E/S Paralelo *(análogo para PB, PC, PE, PF, PG, PH)*

PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
0x02 (0x22)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
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	

DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x01 (0x21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
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	

PINA – Port A Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x00 (0x20)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Interrupciones externas INT0-INT7 *(Vectores INT0addr-INT7addr)*

EICRA – External Interrupt Control Register A

The External Interrupt Control Register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
(0x69)	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	EICRA
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	

EICRB – External Interrupt Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x6A)	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	EICRB
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	

EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
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	

EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	IINTF0	EIFR
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	

ISCn1	ISCn0	Description
0	0	The low level of INTn generates an interrupt request.
0	1	Any edge of INTn generates asynchronously an interrupt request.
1	0	The falling edge of INTn generates asynchronously an interrupt request.
1	1	The rising edge of INTn generates asynchronously an interrupt request.

Interrupciones por cambio PCINT0 –PCINT23 *(Vectores PCINT0addr-PCINT3addr)*

PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
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	

PCMSK1 – Pin Change Mask Register 1

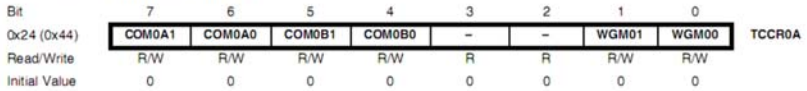
Bit	7	6	5	4	3	2	1	0	
(0x6C)	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
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	

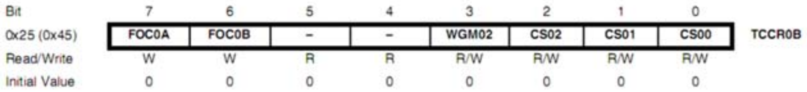
Timer 0 (8 bits) (*análogo para Timer 2*)

TCCR0A – Timer/Counter Control Register A



Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
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	

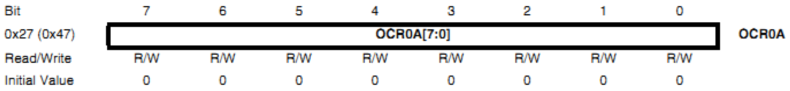
TCCR0B – Timer/Counter Control Register B



Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

OCR0A – Output Compare Register A

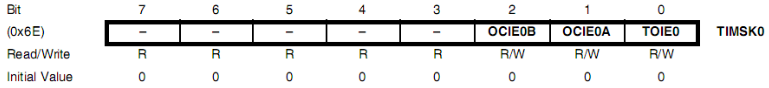
Análogo para OCR0B



Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
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	

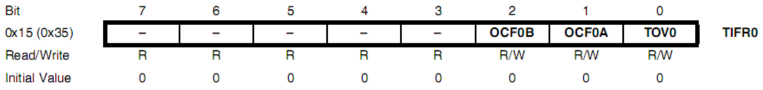
Vectores OCI0Aaddr, OCI0Baddr, TOI0addr

TIMSK0 – Timer/Counter Interrupt Mask Register

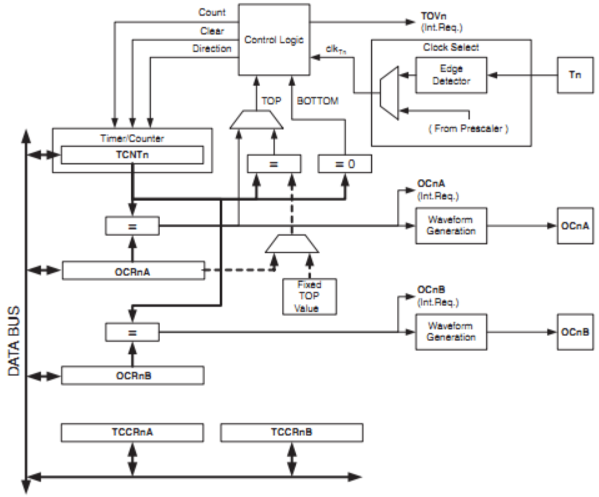


Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIFR0 – Timer/Counter 0 Interrupt Flag Register

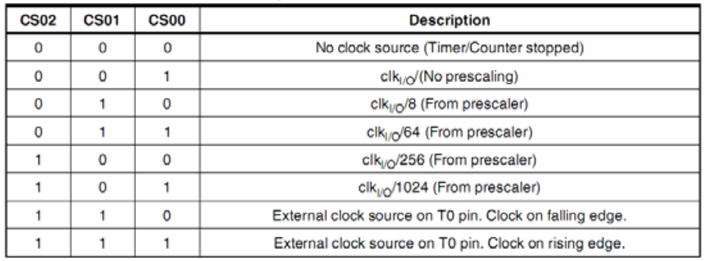


Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



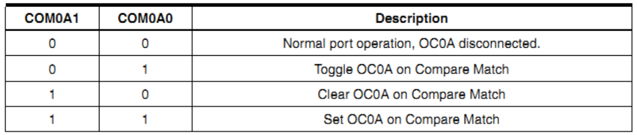
Timer 0 (8 bits) continuación (*análogo para Timer 2*)

Preescaler



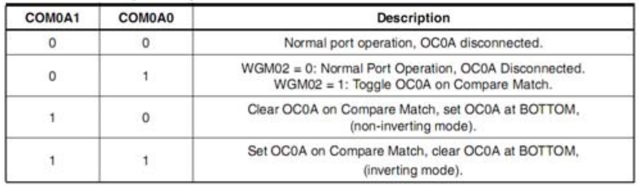
CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Compare Output Mode Non PWM



COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

Compare Output Mode Fast PWM

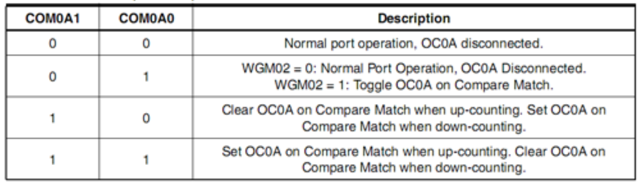


COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.

0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected.
		WGM02 = 1: Toggle OC0A on Compare Match.

1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM, (inverting mode).

Compare Output Mode Phase Correct PWM

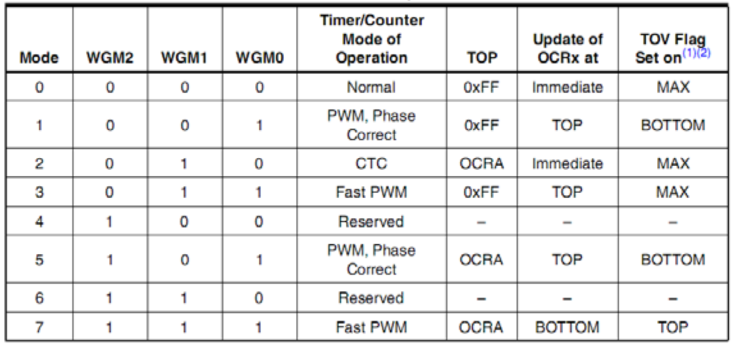


COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.

0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected.
		WGM02 = 1: Toggle OC0A on Compare Match.

1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

Waveform Generation Mode



Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on[(1)(2)]
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	TOP	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Timer 1 (16 bits) (*análogo para Timers 3, 4, 5*)

TCCR1A – Timer/Counter 1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	TCCR1A
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	

TCCR1B – Timer/Counter 1 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	

TCNT1H and TCNT1L – Timer/Counter 1

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	

OCR1AH and OCR1AL – Output Compare Register 1 A *Análogo para OCR1B y OCR1C*

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	

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	

Vectores ICI1addr, OCI1Aaddr, OCI1Baddr, OCI1Caddr, TOI1addr

TIMSK1 – Timer/Counter 1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	OCIE1C	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	OCF1C	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Timer 1 (16 bits) continuación (*análogo para Timers 3, 4, 5*)

Preescaler

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{I/O}/1$ (No prescaling
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

Compare Output Mode Non PWM

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.

0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected.
		WGM02 = 1: Toggle OC0A on Compare Match.

1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

Compare Output Mode Fast PWM

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.

0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected.
		WGM02 = 1: Toggle OC0A on Compare Match.

1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM, (inverting mode).

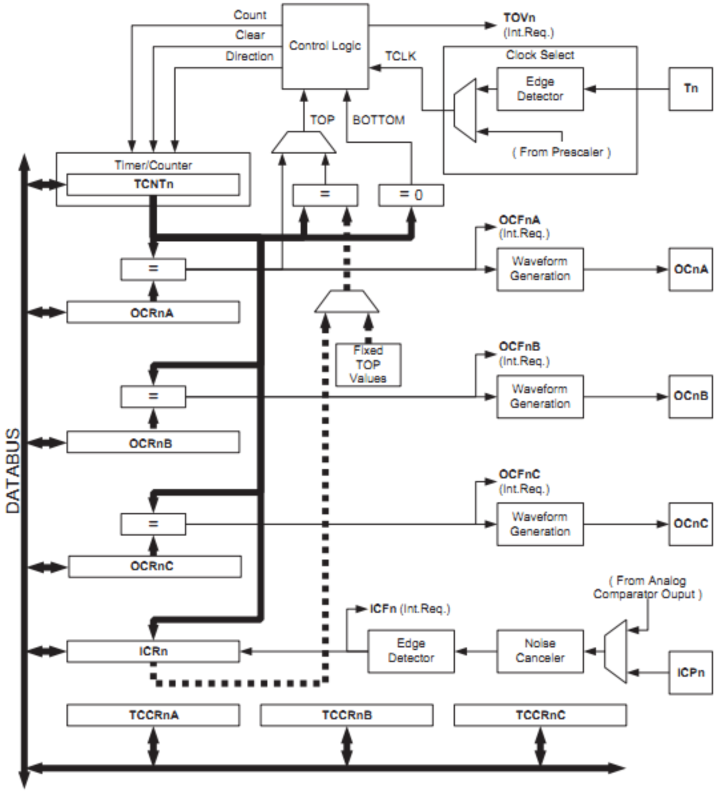
Compare Output Mode Phase Correct PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected.
0	1	WGM13:0 =9 or 11: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected.
1	0	Clear OCnA/OCnB/OCnC on compare match when up-counting. Set OCnA/OCnB/OCnC on compare match when downcounting.
1	1	Set OCnA/OCnB/OCnC on compare match when up-counting. Clear OCnA/OCnB/OCnC on compare match when downcounting.

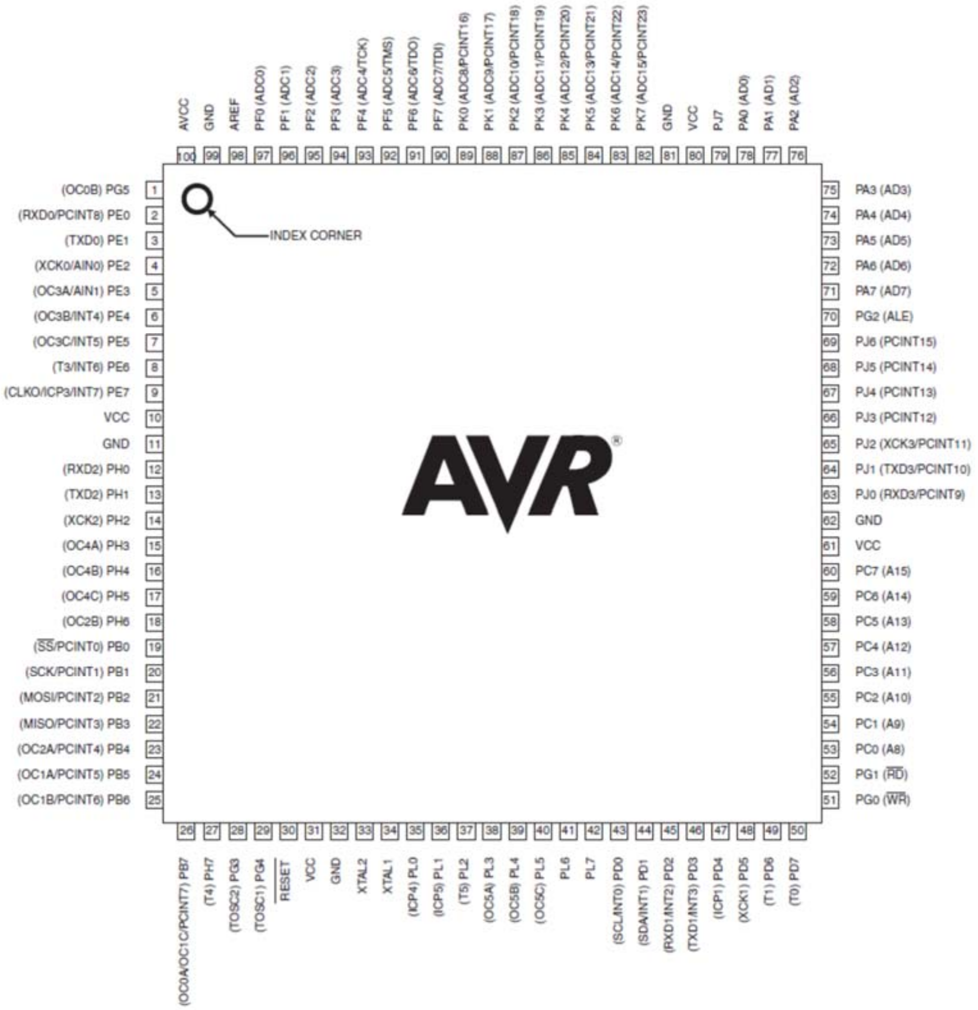
Waveform Generation Mode

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnX at	TOVn 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	OCRnA	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	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM,Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Timer 1 (16 bits) continuación (*análogo para Timers 3, 4, 5*)



Pinout AtMega2560



AVR Instruction Set

Printed on 16/03/2010

Transfer

Register

MOV Rt, Rs	Rt ← Rs
MOVW Rt2, Rs2	Rt+1:Rt ← Rs+1:Rs

Inmediate

LDI Rh, k8	Rh ← k8
SER Rh	Rh ← 0xFF

Direct

LDS Rt, k16	Rt ← DM(k16)
STS k16, Rs	DM(k16) ← Rt

Indirect

LD Rt, Rp	Rt ← DM(Rp)
LD Rt, -Rp	Rp--; Rt ← DM(Rp)
LD Rt, Rp+	Rt ← DM(Rp); Rp++
LDD Rt, Ry+k6	Rt ← DM(Ry+k6)
ST Rp, Rs	DM(Rp) ← Rs
ST -Rp, Rs	Rp--; DM(Rp) ← Rs
ST Rp+, Rs	DM(Rp) ← Rs; Rp++
ST Ry+k6, Rs	DM(Ry+k6) ← Rs

Stack

PUSH Rs	STACK ← Rs
POP Rt	Rs ← STACK

I/O

IN Rt, A	Rt ← IO(A)
OUT A, Rs	IO(A) ← Rs

Program Memory

LPM	R0 ← PM(Z)
LPM Rt, Z	Rt ← PM(Z)
LPM Rt, Z+	Rt ← PM(Z); Z++
ELPM	R0 ← PM(RAMPZ:Z)
ELPM Rt, Z	Rt ← PM(RAMPZ:Z)
ELPM Rt, Z+	Rt ← PM(RAMPZ:Z); Z++
SPM	PM(Z) ← R0

Arithmetic / Logic

Arithmetic

ADD Rt, Rs	Rt ← Rt + Rs
ADC Rt, Rs	Rt ← Rt + Rs + C
ADIW Rd, k6	Rd ← Rd + k6
SUB Rt, Rs	Rt ← Rt - Rs
SBC Rt, Rs	Rt ← Rt - Rs - C
SBIW Rd, k6	Rd+1:Rd ← Rd+1:Rd - k6
SUBI Rh, k8	Rh ← Rh - k8
SBCI Rh, k8	Rh ← Rh - k8 - C
INC Rt	Rt ← Rt +1
DEC Rt	Rt ← Rt -1
CP Rt, Rs	Rt - Rs
CPC Rt, Rs	Rt - Rs - C
CPI Rh, k8	Rh - k8

Logic

AND Rt, Rs	Rt ← Rt • Rs
ANDI Rh, k8	Rh ← Rh • k8
OR Rt, Rs	Rt ← Rt v Rs
ORI Rh, k8	Rh ← Rh v k8
EOR Rt, Rs	Rt ← Rt eor Rs
COM Rt	Rt ← not Rt
NEG Rt	Rt ← -Rt
ROR Rt	Rotate Right
ROL Rt	Rotate Left
LSR Rt	Logic Shift Right
LSL Rt	Logic Shift Left
ASR Rt	Arithmetic Shift Right
SBR Rh, k8	*Rh ← Rh v k8*
CBR Rh, k8	*Rh ← Rh • not k8*
TST Rt	*Rt = 0? (Rt ← Rt • Rt)*
CLR Rt	*Rt ← 0 (Rt ← Rt eor Rt)*

Multiply

MUL Rt, Rs	R1:R0 ← Rt * Rs
MULS Rh, Ri	R1:R0 ← Rh * Ri (signed)
MULSU Rm, Rn	R1:R0 ← Rm * Rn (sig/unsig)
FMUL Rm, Rn	R1:R0 ← (Rt * Rs) << 1
FMULS Rm, Rn	R1:R0 ← (Rh * Ri) << 1
FMULSU Rm, Rn	R1:R0 ← (Rm * Rn) << 1

Jump / Call / Branch

Jump

RJMP k12	PC ← PC + k12
IJMP	PC ← Z
EIJMP	PC ← RAMPZ:Z
JMP k22	PC ← k22

Call/Return

RCALL k12	PC←PC+k12; STACK←PC
ICALL	PC ← Z; STACK ← PC
EICALL	PC←RAMPZ:Z; STACK←PC
CALL k22	PC ← k22; STACK ← PC
RET	PC ← STACK
RETI	PC ← STACK; I ← 1

Branch

CPSE Rt, Rs	Skip next instr if Rt=Rs
SBRC Rs, b	Skip next instr if Rs(b)=0
SBRS Rs, b	Skip next instr if Rs(b)=1
SBIC A, b	Skip next instr if IO(A)(b)=0
SBIS A, b	Skip next instr if IO(A)(b)=1
BRBC s, k7	Branch if SREG(b)=0
BRBS s, k7	Branch if SREG(b)=1
BRSH k7	Branch if same or higer (unsig)
BRLO k7	Branch if lower (unsig)
BRGE k7	Branch if grater or equal (sig)
BRLT k7	Branch if less than (sig)
BREQ k7	Branch if equal
BRNE k7	Branch if not equal
BRCS k7	Branch if carry set
BRCC k7	Branch if carry clear
BRMI k7	Branch if minus
BRPL k7	Branch if plus
BRHS k7	Branch if half-carry set
BRHC k7	Branch if half-carry clear
BRTS k7	Branch if T set
BRTC k7	Branch if T clear
BRVS k7	Branch if overflow set
BRVC k7	Branch if overflow clear
BRIE k7	Branch if interrupt mask set
BRID k7	Branch if interrupt mask clear

Bit / Others

Bit Set

BSET s	SREG(b) ← 1
BCLR s	SREG(b) ← 0
SBI a, b	IO(a)(b) ← 1
CBI a, b	IO(a)(b) ← 0
BST Rt, b	T ← Rt(b)
BLD Rt, b	Rt(b) ← T
SEC	*C=1*
CLC	*C=0*
SEN	*N=1*
CLN	*N=0*
SEZ	*Z=1*
CLZ	*Z=0*
SES	*S=1 (S = V eor N)*
CLS	*S=0 (S = V eor N)*
SEV	*V=1*
CLV	*V=0*
SET	*T=1*
CLT	*T=0*
SEH	*H=1*
CLH	*H=0*
SEI	*I=1*
CLI	*I=0*
SWAP Rt	Rt(7:4) ↔ Rt(3:0)

MCU Control

NOP	No operación
BREAK	Break
SLEEP	Sleep
WDR	Watchdog Reset

Register
Rs, Rt Ordinary Source and Target register R0..R31
Rs2, Rt2 Ordinary Source and Target even register R0, R2, R4...R30
Rh, Ri Source and Target register between R16..R31
Rm, Rn Source and Target register between R16..R23
Rd Twin register R24(R25), R26(R27), R28(R29), R30(R31)
Rp Pointer register X=R26(R27), Y=R28(R29), Z=R30(R31)
Ry Pointer register with displacement Y=R28(R29), Z=R30(R31)

Constant
k6 6-Bit-Constant 0..63
k7 7-Bit-Constant -64..+63
k8 8-Bit-Constant 0..255
k12 12-Bit-Constante -2048..+2047
k16 16-Bit-Address 0..65535
k22 22-Bit-Address 0..4M

RAMPZ Segment Register

DM(..) Data memory
PM(..) Program memory

Instructions in *ITALICS* are pseudointructions

Bit
s State Register bit 0..7
b Bit position 0..7

Port
A Ordinary Port address 0..63
a Lower page port address 0..31

The AVR instruction set is more orthogonal than most eight-bit microcontrollers, in particular the PIC microcontrollers with which AVR competes today. Arithmetic operations work on registers R0-R31 but not directly on RAM and take one clock cycle, except for multiplication and word-wide addition (ADIW and SBIW) which take two cycles. The mostly-regular instruction set makes programming it using C (or even Pascal) compilers pretty straightforward.

The data address space consists of the register file, I/O registers, and SRAM. The AVRs have 32 single-byte registers and are classified as 8-bit RISC devices. RAM and I/O space can be accessed only by copying to or from registers. Indirect access (including optional postincrement, predecrement or constant displacement) is possible through registers X, Y, and Z. All accesses to RAM takes two clock cycles. Moving between registers and I/O is one cycle. Moving eight or sixteen bit data between registers or constant to register is also one cycle. Reading program memory (LPM) takes three cycles.

There are two types of conditional branches: skips and jumps to address. Skips (SBxx) test an arbitrary bit in a register or I/O and skip the next instruction if the test was true. Conditional branches (BRxx) can test an ALU flag and jump to specified address.