

EPREUVE ECRITE

Ministère de L'Education nationale
et de la Formation professionnelle

EXAMEN DE FIN D'ETUDES SECONDAIRES TECHNIQUES

Régime de la formation de technicien

Division électrotechnique

Section: communication

BRANCHE: Microélectronique

SESSION: 2009 DATE: DUREE: 3 heures

Exercice 1 (3 + 3 + 8 = 14 points)

Moteur pas à pas:

- Expliquez la différence qu'il y a entre un moteur pas à pas fonctionnant de façon *unipolaire* et un moteur pas à pas fonctionnant de façon *bipolaire*.
- Citez tous les types de pas que vous connaissez et d'après lesquels un moteur pas à pas peut travailler et expliquez-les brièvement.
- Il faut dès à présent commander un moteur pas à pas.

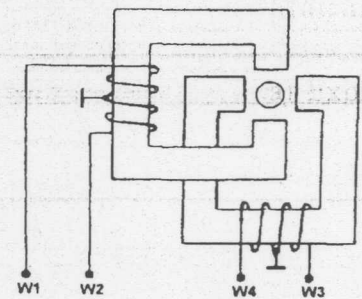
Choisissez un type de fonctionnement qui vous convient et dressez un tableau contenant les octets de commande.

Pour déterminer la valeur des octets de commande, faites attention aux positions des bits correspondant aux bornes de connexion du moteur.

| | | | |
|----|----|----|----|
| W4 | W2 | W3 | W1 |
|----|----|----|----|

Le moteur pas à pas est branché aux broches PC4 – PC7 du Port C. De plus un interrupteur S est branché à la masse à partir du **bit 4** du Port A.

Après lancement du programme le moteur pas à pas doit tourner *dans* le sens des aiguilles d'une montre ou *contre* le sens de l'aiguille d'une montre selon la position de l'interrupteur S.



Ecrivez maintenant le programme en assembleur pour le microcontrôleur ATmega32 et commentez-le de façon appropriée.

Remarques:

- Ne notez pas les définitions et les initialisations standards, mais uniquement les initialisations nécessaires à la bonne marche du moteur pas à pas.
- Pour pouvoir mettre en marche le moteur correctement le sous-programme temporel W50ms qui produit une temporisation de 50 ms est mis à votre disposition!

Exercice 2 (6 + 4 = 10 points)

Interruptions externes:

Développez un programme assembleur pour le microcontrôleur ATmega32 pour d'abord lire un octet à partir du Port C lorsqu'un front montant apparaît à l'entrée d'interruption INT1 (PD3) et pour ensuite mettre cet octet dans un tableau. Le tableau (RxTab) d'une grandeur de 256 octets se trouve dans la mémoire SRAM à partir de l'adresse 0x0100.

- a)
- Nommez tous les registres à fonctions spéciales impliqués dans la programmation de l'interruption et expliquez comment il faut placer les bits correctement à l'intérieur de ces registres.
 - Ecrivez le programme principal qui, entre autres, procède à toutes les initialisations nécessaires. Commentez le programme de façon très détaillée.

Remarque: Ne notez pas les définitions et les initialisations standards.

- b)
- Ecrivez maintenant la sous-routine d'interruption qui lit les données et les met dans le tableau. Commentez également cette partie du programme de façon appropriée.

Remarque: Notez qu'une demande d'interruption peut interrompre à n'importe quel moment un programme qui tourne.



d) Ecrivez la partie du programme assembleur pour configurer l'interface série fonctionnant à la vitesse 19200 baud en mode **8E1**.

Remarque: La commande de la transmission se fait d'après le principe du polling!

e) Dessinez le diagramme fleuve de la sous-routine d'interruption (isr_T0) du timer qui lit les états du Port C et les transmet à l'interface série.

f) Ecrivez le programme assembleur documenté de façon appropriée qui correspond au diagramme fleuve de la sous-routine d'interruption.

Exercice 4 (9 points)

Le convertisseur analogique - numérique:

- Dessinez le schéma bloc du principe de conversion au moyen d'une série de « pesées » successives.
- Indiquez une autre dénomination pour désigner cette méthode de conversion.
- Décrivez en vos propres mots le principe de fonctionnement de ce type de conversion.



A.1 Tableau ASCII

| Lower Bits | Upper Bits | bin | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|------------|------------|-----|------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| bin | hex | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
| 0000 | 0 | NUL | DLE | space | 0 | @ | P | ~ | p | Ç | É | á | ☐ | L | ⌌ | α | ≡ | |
| | | 0 | 16 | 32 | 48 | 64 | 80 | 96 | 112 | 128 | 144 | 160 | 176 | 192 | 208 | 224 | 240 | |
| 0001 | 1 | SOH | DC1 | ! | 1 | A | Q | a | q | ü | æ | í | ☐ | ⌋ | ⌋ | ß | ‡ | |
| | | 1 | 17 | 33 | 49 | 65 | 81 | 97 | 113 | 129 | 145 | 161 | 277 | 193 | 209 | 225 | 241 | |
| 0010 | 2 | STX | DC2 | " | 2 | B | R | b | r | é | Æ | ó | ☐ | ⌋ | ⌋ | Γ | ≥ | |
| | | 2 | 18 | 34 | 50 | 66 | 82 | 98 | 114 | 130 | 146 | 162 | 178 | 194 | 210 | 226 | 242 | |
| 0011 | 3 | ETX | DC3 | # | 3 | C | S | c | s | â | ô | ú | | ⌋ | ⌋ | π | ≤ | |
| | | 3 | 19 | 35 | 51 | 67 | 83 | 99 | 115 | 131 | 147 | 163 | 179 | 195 | 211 | 227 | 243 | |
| 0100 | 4 | EOT | DC4 | \$ | 4 | D | T | d | t | ä | ö | ñ | ⌋ | — | ⌋ | Σ | ∫ | |
| | | 4 | 20 | 36 | 52 | 68 | 84 | 100 | 116 | 132 | 148 | 164 | 180 | 196 | 212 | 228 | 244 | |
| 0101 | 5 | ENQ | NAK | % | 5 | E | U | e | u | à | ò | Ñ | ⌋ | ⌋ | ⌋ | σ | ∫ | |
| | | 5 | 21 | 37 | 53 | 69 | 85 | 101 | 117 | 133 | 149 | 165 | 181 | 197 | 213 | 229 | 245 | |
| 0110 | 6 | ACK | SYN | & | 6 | F | V | f | v | å | û | ª | ⌋ | ⌋ | ⌋ | μ | ÷ | |
| | | 6 | 22 | 38 | 54 | 70 | 86 | 102 | 118 | 134 | 150 | 166 | 182 | 198 | 214 | 230 | 246 | |
| 0111 | 7 | BEL | ETB | ' | 7 | G | W | g | w | ç | ù | º | ⌋ | ⌋ | ⌋ | τ | ≈ | |
| | | 7 | 23 | 39 | 55 | 71 | 87 | 103 | 119 | 135 | 151 | 167 | 183 | 199 | 215 | 231 | 247 | |
| 1000 | 8 | BS | CAN | (| 8 | H | X | h | x | ê | ÿ | ¿ | ⌋ | ⌋ | ⌋ | Φ | ° | |
| | | 8 | 24 | 40 | 56 | 72 | 88 | 104 | 120 | 136 | 152 | 168 | 184 | 200 | 216 | 232 | 248 | |
| 1001 | 9 | TAB | EM |) | 9 | I | Y | i | y | ë | Ö | ⌋ | ⌋ | ⌋ | ⌋ | ⊙ | · | |
| | | 9 | 25 | 41 | 57 | 73 | 89 | 105 | 121 | 137 | 153 | 169 | 185 | 201 | 217 | 233 | 249 | |
| 1010 | A | LF | SUB | * | : | J | Z | j | z | è | Ü | ⌋ | ⌋ | ⌋ | ⌋ | Ω | · | |
| | | 10 | 26 | 42 | 58 | 74 | 90 | 106 | 122 | 138 | 154 | 170 | 186 | 202 | 218 | 234 | 250 | |
| 1011 | B | VT | ESC | + | ; | K | [| k | { | ï | ç | ½ | ⌋ | ⌋ | ⌋ | δ | √ | |
| | | 11 | 27 | 43 | 59 | 75 | 91 | 107 | 123 | 139 | 155 | 171 | 187 | 203 | 219 | 235 | 251 | |
| 1100 | C | FF | FS | , | < | L | \ | l | | î | £ | ¼ | ⌋ | ⌋ | ⌋ | ∞ | n | |
| | | 12 | 28 | 44 | 60 | 76 | 92 | 108 | 124 | 140 | 156 | 172 | 188 | 204 | 220 | 236 | 252 | |
| 1101 | D | CR | GS | - | = | M |] | m | } | ï | ¥ | i | ⌋ | = | ⌋ | φ | 2 | |
| | | 13 | 29 | 45 | 61 | 77 | 93 | 109 | 125 | 141 | 157 | 173 | 189 | 205 | 221 | 237 | 253 | |
| 1110 | E | SO | RS | . | > | N | ^ | n | ~ | Ä | Ē | « | ⌋ | ⌋ | ⌋ | ε | ■ | |
| | | 14 | 30 | 46 | 62 | 78 | 94 | 110 | 126 | 142 | 158 | 174 | 190 | 206 | 222 | 238 | 254 | |
| 1111 | F | SI | US | / | ? | O | _ | o | DEL | Å | f | » | ⌋ | ⌋ | ⌋ | ∩ | | |
| | | 15 | 31 | 47 | 63 | 79 | 95 | 111 | 127 | 143 | 159 | 175 | 191 | 207 | 223 | 239 | 255 | |

ASCII CONTROL CHARACTER ABBREVIATIONS

| | | | | | | | | |
|-----|---|-----------------------|-----|---|-------------------------|-----|---|---------------------------------|
| NUL | : | null | VT | : | vertical tabulation | SYN | : | synchronous idle |
| SOH | : | start of heading | FF | : | form feed | ETB | : | end of transmission block |
| STX | : | start of text | CR | : | carriage return | CAN | : | cancel |
| ETX | : | end of text | SO | : | shift out | EM | : | end of medium |
| EOT | : | end of transmission | SI | : | shift in | SUB | : | substitute |
| ENQ | : | enquiry | DLE | : | data link escape | ESC | : | escape |
| ACK | : | acknowledge | DC1 | : | device control 1 / XON | FS | : | file separator |
| BEL | : | bell | DC2 | : | device control 2 | GS | : | group separator |
| BS | : | backspace | DC3 | : | device control 3 / XOFF | RS | : | record separator / req. to send |
| TAB | : | horizontal tabulation | DC4 | : | device control 4 | US | : | unit separator |
| LF | : | linefeed | NAK | : | negative acknowledge | DEL | : | delete |

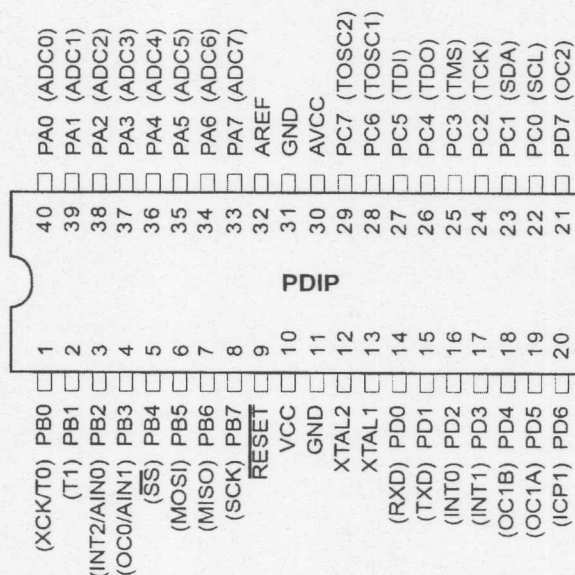


A.2 Set des registres

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|-------------------------------------------|--------|------------------------------------------------------|--------|--------|--------|--------|------------|--------|--------|-----------------------|
| \$3F (\$5F) | SREG | I | T | H | S | V | N | Z | C | 10 |
| \$3E (\$5E) | SPH | - | - | - | - | SP11 | SP10 | SP9 | SP8 | 12 |
| \$3D (\$5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | 12 |
| \$3C (\$5C) | OCR0 | Timer/Counter0 Output Compare Register | | | | | | | | 82 |
| \$3B (\$5B) | GICR | INT1 | INT0 | INT2 | - | - | - | IVSEL | IVCE | 47, 67 |
| \$3A (\$5A) | GIFR | INTF1 | INTF0 | INTF2 | - | - | - | - | - | 68 |
| \$39 (\$59) | TIMSK | OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 | 82, 112, 130 |
| \$38 (\$58) | TIFR | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 | 83, 113, 130 |
| \$37 (\$57) | SPMCR | SPMIE | RWWSB | - | RWWSRE | BLBSET | PGWRT | PGERS | SPMEN | 248 |
| \$36 (\$56) | TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE | 177 |
| \$35 (\$55) | MCUCR | SE | SM2 | SM1 | SM0 | ISC11 | ISC10 | ISC01 | ISC00 | 32, 66 |
| \$34 (\$54) | MCUCSR | JTD | ISC2 | - | JTRF | WDRF | BORF | EXTRF | PORF | 40, 67, 228 |
| \$33 (\$53) | TCCR0 | FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 | 80 |
| \$32 (\$52) | TCNT0 | Timer/Counter0 (8 Bits) | | | | | | | | 82 |
| \$31 ⁽¹⁾ (\$51) ⁽¹⁾ | OSCCAL | Oscillator Calibration Register | | | | | | | | 30 |
| | OCDR | On-Chip Debug Register | | | | | | | | 224 |
| \$30 (\$50) | SFJOR | ADTS2 | ADTS1 | ADTS0 | - | ACME | PUD | PSR2 | PSR10 | 56, 85, 131, 198, 218 |
| \$2F (\$4F) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | FOC1A | FOC1B | WGM11 | WGM10 | 107 |
| \$2E (\$4E) | TCCR1B | ICNC1 | ICES1 | - | WGM13 | WGM12 | CS12 | CS11 | CS10 | 110 |
| \$2D (\$4D) | TCNT1H | Timer/Counter1 – Counter Register High Byte | | | | | | | | 111 |
| \$2C (\$4C) | TCNT1L | Timer/Counter1 – Counter Register Low Byte | | | | | | | | 111 |
| \$2B (\$4B) | OCR1AH | Timer/Counter1 – Output Compare Register A High Byte | | | | | | | | 111 |
| \$2A (\$4A) | OCR1AL | Timer/Counter1 – Output Compare Register A Low Byte | | | | | | | | 111 |
| \$29 (\$49) | OCR1BH | Timer/Counter1 – Output Compare Register B High Byte | | | | | | | | 111 |
| \$28 (\$48) | OCR1BL | Timer/Counter1 – Output Compare Register B Low Byte | | | | | | | | 111 |
| \$27 (\$47) | ICR1H | Timer/Counter1 – Input Capture Register High Byte | | | | | | | | 112 |
| \$26 (\$46) | ICR1L | Timer/Counter1 – Input Capture Register Low Byte | | | | | | | | 112 |
| \$25 (\$45) | TCCR2 | FOC2 | WGM20 | COM21 | COM20 | WGM21 | CS22 | CS21 | CS20 | 125 |
| \$24 (\$44) | TCNT2 | Timer/Counter2 (8 Bits) | | | | | | | | 127 |
| \$23 (\$43) | OCR2 | Timer/Counter2 Output Compare Register | | | | | | | | 127 |
| \$22 (\$42) | ASSR | - | - | - | - | AS2 | TCN2UB | OCR2UB | TCR2UB | 128 |
| \$21 (\$41) | WDTCR | - | - | - | WDTOE | WDE | WDP2 | WDP1 | WDP0 | 42 |
| \$20 ⁽²⁾ (\$40) ⁽²⁾ | UBRRH | URSEL | - | - | - | - | UBRR[11:8] | | | 164 |
| | UCSRC | URSEL | UMSEL | UPM1 | UPM0 | USBS | UCSZ1 | UCSZ0 | UCPOL | 162 |
| \$1F (\$3F) | EEARH | - | - | - | - | - | - | EEAR9 | EEAR8 | 19 |
| \$1E (\$3E) | EEARL | EEPROM Address Register Low Byte | | | | | | | | 19 |
| \$1D (\$3D) | EEDR | EEPROM Data Register | | | | | | | | 19 |
| \$1C (\$3C) | EEDR | - | - | - | - | EERIE | EEMWE | EWE | EERE | 19 |
| \$1B (\$3B) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | 64 |
| \$1A (\$3A) | DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | 64 |
| \$19 (\$39) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | 64 |
| \$18 (\$38) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 64 |
| \$17 (\$37) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | 64 |
| \$16 (\$36) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | 65 |
| \$15 (\$35) | PORTC | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 65 |
| \$14 (\$34) | DDRC | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | 65 |
| \$13 (\$33) | PINC | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | 65 |
| \$12 (\$32) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 65 |
| \$11 (\$31) | DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | 65 |
| \$10 (\$30) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | 65 |
| \$0F (\$2F) | SPDR | SPI Data Register | | | | | | | | 138 |
| \$0E (\$2E) | SPSR | SPIF | WCOL | - | - | - | - | - | SPI2X | 138 |
| \$0D (\$2D) | SPCR | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | 136 |
| \$0C (\$2C) | UDR | USART I/O Data Register | | | | | | | | 159 |
| \$0B (\$2B) | UCSRA | RXC | TXC | UDRE | FE | DOR | PE | U2X | MPCM | 160 |
| \$0A (\$2A) | UCSRB | RXCIE | TXCIE | UDRIE | RXEN | TXEN | UCSZ2 | RXB8 | TXB8 | 161 |
| \$09 (\$29) | UBRRL | USART Baud Rate Register Low Byte | | | | | | | | 164 |
| \$08 (\$28) | ACSR | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | 199 |
| \$07 (\$27) | ADMUX | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | 214 |
| \$06 (\$26) | ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | 216 |
| \$05 (\$25) | ADCH | ADC Data Register High Byte | | | | | | | | 217 |
| \$04 (\$24) | ADCL | ADC Data Register Low Byte | | | | | | | | 217 |
| \$03 (\$23) | TWDR | Two-wire Serial Interface Data Register | | | | | | | | 179 |
| \$02 (\$22) | TWAR | TWA6 | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE | 179 |
| \$01 (\$21) | TWSR | TWS7 | TWS6 | TWS5 | TWS4 | TWS3 | - | TWPS1 | TWPS0 | 178 |
| \$00 (\$20) | TWBR | Two-wire Serial Interface Bit Rate Register | | | | | | | | 177 |



A.3 Schéma de brochage



A.4 Tableau des vecteurs d'interruption

| Numéro vecteur | Adresse de la mémoire programme | Adresse symbolique | Source des interruptions | Responsable de l'interruption |
|----------------|---------------------------------|--------------------|--------------------------|-------------------------------------------------|
| 1 | 0x000 | Pas d'adresse | RESET | Pin externe, Power-On-Reset, Brown-Out-Reset... |
| 2 | 0x002 | INT0addr | INT0 | Entrée interruption externe 0 |
| 3 | 0x004 | INT1addr | INT1 | Entrée interruption externe 1 |
| 4 | 0x006 | INT2addr | INT2 | Entrée interruption externe 2 |
| 5 | 0x008 | OC2addr | TIMER2 COMP | Compare Match du Timer 2 |
| 6 | 0x00A | OVF2addr | TIMER2 OVF | Overflow Match du Timer 2 |
| 7 | 0x00C | ICP1addr | TIMER1 CAPT | Capture Event du Timer 1 |
| 8 | 0x00E | OC1Aaddr | TIMER1 COMPA | Compare Match A du Timer 1 |
| 9 | 0x010 | OC1Baddr | TIMER1 COMPB | Compare Match B du Timer 1 |
| 10 | 0x012 | OVF1addr | TIMER1 OVF | Overflow du Timer 1 |
| 11 | 0x014 | OCOaddr | TIMERO COMP | Compare Match du Timer 0 |
| 12 | 0x016 | OVFOaddr | TIMERO OVF | Overflow du Timer 0 |
| 13 | 0x018 | SPIaddr | SPI, STC | Transmission série terminée |
| 14 | 0x01A | URXCaddr | USART, RXC | USART, registre réception plein |
| 15 | 0x01B | UDREaddr | USART, UDRE | USART, registre UDR vide |
| 16 | 0x01E | UTXCaddr | USART, TXC | USART, registre transmission vide |
| 17 | 0x020 | ADCCaddr | ADC | Conversion AN complète |
| 18 | 0x022 | ERDYaddr | EE_RDY | EEPROM-Interface |
| 19 | 0x024 | ACIaddr | ANA_COMP | Comparateur analogique |
| 20 | 0x026 | TWIaddr | TWI | I ² C-Interface |
| 21 | 0x028 | SPMRaddr | SPM_RDY | Interface pour mémoire programme |

A.5 Registre d'état SREG

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------------------|-------------|------------|----------------|--------|---------------|---------------|-----------|------------|
| SREG 0x3F | I Interrupt | T Transfer | H Halfcarry | S Sign | V Overflow | N Negative | Z Zero | C Carry |
| Valeur de début | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Bit 7 ou bit I pour global Interrupt enable

Ce bit sert à autoriser ou à interdire la prise en compte des interruptions par l'unité centrale.

Bit 6 ou bit T pour copy sTorage

Ce bit sert de source ou de destination pour le bit manipulé par les instructions de copie de bit que sont **bld** et **bst**. Un bit quelconque d'un registre peut ainsi être copié dans T au moyen d'une instruction **bst** tandis que ce même bit T peut être copié dans un bit quelconque d'un registre au moyen d'une instruction **bld**.

Bit 5 ou bit H pour Half-carry

Ce bit est la demi-retenue classique dans tout registre d'état de microcontrôleur qui se respecte. Elle est mise à 1 lors de certaines opérations arithmétiques qui génèrent une retenue du quartet de poids faibles d'un mot de 8 bits vers le quartet de poids fort.

Bit 4 ou bit S pour Sign bit

Ce bit S ou bit de signe est le résultat du OU exclusif réalisé entre le bit d'indication de nombre négatif N et le bit de débordement V.

Bit 3 ou bit V pour Overflow bit

Ce bit est positionné à 1 lors d'un débordement se produisant lors de certaines opérations arithmétiques en complément à deux.

Bit 2 ou bit N pour Negative bit

Ce bit est positionné à 1 lorsque le résultat de certaines opérations arithmétiques et/ou logiques est négatif.

Bit 1 ou bit Z pour Zero bit

Ce bit est positionné à 1 lorsque le résultat de certaines opérations arithmétiques et/ou logiques est nul.

Bit 0 ou bit C pour Carry

Ce bit est la retenue utilisée lors de certaines opérations arithmétiques et/ou logiques qui le positionnent automatiquement si nécessaire.



A.6 Les ports d'entrée et de sortie

DDRx = Data Direction Register PORTx
PORTx = PORTx Data Register
PINx = PORTx Input Pins Address

| | | | | | | | | |
|-----------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PORTX | PORTX 7 | PORTX 6 | PORTX 5 | PORTX 4 | PORTX 3 | PORTX 2 | PORTX 1 | PORTX 0 |
| Valeur initiale | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Accessibilité | L/E | L/E | L/E | L/E | L/E | L/E | L/E | L/E |

| | | | | | | | | |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DDRX | DDRX 7 | DDRX 6 | DDRX 5 | DDRX 4 | DDRX 3 | DDRX 2 | DDRX 1 | DDRX 0 |
| Valeur initiale | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Accessibilité | L/E | L/E | L/E | L/E | L/E | L/E | L/E | L/E |

| | | | | | | | | |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PINX | PINX 7 | PINX 6 | PINX 5 | PINX 4 | PINX 3 | PINX 2 | PINX 1 | PINX 0 |
| Valeur initiale | - | - | - | - | - | - | - | - |
| Accessibilité | L | L | L | L | L | L | L | L |

| DDRX _n | PORTX _n | E/S | Tirage | Commentaire |
|-------------------|--------------------|--------|--------|------------------------------------------------------|
| 0 | 0 | Entrée | Non | PX _n est une entrée à haute impédance |
| 0 | 1 | Entrée | Oui | PX _n fournit du courant au niveau bas (*) |
| 1 | 0 | Sortie | Non | PX _n est une sortie à zéro |
| 1 | 1 | Sortie | Non | PX _n est une sortie à un |

(*) Si le bit PUD (Pull-Up Disable) dans le registre SFIOR est à l'état 0 (valeur initiale) une résistance Pull-Up interne est branchée à la masse. Si PUD est à l'état 1, la sortie est une sortie haute impédance !



A.7 Interruptions

Elles sont produites par des événements qui surviennent aux ports : INT0 au port PD2, INT1 au port PD3 et INT2 au port PB2 (interruption externe asynchrone).

MCUCR = MCU Control Register SRAM

| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|--------------------------------------|-------|-------|-----------------------------------------------------------------------------------------------------|-------|-----------------------------------------------------------------------------------------------------|-------|
| SM2 | SE | SM1 | SM0 | ISC11 | ISC10 | ISC01 | ISC00 |
| Sleep | Mode sleep 0: bloqué 1: libéré | Sleep | Sleep | INT1 00: si état devient BAS 01: si état change 10: si front descend 11: si front monte | | INT0 00: si état devient BAS 01: si état change 10: si front descend 11: si front monte | |

L'état des bits après un RESET est 0. Ils doivent être programmés à l'aide de masques ET ou OU puisque le registre n'est pas adressable par bits.

MCUCSR = MCU Control and Status Register SRAM

| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------------------------------------------|---|------|------|------|-------|------|
| JTD | ISC2 | - | JTRF | WDRF | BORF | EXTRF | PORF |
| | INT2 front 0: descendant 1: montant | | | | | | |

L'état de départ de ISC2 est 0.

GICR = General Interrupt Control Register SRAM

| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------------------------|----------------------------------|----------------------------------|---|---|---|-------|------|
| INT1 | INT0 | INT2 | - | - | - | IVSEL | IVCE |
| INT1 0: interdit 1: libéré | INT0 0: interdit 1: libéré | INT2 0: interdit 1: libéré | | | | | |

L'état des trois bits après un RESET est 0.

GIFR = General Interrupt Flag Register SRAM

| bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------------|----------------------|----------------------|---|---|---|---|---|
| INTF1 | INTF0 | INTF2 | - | - | - | - | - |
| 0: pas mis 1: mis | 0: pas mis 1: mis | 0: pas mis 1: mis | | | | | |

L'état des trois bits après un RESET est 0. Ils sont mis automatiquement à 1 par le signal de l'interruption et remis automatiquement à 0 quand la routine d'interruption est lancée. On peut également les mettre à l'état 1 par programmation.



A.8 Timer

TCCR0 = Timer/Counter Control Register 0 adresse-SRAM 0x53, adresse-SFR 0x33

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|-------|------------|-------|-------|-------|------|------|------|
| FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 |
| Write | Read/Write | R/W | R/W | R/W | R/W | R/W | R/W |

WGM0n Wave form Generation Mode (WGM01, WGM00)

Ces 2 bits déterminent le mode de fonctionnement du timer :

| $2^1 2^0$ | WGM0n |
|-----------|-------------------------------------|
| 0 0 | mode normal |
| 0 1 | mode PWM en phase correcte |
| 1 0 | mode CTC (clear timer compare mode) |
| 1 1 | Fast PWM mode (mode PWM rapide) |

COM0n Compare Match Output Mode (COM01, COM00)

Ces 2 bits déterminent le comportement de la broche de sortie OC0 :

| $2^1 2^0$ | COM0n (comportement en mode normal) |
|-----------|------------------------------------------------------------------|
| 0 0 | OC0 désactivé (broche normal du Port) |
| 0 1 | Inverse l'état (Toggle) de OC0 si égalité lors de la comparaison |
| 1 0 | Mise à 0 de OC0 si égalité lors de la comparaison |
| 1 1 | Mise à 1 de OC0 si égalité lors de la comparaison |

| $2^1 2^0$ | COM0n (comportement en mode Fast PWM) |
|-----------|---------------------------------------|
| 0 0 | OC0 désactivé (broche normal du Port) |
| 0 1 | réservé |
| 1 0 | Mode Fast PWM non inversé |
| 1 1 | Mode Fast PWM inversé |

CS0n Clock Select Timer0 (CS02, CS01, CS00)

| $2^2 2^1 2^0$ | CS0n (source du signal d'horloge) |
|---------------|----------------------------------------------------------------------|
| 0 0 0 | Timer Stop (pas de consommation de courant ; default après RESET) |
| 0 0 1 | signal d'horloge :1 |
| 0 1 0 | signal d'horloge :8 |
| 0 1 1 | signal d'horloge :64 |
| 1 0 0 | signal d'horloge :256 |
| 1 0 1 | signal d'horloge :1024 |
| 1 1 0 | signal d'horloge externe : front descendant sur T0 (PB0) |
| 1 1 1 | signal d'horloge externe : front montant sur T0 (PB0) |

FC0 Force Output Compare

0 Ce bit ne devrait pas être utilisé ici. Il doit toujours être 0



TIFR = Timer Interrupt Flag Register adr.-SRAM = 0x58 et adr.-SFR = 0x38

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|--------|--------|--------|--------|--------|--------|---------------------|------------------------|
| OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 |
| Timer2 | Timer2 | Timer1 | Timer1 | Timer1 | Timer1 | Timer0 1 : Match | Timer0 1 : Overflow |

OCF0 Output Compare Flag 0

- 0 Le contenu du registre de comptage ne correspond pas à celui du registre de comparaison.
- 1 Au cas où les contenus des registres de comptage TCNT0 et de comparaison OCR0 correspondent (anglais : compare match). Ce bit est remis automatiquement à 0 si l'interruption OC a été exécutée. On peut effacer cet indicateur manuellement en inscrivant un 1 logique.

TOV0 Timer/Counter 0 Overflow Flag

- 0 Il n'y avait eu aucun débordement.
- 1 En cas d'un débordement du registre de comptage TCNT0 cet indicateur est mis à l'état 1. Il est automatiquement remis à 0 si l'interruption OV a été exécutée. On peut effacer cet indicateur manuellement en inscrivant un 1 logique.

TIMSK = Timer Interrupt Mask Register adr.-SRAM = 0x59, adr.-SFR = 0x39

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|--------|--------|--------|--------|--------|--------|-------------------------------|----------------------------------|
| OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 |
| Timer2 | Timer2 | Timer1 | Timer1 | Timer1 | Timer1 | Timer0 match 1 : libéré | Timer0 Overflow 1 : libéré |

OCIE0 Timer/Counter 0 Output Compare Match Interrupt Enable

- 0 Si ce bit est à l'état 1 on peut déclencher une interruption au moment où l'indicateur OCF0 (du registre TIFR) passe à l'état 1 (les contenus du registre de comptage TCNT0 et du registre de comparaison OCR0 correspondent) ou au moment où l'indicateur est mis à l'état 1 manuellement. Attention : les interruptions doivent être libérées globalement (I = 1 dans SREG avec l'instruction sei).
- 1 Interruption OCF0 pas permise.

TOIE0 Timer/Counter 0 Overflow Interrupt Enable

- 0 Si ce bit est à l'état 1 on peut déclencher une interruption au moment où l'indicateur TOV0 (du registre TIFR) passe à l'état 1, si donc il y a eu un débordement, ou si l'indicateur a été mis manuellement à l'état 1. Attention : les interruptions doivent être libérées globalement (I = 1 dans SREG avec l'instruction sei).
- 1 Interruption TOV0 pas permise.



Le OCR0 = Output Compare Register 0 adresse SRAM = 0x5C et adresse SFR = 0x3C est un compteur à 8 bits périodique. Attention : les commandes SBI, CBI, SBIS et SBIC ne peuvent pas être utilisées ! Les bits de ce registre vont de OCR00 à OCR07.

Le TCNT0 = Timer/Counter 0 adresse SRAM = 0x52 et adresse SFR = 0x32 est un compteur à 8 bits périodique. Attention : les commandes SBI, CBI, SBIS et SBIC ne peuvent pas être utilisées ! Les bits de ce registre vont de TCNT00 à TCNT07.

A.9 L'interface série

UCSRA (USART Control and Status Register A)

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-----|-----|------|----|-----|----|-----|------|
| UCSRA | RXC | TXC | UDRE | FE | DOR | PE | U2X | MPCM |
| Valeur début | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Read/ Write | R | R/W | R | R | R | R | R/W | R/W |

RXC USART Receive Complete

0 : si le registre tampon d'entrée est vide ou le récepteur est désactivé

1 : est mis à 1, si le registre tampon d'entrée n'est pas vide. Les données peuvent donc être lues.

TXC USART Transmit Complete

0 : il y a encore des données dans le registre de décalage ou le registre de données : Aucun caractère nouveau ne peut être transmis.

1 : est mis à 1, si toutes les données (toute la trame) a été sortie du registre de décalage et le registre UDR ne contient pas encore de données nouvelles. Ce bit doit être mis à 0 manuellement avant toute sortie. Il peut également donner lieu à une interruption (voir UCSRB).

UDRE USART Data Register Empty

0 : UDR n'est pas vide

1 : est mis à 1, si le UDR a été vidé et l'USART peut recevoir de nouvelles données. Ce bit peut également conduire à une interruption.

FE Frame Error

0 : pas d'erreur de trame

1 : erreur de trame, bit STOP pas valablement reconnu

DOR Data OverRun

0 : pas d'erreur de débordement de données

1 : erreur de débordement qui indique qu'un bit START est reconnu alors que les deux tampons RXB et UDR ne sont pas vides.

PE Parity Error

0 : pas d'erreur de parité

1 : erreur de parité



UCSRB (USART Control and Status Register B)

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-------|-------|-------|------|------|-------|------|------|
| UCSRB | RXCIE | TXCIE | UDRIE | RXEN | TXEN | UCSZ2 | RXB8 | TXB8 |
| Valeur début | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/ Write | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W |

RXCIE **RXC Interrupt Enable**

0: pas d'interruption RXC permise.

1 : permet une interruption qui survient au moment où l'indicateur RXC est mis à 1, donc s'il y a de nouvelles données dans le registre tampon de réception. Les interruptions doivent être libérées globalement (**sei →I=1 dans SREG**).

TXCIE **TXC Interrupt Enable**

0: pas d'interruption TXC permise.

1 : permet une interruption qui survient au moment où l'indicateur TXC est mis à 1, donc si le registre de décalage et le registre UDR sont vides. Les interruptions doivent être libérées globalement (**sei →I=1 dans SREG**).

UDRIE **UDRE Interrupt Enable**

0: pas d'interruption TXC permise.

1 : permet une interruption qui survient au moment où l'indicateur UDRE est mis à 1, donc si le registre UDR est vide. Les interruptions doivent être libérées globalement (**sei →I=1 dans SREG**).

RXEN **Receiver Enable**

0 : désactive le récepteur.

1 : met en service le récepteur. La broche (Pin) RxD (PORTD0) est réservé comme entrée et n'est plus accessible à d'autres applications. L'entrée n'a donc pas besoin d'être expressément initialisée.

TXEN **Transmitter Enable**

0 : désactive l'émetteur.

1 : met en service l'émetteur. La broche (Pin)TxD (PORTD1) est réservé comme sortie et n'est plus accessible à d'autres applications. La sortie n'a donc pas besoin d'être expressément initialisée.

UCSZ2 **USART Character Size 2**

Voir prochain registre UCSRC



UCSRC (USART Control and Status Register C)

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-------|-------|------|------|------|-------|-------|-------|
| UCSRC | URSEL | UMSEL | UPM1 | UPM0 | USBS | UCSZ1 | UCSZ0 | UCPOL |
| Valeur début | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Read/ Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

URSEL USART Register Select (la position de mémoire est occupée par 2 registres)

0: le registre UBRRH est choisi. Si ce registre est lu, le bit est 0.

1: le registre UCSRC est choisi (par défaut). Si ce registre est lu, le bit est 1

TXCIE USART Mode Select

0: mode asynchrone.

1: mode synchrone.

UPM USART Parity Mode : UPM1, UPM0

Si le bit de parité est mis (parité paire ou impaire) la parité est générée par l'équipement, puis insérée dans la trame. L'USART en tant que récepteur contrôle la parité. Si une erreur est apparue, l'indicateur PE (Parity Error) est mis dans UCSRA.

00: pas de parité.

01: réservée.

10: parité paire activée.

11: parité impaire activée.

USBS USART STOP Bit Select

0: 1 bit STOP.

1: 2 bits STOP.

UCSZ USART Character Size : UCSZ2 (voir UCSRB: par défaut 0), UCSZ1, UCSZ0

Ces trois bits déterminent le nombre de bits de données

210(UCSZ)

000 5 bit

001 6 bit

010 7 bit

011 8 bit

100 réservé

101 réservé

110 réservé

111 9 bit

UCPOL USART Clock Polarity

Seulement utilisé lors d'une transmission synchrone. Ce bit reste 0 pour la transmission asynchrone.

UBRRH (USART Baud Rate Register High)

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-------|---|---|---|--------|--------|-------|-------|
| UBRRH | URSEL | - | - | - | UBRR11 | UBRR10 | UBRR9 | UBRR8 |
| Valeur début | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/ Write | R/W | R | R | R | R/W | R/W | R/W | R/W |



UBRRL (USART Baud Rate Register Low)

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|
| UBRRL | UBRR7 | UBRR6 | UBRR5 | UBRR4 | UBRR3 | UBRR2 | UBRR1 | UBRR0 |
| Valeur début | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/ Write | R/W | R | R | R | R/W | R/W | R/W | R/W |

URSEL USART Register Select

La position de mémoire est occupée par 2 registres

1 : USCRRC est adressé

0 : UBRRH est adressé

UBRR USART Baud Rate Register

Les 12 bits représentent le diviseur qui permet de calculer la vitesse de pas. L'échantillonnage du signal d'entrée RxD se fait au moyen d'un multiple de 16 de la vitesse de pas. Les formules de calcul sont les suivantes :

$$1. \text{ vitesse de pas} = \frac{\text{fréquence de l'horloge système}}{16 \cdot (\text{diviseur} + 1)}$$

$$2. \text{ diviseur} = \frac{\text{fréquence de l'horloge système}}{16 \cdot \text{vitesse de pas}} - 1$$

$$3. \text{ erreur}(\%) = \left(\frac{\text{vitesse de pas calculée}}{\text{vitesse de pas standard}} - 1 \right) \cdot 100\%$$

On doit prendre soin que l'erreur reste en-dessous de 0,5%, sinon le nombre d'erreurs augmentera pendant la transmission. Ceci peut être réalisé en échangeant le quartz ou en doublant la vitesse de pas avec U2X dans le registre UCSRA.

UDR (USART I/O Data Register)

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|------|------|------|------|------|------|------|------|
| UDR R/W | RXB7 | RXB6 | RXB5 | RXB4 | RXB3 | RXB2 | RXB1 | RXB0 |
| | TXB7 | TXB6 | TXB5 | TXB4 | TXB3 | TXB2 | TXB1 | TXB0 |
| Valeur début | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read/ Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

TXB Transmit Data Buffer Register (registre tampon de transmission)

RXD Receive Data Buffer Register (registre tampon de réception)



A.10 Le convertisseur analogique – numérique (CAN)

ADMUX = ADC Multiplexer Selection

| | | | | | | | | | |
|--------------|-------|-------|-------|------|------|------|------|------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | ADMUX |
| Read / Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Startwert | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

A côté du multiplexeur de canaux ce registre fait également la sélection de la source de tension de référence et ce à l'aide des bits REFS0 et REFS1.

| REFS1 | REFS0 | description |
|-------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | 0 | Une tension de référence externe est branchée au Pin AREF, la source de tension interne est débranchée |
| 0 | 1 | Une tension de service du CAN au Pin AVCC est utilisée en tant que tension de référence. Elle se trouve au Pin AREF auquel doit être branché un condensateur. |
| 1 | 0 | Reserviert |
| 1 | 1 | Une tension de référence interne est branchée et est utilisée. Elle est branchée au Pin AREF auquel doit être branché un condensateur. |

Le bit ADLAR (ADC Left Adjust Result) décide si le résultat à 10 bits de la conversion se trouve dans le registre de données à 16 bits (ADCH et ADCL) en commençant par la gauche ou la droite.

ADLAR = 0

| | | | | | | | | | |
|-----|------|------|------|------|------|------|------|------|------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
| | - | - | - | - | - | - | ADC9 | ADC8 | ADCH |
| | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 | ADCL |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |

ADLAR = 1

| | | | | | | | | | |
|-----|------|------|------|------|------|------|------|------|------|
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
| | ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADCH |
| | ADC1 | ADC0 | - | - | - | - | - | - | ADCL |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |

Si on a besoin de la résolution maximale de 10 bits, il est favorable de mettre ADLAR = 0. Les 8 bits les moins significatifs sont mis dans ADCL et les 2 bits qui restent se trouvent aux positions 0 et 1 du registre ADCH. Les positions correspondent ainsi aux poids respectifs d'un nombre binaire à 16 bits.

Si au contraire on n'a besoin que d'une résolution de 8 bits, les deux bits les moins significatifs deviennent superflus de sorte que tous les bits sont décalés de 2 positions vers la droite pour qu'elles correspondent aux poids d'un nombre binaire à 8 bits. Si ADLAR = 1 cette adaptation a lieu automatiquement et de suite c'est-à-dire qu'elle est indépendante d'une éventuelle conversion en cours.



MUXx Analog Channel and Gain Selection Bits

Au moyen de 5 bits de commande on peut sélectionner 32 possibilités pour connecter une entrée du CAN à une entrée des entrées analogiques (voir tableau ci-dessous).

| MUX 4..0 | entrée asymétrique | entrée non-inversée | entrée inversée | amplification |
|----------|--------------------|---------------------|-----------------|---------------|
| 00000 | ADC0 | | | |
| 00001 | ADC1 | | | |
| 00010 | ADC2 | | | |
| 00011 | ADC3 | | | |
| 00100 | ADC4 | | | |
| 00101 | ADC5 | | | |
| 00110 | ADC6 | | | |
| 00111 | ADC7 | | | |
| 01000 | | ADC0 | ADC0 | 10x |
| 01001 | | ADC1 | ADC0 | 10x |
| 01010 | | ADC0 | ADC0 | 200x |
| 01011 | | ADC1 | ADC0 | 200x |
| 01100 | | ADC2 | ADC2 | 10x |
| 01101 | | ADC3 | ADC2 | 10x |
| 01110 | | ADC2 | ADC2 | 200x |
| 01111 | | ADC3 | ADC2 | 200x |
| 10000 | | ADC0 | ADC1 | 1x |
| 10001 | | ADC1 | ADC1 | 1x |
| 10010 | | ADC2 | ADC1 | 1x |
| 10011 | | ADC3 | ADC1 | 1x |
| 10100 | | ADC4 | ADC1 | 1x |
| 10101 | | ADC5 | ADC1 | 1x |
| 10110 | | ADC6 | ADC1 | 1x |
| 10111 | | ADC7 | ADC1 | 1x |
| 11000 | | ADC0 | ADC2 | 1x |
| 11001 | | ADC1 | ADC2 | 1x |
| 11010 | | ADC2 | ADC2 | 1x |
| 11011 | | ADC3 | ADC2 | 1x |
| 11100 | | ADC4 | ADC2 | 1x |
| 11101 | | ADC5 | ADC2 | 1x |
| 11110 | 1,22 V | | | |
| 11111 | 0 V (GND) | | | |



ADCSRA = ADC Control and Status Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------------|------|------|-------|------|------|-------|-------|-------|--------|
| | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | ADCSRA |
| Read / Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Startwert | 0 | 0 | 0 | - | - | - | - | - | |

ADSC

ADC Start Conversion

Ce bit sert à lancer une conversion. Cependant il peut également indiquer si une conversion est en cours de se faire.

- 0 Le bit devient 0 si la conversion est terminée. Le reset de ce bit ne joue aucun rôle.
- 1 En mode Single Conversion ce bit doit chaque fois être mis à 1 avant le lancement d'une conversion. En mode Free Running il suffit de mettre une fois à 1 pour lancer la première conversion. Le bit peut être mis en même temps que le bit ADEN !

ADATE

ADC Auto Trigger Enable

- 0 Un 0 logique termine le mode Auto Trigger.
- 1 Un 1 active le mode Auto Trigger. Une conversion est lancée avec le front montant du signal Trigger. La source de ce dernier est déterminé à l'aide des bits ADTS2 .. ADTS0 du registre spécial SFIOR.

ADIF

ADC Interrupt Flag

- 0 Il n'y pas de nouvelles données de conversion disponibles dans le registre de données.
- 1 Ce bit est 1 si une conversion est terminée et le registre de données (ADCH et ADCL) est actualisé. Si ADIF conduit à une interruption (ADIE et I à l'état 1) il est remis à 0 automatiquement après que la routine d'interruption a été exécutée. Le bit peut aussi être reseté manuellement en inscrivant un état 1 logique dans cette position de bit.

ADIE

ADC Interrupt Enable

- 0 La commande des interruptions de l'unité de conversion est débranchée.
- 1 Si ce bit est à l'état 1 et si l'indicateur I est à l'état 1 une interruption est lancée après que la conversion ait été terminée.



ADPS**ADC Prescaler Select Bits**

Ces bits définissent le facteur de division entre le signal horloge du système et celui du CAN.

| 2^2 | 2^1 | 2^0 | facteur de division |
|-------|-------|-------|---------------------|
| 0 | 0 | 0 | 2 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 8 |
| 1 | 0 | 0 | 16 |
| 1 | 0 | 1 | 32 |
| 1 | 1 | 0 | 64 |
| 1 | 1 | 1 | 128 |

SFIOR = Special Function IO Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------------|-------|-------|-------|---|------|-----|------|-------|-------|
| | ADTS2 | ADTS1 | ADTS0 | - | ACME | PUD | PSR2 | PSR10 | SFIOR |
| Read / Write | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | |
| Startwert | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Seulement les bits ADTS2 .. ADTS0 (ADC Auto Trigger Source) sont importants. Ils déterminent la source externe ou interne qui peut lancer une conversion. Pour cela le bit ADATE du registre ADCSRA doit être mis à l'état 1.

| ADTS | 2^2 | 2^1 | 2^0 | Trigger source |
|-------|-------|-------|-------|-------------------------------------|
| 0 0 0 | 0 | 0 | 0 | Free Running Mode |
| 0 0 1 | 0 | 0 | 1 | Analog Comparator |
| 0 1 0 | 0 | 1 | 0 | External Interrupt Request 0 (INT0) |
| 0 1 1 | 0 | 1 | 1 | Timer/Counter0 Compare Match |
| 1 0 0 | 1 | 0 | 0 | Timer/Counter0 Overflow |
| 1 0 1 | 1 | 0 | 1 | Timer/Counter1 Compare Match B |
| 1 1 0 | 1 | 1 | 0 | Timer/Counter1 Overflow |
| 1 1 1 | 1 | 1 | 1 | Timer/Counter1 Capture Event |



A.11 Set des instructions

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|------------------------------------------|----------|------------------------------------------|------------------------------------------------|------------|-----------|
| ARITHMETIC AND LOGIC INSTRUCTIONS | | | | | |
| ADD | Rd, Rr | Add two Registers | $Rd \leftarrow Rd + Rr$ | Z,C,N,V,H | 1 |
| ADC | Rd, Rr | Add with Carry two Registers | $Rd \leftarrow Rd + Rr + C$ | Z,C,N,V,H | 1 |
| ADIW | Rdl,K | Add Immediate to Word | $Rdh:Rdl \leftarrow Rdh:Rdl + K$ | Z,C,N,V,S | 2 |
| SUB | Rd, Rr | Subtract two Registers | $Rd \leftarrow Rd - Rr$ | Z,C,N,V,H | 1 |
| SUBI | Rd, K | Subtract Constant from Register | $Rd \leftarrow Rd - K$ | Z,C,N,V,H | 1 |
| SBC | Rd, Rr | Subtract with Carry two Registers | $Rd \leftarrow Rd - Rr - C$ | Z,C,N,V,H | 1 |
| SBCI | Rd, K | Subtract with Carry Constant from Reg. | $Rd \leftarrow Rd - K - C$ | Z,C,N,V,H | 1 |
| SBIW | Rdl,K | Subtract Immediate from Word | $Rdh:Rdl \leftarrow Rdh:Rdl - K$ | Z,C,N,V,S | 2 |
| AND | Rd, Rr | Logical AND Registers | $Rd \leftarrow Rd \cdot Rr$ | Z,N,V | 1 |
| ANDI | Rd, K | Logical AND Register and Constant | $Rd \leftarrow Rd \cdot K$ | Z,N,V | 1 |
| OR | Rd, Rr | Logical OR Registers | $Rd \leftarrow Rd \vee Rr$ | Z,N,V | 1 |
| ORI | Rd, K | Logical OR Register and Constant | $Rd \leftarrow Rd \vee K$ | Z,N,V | 1 |
| EOR | Rd, Rr | Exclusive OR Registers | $Rd \leftarrow Rd \oplus Rr$ | Z,N,V | 1 |
| COM | Rd | One's Complement | $Rd \leftarrow \sim Rd$ | Z,C,N,V | 1 |
| NEG | Rd | Two's Complement | $Rd \leftarrow \sim Rd + 1$ | Z,C,N,V,H | 1 |
| SBR | Rd,K | Set Bit(s) in Register | $Rd \leftarrow Rd \vee K$ | Z,N,V | 1 |
| CBR | Rd,K | Clear Bit(s) in Register | $Rd \leftarrow Rd \cdot (\sim K)$ | Z,N,V | 1 |
| INC | Rd | Increment | $Rd \leftarrow Rd + 1$ | Z,N,V | 1 |
| DEC | Rd | Decrement | $Rd \leftarrow Rd - 1$ | Z,N,V | 1 |
| TST | Rd | Test for Zero or Minus | $Rd \leftarrow Rd \cdot Rd$ | Z,N,V | 1 |
| CLR | Rd | Clear Register | $Rd \leftarrow Rd \oplus Rd$ | Z,N,V | 1 |
| SER | Rd | Set Register | $Rd \leftarrow \$FF$ | None | 1 |
| MUL | Rd, Rr | Multiply Unsigned | $R1:R0 \leftarrow Rd \times Rr$ | Z,C | 2 |
| MULS | Rd, Rr | Multiply Signed | $R1:R0 \leftarrow Rd \times Rr$ | Z,C | 2 |
| MULSU | Rd, Rr | Multiply Signed with Unsigned | $R1:R0 \leftarrow Rd \times Rr$ | Z,C | 2 |
| FMUL | Rd, Rr | Fractional Multiply Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$ | Z,C | 2 |
| FMULS | Rd, Rr | Fractional Multiply Signed | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$ | Z,C | 2 |
| FMULSU | Rd, Rr | Fractional Multiply Signed with Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$ | Z,C | 2 |
| BRANCH INSTRUCTIONS | | | | | |
| RJMP | k | Relative Jump | $PC \leftarrow PC + k + 1$ | None | 2 |
| IJMP | | Indirect Jump to (Z) | $PC \leftarrow Z$ | None | 2 |
| JMP | k | Direct Jump | $PC \leftarrow k$ | None | 3 |
| RCALL | k | Relative Subroutine Call | $PC \leftarrow PC + k + 1$ | None | 3 |
| ICALL | | Indirect Call to (Z) | $PC \leftarrow Z$ | None | 3 |
| CALL | k | Direct Subroutine Call | $PC \leftarrow k$ | None | 4 |
| RET | | Subroutine Return | $PC \leftarrow \text{Stack}$ | None | 4 |
| RETI | | Interrupt Return | $PC \leftarrow \text{Stack}$ | I | 4 |
| CPSE | Rd,Rr | Compare, Skip if Equal | if $(Rd = Rr) PC \leftarrow PC + 2$ or 3 | None | 1 / 2 / 3 |
| CP | Rd,Rr | Compare | $Rd - Rr$ | Z, N,V,C,H | 1 |
| CPC | Rd,Rr | Compare with Carry | $Rd - Rr - C$ | Z, N,V,C,H | 1 |
| CPI | Rd,K | Compare Register with Immediate | $Rd - K$ | Z, N,V,C,H | 1 |
| SBRC | Rr, b | Skip if Bit in Register Cleared | if $(Rr(b)=0) PC \leftarrow PC + 2$ or 3 | None | 1 / 2 / 3 |
| SBRS | Rr, b | Skip if Bit in Register is Set | if $(Rr(b)=1) PC \leftarrow PC + 2$ or 3 | None | 1 / 2 / 3 |
| SBIC | P, b | Skip if Bit in I/O Register Cleared | if $(P(b)=0) PC \leftarrow PC + 2$ or 3 | None | 1 / 2 / 3 |
| SBIS | P, b | Skip if Bit in I/O Register is Set | if $(P(b)=1) PC \leftarrow PC + 2$ or 3 | None | 1 / 2 / 3 |
| BRBS | s, k | Branch if Status Flag Set | if $(SREG(s) = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRBC | s, k | Branch if Status Flag Cleared | if $(SREG(s) = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BREQ | k | Branch if Equal | if $(Z = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRNE | k | Branch if Not Equal | if $(Z = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRCS | k | Branch if Carry Set | if $(C = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRCC | k | Branch if Carry Cleared | if $(C = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRSH | k | Branch if Same or Higher | if $(C = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRLO | k | Branch if Lower | if $(C = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRMI | k | Branch if Minus | if $(N = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRPL | k | Branch if Plus | if $(N = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRGE | k | Branch if Greater or Equal, Signed | if $(N \oplus V = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRLT | k | Branch if Less Than Zero, Signed | if $(N \oplus V = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRHS | k | Branch if Half Carry Flag Set | if $(H = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRHC | k | Branch if Half Carry Flag Cleared | if $(H = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRTS | k | Branch if T Flag Set | if $(T = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRTC | k | Branch if T Flag Cleared | if $(T = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRVS | k | Branch if Overflow Flag is Set | if $(V = 1) PC \leftarrow PC + k + 1$ | None | 1 / 2 |
| BRVC | k | Branch if Overflow Flag is Cleared | if $(V = 0) PC \leftarrow PC + k + 1$ | None | 1 / 2 |



| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|--------------------------------------|----------|----------------------------------|------------------------------------------|---------|---------|
| BRIE | k | Branch if Interrupt Enabled | if (I = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRID | k | Branch if Interrupt Disabled | if (I = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| DATA TRANSFER INSTRUCTIONS | | | | | |
| MOV | Rd, Rr | Move Between Registers | Rd ← Rr | None | 1 |
| MOVW | Rd, Rr | Copy Register Word | Rd+1:Rd ← Rr+1:Rr | None | 1 |
| LDI | Rd, K | Load Immediate | Rd ← K | None | 1 |
| LD | Rd, X | Load Indirect | Rd ← (X) | None | 2 |
| LD | Rd, X+ | Load Indirect and Post-Inc. | Rd ← (X), X ← X + 1 | None | 2 |
| LD | Rd, -X | Load Indirect and Pre-Dec. | X ← X - 1, Rd ← (X) | None | 2 |
| LD | Rd, Y | Load Indirect | Rd ← (Y) | None | 2 |
| LD | Rd, Y+ | Load Indirect and Post-Inc. | Rd ← (Y), Y ← Y + 1 | None | 2 |
| LD | Rd, -Y | Load Indirect and Pre-Dec. | Y ← Y - 1, Rd ← (Y) | None | 2 |
| LDD | Rd, Y+q | Load Indirect with Displacement | Rd ← (Y + q) | None | 2 |
| LD | Rd, Z | Load Indirect | Rd ← (Z) | None | 2 |
| LD | Rd, Z+ | Load Indirect and Post-Inc. | Rd ← (Z), Z ← Z+1 | None | 2 |
| LD | Rd, -Z | Load Indirect and Pre-Dec. | Z ← Z - 1, Rd ← (Z) | None | 2 |
| LDD | Rd, Z+q | Load Indirect with Displacement | Rd ← (Z + q) | None | 2 |
| LDS | Rd, k | Load Direct from SRAM | Rd ← (k) | None | 2 |
| ST | X, Rr | Store Indirect | (X) ← Rr | None | 2 |
| ST | X+, Rr | Store Indirect and Post-Inc. | (X) ← Rr, X ← X + 1 | None | 2 |
| ST | -X, Rr | Store Indirect and Pre-Dec. | X ← X - 1, (X) ← Rr | None | 2 |
| ST | Y, Rr | Store Indirect | (Y) ← Rr | None | 2 |
| ST | Y+, Rr | Store Indirect and Post-Inc. | (Y) ← Rr, Y ← Y + 1 | None | 2 |
| ST | -Y, Rr | Store Indirect and Pre-Dec. | Y ← Y - 1, (Y) ← Rr | None | 2 |
| STD | Y+q, Rr | Store Indirect with Displacement | (Y + q) ← Rr | None | 2 |
| ST | Z, Rr | Store Indirect | (Z) ← Rr | None | 2 |
| ST | Z+, Rr | Store Indirect and Post-Inc. | (Z) ← Rr, Z ← Z + 1 | None | 2 |
| ST | -Z, Rr | Store Indirect and Pre-Dec. | Z ← Z - 1, (Z) ← Rr | None | 2 |
| STD | Z+q, Rr | Store Indirect with Displacement | (Z + q) ← Rr | None | 2 |
| STS | k, Rr | Store Direct to SRAM | (k) ← Rr | None | 2 |
| LPM | | Load Program Memory | R0 ← (Z) | None | 3 |
| LPM | Rd, Z | Load Program Memory | Rd ← (Z) | None | 3 |
| LPM | Rd, Z+ | Load Program Memory and Post-Inc | Rd ← (Z), Z ← Z+1 | None | 3 |
| SPM | | Store Program Memory | (Z) ← R1:R0 | None | - |
| IN | Rd, P | In Port | Rd ← P | None | 1 |
| OUT | P, Rr | Out Port | P ← Rr | None | 1 |
| PUSH | Rr | Push Register on Stack | Stack ← Rr | None | 2 |
| POP | Rd | Pop Register from Stack | Rd ← Stack | None | 2 |
| BIT AND BIT-TEST INSTRUCTIONS | | | | | |
| SBI | P,b | Set Bit in I/O Register | I/O(P,b) ← 1 | None | 2 |
| CBI | P,b | Clear Bit in I/O Register | I/O(P,b) ← 0 | None | 2 |
| LSL | Rd | Logical Shift Left | Rd(n+1) ← Rd(n), Rd(0) ← 0 | Z,C,N,V | 1 |
| LSR | Rd | Logical Shift Right | Rd(n) ← Rd(n+1), Rd(7) ← 0 | Z,C,N,V | 1 |
| ROL | Rd | Rotate Left Through Carry | Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7) | Z,C,N,V | 1 |
| ROR | Rd | Rotate Right Through Carry | Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0) | Z,C,N,V | 1 |
| ASR | Rd | Arithmetic Shift Right | Rd(n) ← Rd(n+1), n=0..6 | Z,C,N,V | 1 |
| SWAP | Rd | Swap Nibbles | Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0) | None | 1 |
| BSET | s | Flag Set | SREG(s) ← 1 | SREG(s) | 1 |
| BCLR | s | Flag Clear | SREG(s) ← 0 | SREG(s) | 1 |
| BST | Rr, b | Bit Store from Register to T | T ← Rr(b) | T | 1 |
| BLD | Rd, b | Bit load from T to Register | Rd(b) ← T | None | 1 |
| SEC | | Set Carry | C ← 1 | C | 1 |
| CLC | | Clear Carry | C ← 0 | C | 1 |
| SEN | | Set Negative Flag | N ← 1 | N | 1 |
| CLN | | Clear Negative Flag | N ← 0 | N | 1 |
| SEZ | | Set Zero Flag | Z ← 1 | Z | 1 |
| CLZ | | Clear Zero Flag | Z ← 0 | Z | 1 |
| SEI | | Global Interrupt Enable | I ← 1 | I | 1 |
| CLI | | Global Interrupt Disable | I ← 0 | I | 1 |
| SES | | Set Signed Test Flag | S ← 1 | S | 1 |
| CLS | | Clear Signed Test Flag | S ← 0 | S | 1 |
| SEV | | Set Twos Complement Overflow | V ← 1 | V | 1 |
| CLV | | Clear Twos Complement Overflow | V ← 0 | V | 1 |
| SET | | Set T in SREG | T ← 1 | T | 1 |
| CLT | | Clear T in SREG | T ← 0 | T | 1 |
| SEH | | Set Half Carry Flag in SREG | H ← 1 | H | 1 |
| CLH | | Clear Half Carry Flag in SREG | H ← 0 | H | 1 |
| MCU CONTROL INSTRUCTIONS | | | | | |
| NOP | | No Operation | | None | 1 |
| SLEEP | | Sleep | (see specific descr. for Sleep function) | None | 1 |
| WDR | | Watchdog Reset | (see specific descr. for WDR/timer) | None | 1 |
| BREAK | | Break | For On-Chip Debug Only | None | N/A |

