

## Периферійний послідовний інтерфейс UART

Схема периферійного послідовного порта UART наведена на рисунку 1

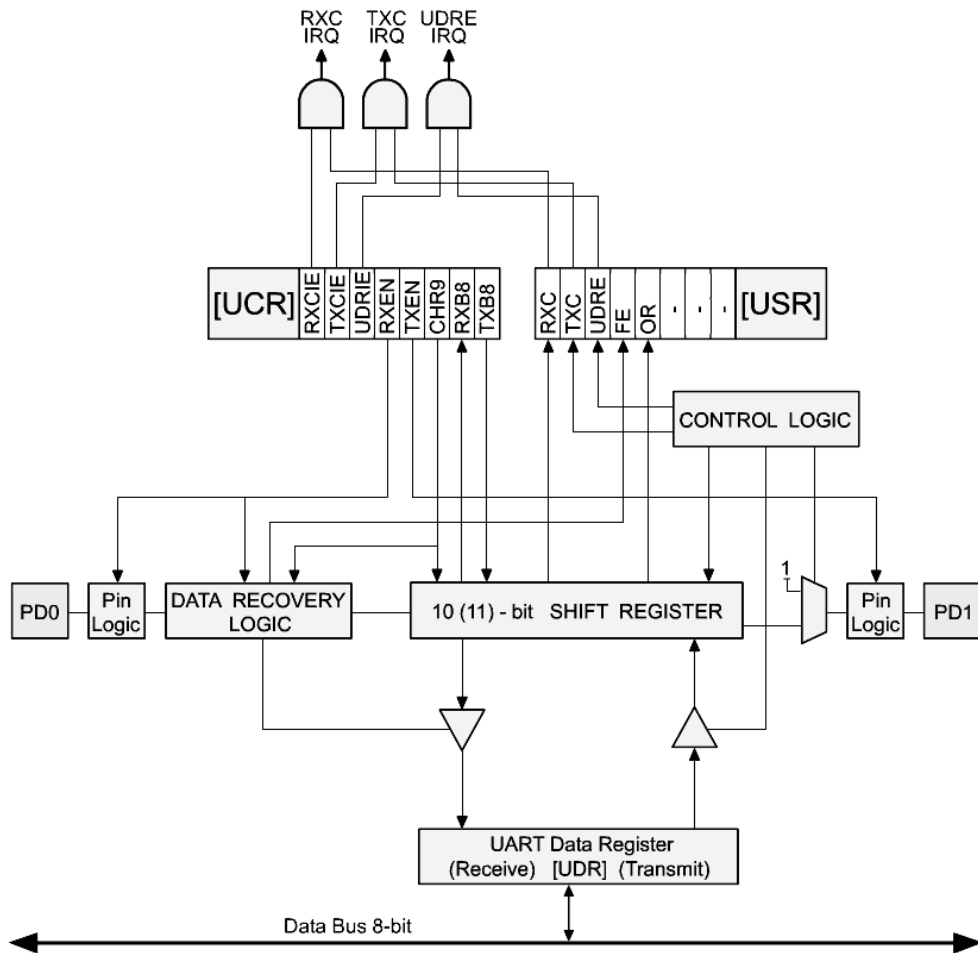


Рисунок 1 – Схема периферійного послідовного порта UART

До складу мікроконтролера AT90S2313 входить універсальний дуплексний послідовний порт (UART). Його основні можливості: широкий діапазон швидкостей обміну даними; висока швидкість передачі при низькій частоті XTAL; 8 або 9-розрядний формат даних; виявлення помилок утрати даних при прийомі; виявлення помилок формату кадрів; виявлення помилкового стартового біта; три окремих переривання: по завершенню передачі, по порожньому регістру передавача і по завершенню прийому.

При передачі модуль UART додає до вхідного символу (8 або 9 біт) на початку — старт-біт (нуль), а в кінці — стоп-біт (одиниця), формуючи таким чином 10- або 11-бітову послідовність. Отримані значення передаються до

регістра зсуву, який по черзі передає біти на вихід передавача TXD (вивід PD1). Швидкість видачі біт на вихід передавача визначається параметром *baud rate* (швидкість передачі інформації; вимірюється в бодах), яким можна керувати.

Приймач модуля UART безперервно перевіряє стан входу RXD, на якому за відсутності даних встановлюється рівень «1». Приймач зчитує інформацію з входу в 16 разів швидше. При виявленні на виводі RXD рівня «0» (тобто можливого старт-біта) мікроконтролер пропускає шість відліків, а потім робить три вибірки. Ці вибірки доводяться на відліки 8, 9 і 10 для кожного біта, що приймається, і, таким чином, зчитування значення біта відбувається в середині інтервалу його передачі, що дозволяє працювати з сигналами, що мають фронти великої тривалості. Якщо мікроконтролер виявляє, що на виводі RXD все ще присутній рівень «0», тобто прийшов стар-біт, модуль UART переходить в робочий режим і починає зчитувати байт. Якщо ж на виводі RXD вже присутній рівень «1», вважається, що перший відлік був просто шумом, і модуль переходить до очікування коректного символу. Якщо приймач визначив, що прийшов дійсний символ, він починає брати по три відліки кожного біта в середині інтервалу його передачі. Якщо значення всіх трьох відліків біта не збігаються, то значення біта набуває рівним значенню двох однакових відліків. На завершення модуль зчитує вибірки, що відносяться до стоп-біту. Для того, щоб було вирішено про коректний прийом символу, принаймні, дві з цих вибірок мають дорівнювати одиниці. Інакше модуль вважає символ за невірно кадрований і реєструє помилку кадрування (*framing error*).

Всі ці операції виконує модуль UART. Цей модуль містить 4 регістри вводу/виводу:

- UDR (UART Data Register \$0C) — регістр даних UART, містить байт даних, що приймається або передається;
- UCR (UART Control Register \$0A) — регістр керування, управляє конфігурацією UART, а також містить 8-й біт даних;

- **USR (UART Status Register \$0B)** — реєстр статусу UART, відображає стан модуля UART, зокрема прапорів переривань;
- **UBRR (UART Baud Rate Register, \$09)** — реєстр швидкості передачі, задається швидкість передачі даних по UART.

Швидкість передачі визначається за формулою:

$$BAUD = FCK / 16(UBRR + 1),$$

де **BAUD** - частота в бодах;

**FCK** - частота мікроконтролера;

**UBRR** - уміст реєстра UBRR (0 -255).

Якщо використовується резонатор 4 МГц, а в реєстрі UBRR записане 25, швидкість передачі дорівнює 9615 біт/с. Існує певний набір стандартних швидкостей передачі: 2400, 4800, 9600, якого бажано дотримуватися для забезпечення сумісності з іншими пристроями. З цієї причини «рівні» значення тактових частот, наприклад 4 МГц, не дуже підходять для пристроїв, що, використовують UART, оскільки в цьому випадку важко набути точних стандартних значень швидкості. Набагато зручніше використовувати резонатори з такими частотами, як 1.8432 МГц, 2.4576 МГц, 3.6864 МГц, 4.608 МГц, 7.3728 МГц і 9.216 МГц. Якщо взяти частоту 3.6864 МГц, то при UBRR= 23 швидкість передачі буде 9600.

Призначення біт реєстрів UCR і USR приведені у таблиці 1, 2. Реєстр UBRR використовується для управління швидкістю передачі даних. Вона має бути однаковою для приймаючого і передавального пристроїв.

Таблиця 1 – UCR реєстр керування UART (\$0A)

Біт	7	6	5	4	3	2	1	0
Назва	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8

**TXB8** – 8-й біт даних, що передаються (при обміні 9-бітними даними містить 9-біт символа, що передається);

**RXB8** – 8-й біт даних, що приймаються (при обміні 9-бітними даними містить 9-біт символа, що приймається);

*CHR9* – формат посилок (0– 8 бітні данні плюс старт-біт або стоп-біт; 1– 9 бітні данні плюс старт-біт або стоп-біт);

*TXEN* – дозвіл передачі (0- заборона але тільки після завершення поточної передачі, 1 – дозволена робота передавача)

*RXEN* – дозвіл прийому (0- заборона, 1 – дозволена робота приймача);

*UDRIE* – дозвіл переривання, коли очищений регістр даних UART (0 – переривання заборонено, 1 – дозволено);

*TXCIE* – дозвіл переривання по завершенню передачі (0 – переривання заборонено, 1 – дозволено)

*RXCIE* – дозвіл переривання по завершенню прийому (0 – переривання заборонено, 1 – дозволено)

Таблиця 2 – USR регістр стану UART (\$0B)

Біт	7	6	5	4	3	2	1	0
Назва	RXC	TXC	UDRE	FE	OR	-	-	-

*OR* – переповнення: 0–зміст UDR був вдало перезавантажений у регістр зсуву, 1– зміст UDR було перезаписано до того, як байт скопійований у регістр зсуву;

*FE* –помилка кадрування: 0– помилка відсутня (коректний стоп-біт), 1 – помилка виявлена (некоректний стоп-біт)

*UDRE* – регістр даних UART пуст: 0 – байт, що знаходиться в регістрі UDR ще не перевантажений до регістра зсуву, 1 – зміст регістра UDR перевантажений до регістра зсуву

*TXC* – передача завершена (встановлюється в 1 після передачі символу, якщо відсутні інші дані у UDR)

*RXC* – прийом завершений (встановлюється в 1 після прийому символу й збережені його у UDR)

\$007    rjmp UART\_RXC;    Оброблювач завершення прийому UART (UART RX Complete Handler)

\$008    rjmp UART\_DRE;    Оброблювач порожнього регістра даних (UDR Empty Handler)

\$009 rjmp UART\_TXC; Оброблювач завершення передачі UART (UART TX)  
Приклад налаштування приймача та передавача UART

; Дозволити роботи приймама UART  
; Дозволити переривання по завершенню прийому

*;Set PULL-UP Resistors*

sbi PORTD,0

sbi PORTD,1

*;Set Baud Rate*

Ldi R17,47

out UBRR, r17

*;Set Control Register*

Ldi R16, \$90

Out UCR,r16

*;Interrupt handler*

UART\_UDRE:

in R16, UDR

reti

; Дозволити роботи передавача UART  
; Дозволити переривання коли очищений регістр даних

*;Set PULL-UP Resistors*

sbi PORTD,0

sbi PORTD,1

*;Set Baud Rate*

Ldi R17,47

out UBRR, r17

*;Set Control Register*

Ldi R16, \$28

Out UCR,r16

*;Interrupt handler*

UART\_TXC:

- - program code - -

reti

*;Interrupt handler*

UART\_UDRE:

Out UDR, R16

reti

## Приклад програми передавача та приймача по інфрачервоному каналу для кодового замка з використанням UART

### **; Передача 5 байт по UART**

```
.include "tn2313def.inc"
; тактова частота 3.6864 МГц

.ORG 000
.equ s0 = 0b00100100
.equ s1 = 0b11000000 ;
.equ s2 = 0b11100000 ;
.equ s3 = 0b11110000 ;
.equ s4 = 0b11111000
.def tmp= r16
rjmp reset
reset:
ldi tmp, 0b00001001
out UCR, tmp ; установлений біт TXEN=1 (дозвіл передачі)
; TXB8=1 (після байту) передається 1
ldi tmp, 23 ; налаштування швидкості передачі
out UBRR, tmp ; BAUD=9600
ldi tmp, low(RAMEND)
out SPL, tmp
ldi tmp, 0b00000001;
out ddrb, tmp ;
out portb, tmp ;
scan:
ldi tmp, 0b00000000;
sbis pinb, 0 ;
rjmp load ;
rcall trans ;
rjmp scan ;
load: ;
ldi tmp, s0 ;
rcall trans
ldi tmp, s1 ;
rcall trans ;
ldi tmp, s2
rcall trans ;
ldi tmp, s3 ;
rcall trans ;
ldi tmp, s4
rcall trans
rjmp scan ;
trans: ;
sbis USR, UDRE ; якщо регістр даних UART пустий, то в регістр
rjmp trans ; UDR завантажуюмо нове число
out UDR, tmp ;
ret ;
.EXIT
```

### **; Прийом 5 байт по UART для кодового замка**

```
.include "tn2313def.inc"
.ORG 0
.equ s0 = 0b00100100
.equ s1 = 0b11000000 ;
```

```

.equ    s2 = 0b11100000 ;
.equ    s3 = 0b11110000 ;
.equ    s4 = 0b11111000 ;
.def    tmp = r16
rjmp   reset
reset:
;===== Ініціалізація стека =====
    ldi tmp, low(RAMEND)
    out SPL, tmp
;===== Ініціалізація UART =====
    ldi tmp,0b00010001 ; установлений біт TXEN=1 (дозвіл передачі)
    out UCR, tmp      ; TXB8=1 (після байту) передається 1
    ldi tmp,23        ;налаштування швидкості передачі
    out UBRR, tmp     ; BAUD=9600
;===== Ініціалізація портів В и D.=====
    ldi tmp,0b11111111
    out ddrb, tmp
    ldi tmp,0b11111100
    out ddrd, tmp
    cbi portd,5       ;закриваємо замок
    ldi r25, 0b00000000
    rjmp main
;===== Основний цикл =====
; прийом даних по UART
scan:
    sbis USR, RXC ; якщо прийом завершений, то RXC=1
    rjmp scan
    in tmp, UDR   ; то читаємо дані з регістра даних UDR
    ret
main: ;Порівнюємо прийняті слова з кодами
ss0:
    rcall scan
    cpi tmp, s0
    brne ss0
ss1:
    rcall scan
    cpi tmp,s1
    brne ss0
ss2:
    rcall scan
    cpi tmp,s2
    brne ss0
ss3:
    rcall scan
    cpi tmp,s3
    brne ss0
ss4:
    rcall scan
    cpi tmp,s4
    brne ss0
;===== Якщо все співпало, то
    sbi portd,5       ; відкриваємо замок
    ldi r27,20
    cbi portd,5       ; закриття замку
    rjmp main        ; повернення до початку
.EXIT

```