x86)	ICR1[7:0]								ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The input capture is updated with the counter (TCNT1) value each time an event occurs on the ICP1 pin (or optionally on the analog comparator output for Timer/Counter1). The input capture can be used for defining the counter TOP value.

The input capture register is 16-bit in size. To ensure that both the high and low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See [Section 15.3 "Accessing 16-bit Registers" on page 91](#).

### 15.11.8 TIMSK1 – Timer/Counter1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 5 – ICIE1: Timer/Counter1, Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 input capture interrupt is enabled. The corresponding interrupt vector (see [Section 11. "Interrupts" on page 49](#)) is executed when the ICF1 flag, located in TIFR1, is set.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the Atmel ATmega328P, and will always read as zero.

- **Bit 2 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 output compare B match interrupt is enabled. The corresponding interrupt vector (see [Section 11. "Interrupts" on page 49](#)) is executed when the OCF1B flag, located in TIFR1, is set.

- **Bit 1 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 output compare A match interrupt is enabled. The corresponding interrupt vector (see [Section 11. "Interrupts" on page 49](#)) is executed when the OCF1A flag, located in TIFR1, is set.

- **Bit 0 – TOIE1: Timer/Counter1, Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 overflow interrupt is enabled. The corresponding interrupt vector (see [Section 11. "Interrupts" on page 49](#)) is executed when the TOV1 flag, located in TIFR1, is set.

### 15.11.9 TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATmega328P, and will always read as zero.

- **Bit 5 – ICF1: Timer/Counter1, Input Capture Flag**

This flag is set when a capture event occurs on the ICP1 pin. When the input capture register (ICR1) is set by the WGM13:0 to be used as the TOP value, the ICF1 flag is set when the counter reaches the TOP value.

ICF1 is automatically cleared when the input capture interrupt vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- **Bit 4, 3 – Res: Reserved Bits**

These bits are unused bits in the Atmel ATmega328P, and will always read as zero.

- **Bit 2 – OCF1B: Timer/Counter1, Output Compare B Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register B (OCR1B).

Note that a forced output compare (FOC1B) strobe will not set the OCF1B flag.

OCF1B is automatically cleared when the output compare match B interrupt vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

- **Bit 1 – OCF1A: Timer/Counter1, Output Compare A Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register A (OCR1A).

Note that a forced output compare (FOC1A) strobe will not set the OCF1A flag.

OCF1A is automatically cleared when the output compare match A interrupt vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- **Bit 0 – TOV1: Timer/Counter1, Overflow Flag**

The setting of this flag is dependent of the WGM13:0 bits setting. In normal and CTC modes, the TOV1 flag is set when the timer overflows. Refer to [Table 15-5 on page 109](#) for the TOV1 flag behavior when using another WGM13:0 bit setting.

TOV1 is automatically cleared when the Timer/Counter1 overflow interrupt vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

## 11. Interrupts

This section describes the specifics of the interrupt handling as performed in Atmel® ATmega328P. For a general explanation of the AVR® interrupt handling, refer to [Section 6.7 "Reset and Interrupt Handling" on page 15](#).

- Each interrupt vector occupies two instruction words in Atmel ATmega328P.
- In Atmel ATmega328P, the reset vector is affected by the BOOTRST fuse, and the interrupt vector start address is affected by the IVSEL bit in MCUCR.

### 11.1 Interrupt Vectors in ATmega328P

Table 11-1. Reset and Interrupt Vectors in ATmega328P

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

Table 11-2 shows reset and interrupt vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. This is also the case if the reset vector is in the application section while the interrupt vectors are in the boot section or vice versa.

**Table 11-2. Reset and Interrupt Vectors Placement in ATmega328P<sup>(1)</sup>**

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot reset address + 0x0002
0	0	Boot reset address	0x002
0	1	Boot reset address	Boot reset address + 0x0002

Note: 1. For the BOOTRST fuse "1" means unprogrammed while "0" means programmed.

The most typical and general program setup for the reset and interrupt vector addresses in Atmel® ATmega328P is:

Address	Labels	Code	Comments
0x0000		jmp RESET	; Reset Handler
0x0002		jmp EXT_INT0	; IRQ0 Handler
0x0004		jmp EXT_INT1	; IRQ1 Handler
0x0006		jmp PCINT0	; PCINT0 Handler
0x0008		jmp PCINT1	; PCINT1 Handler
0x000A		jmp PCINT2	; PCINT2 Handler
0x000C		jmp WDT	; Watchdog Timer Handler
0x000E		jmp TIM2_COMPA	; Timer2 Compare A Handler
0x0010		jmp TIM2_COMPB	; Timer2 Compare B Handler
0x0012		jmp TIM2_OVF	; Timer2 Overflow Handler
0x0014		jmp TIM1_CAPT	; Timer1 Capture Handler
0x0016		jmp TIM1_COMPA	; Timer1 Compare A Handler
0x0018		jmp TIM1_COMPB	; Timer1 Compare B Handler
0x001A		jmp TIM1_OVF	; Timer1 Overflow Handler
0x001C		jmp TIM0_COMPA	; Timer0 Compare A Handler
0x001E		jmp TIM0_COMPB	; Timer0 Compare B Handler
0x0020		jmp TIM0_OVF	; Timer0 Overflow Handler
0x0022		jmp SPI_STC	; SPI Transfer Complete Handler
0x0024		jmp USART_RXC	; USART, RX Complete Handler
0x0026		jmp USART_UDRE	; USART, UDR Empty Handler
0x0028		jmp USART_TXC	; USART, TX Complete Handler
0x002A		jmp ADC	; ADC Conversion Complete Handler
0x002C		jmp EE_RDY	; EEPROM Ready Handler
0x002E		jmp ANA_COMP	; Analog Comparator Handler
0x0030		jmp TWI	; 2-wire Serial Interface Handler
0x0032		jmp SPM_RDY	; Store Program Memory Ready Handler
0x0033	RESET:	ldi r16, high(RAMEND);	Main program start
0x0034		out SPH,r16	; Set Stack Pointer to top of RAM
0x0035		ldi r16, low(RAMEND)	
0x0036		out SPL,r16	
0x0037		sei	; Enable interrupts
0x0038	<instr>	xxx	
...	...	...	...