

**UART0**  
**Cổng truyền thông bất đồng bộ nối tiếp**  
**--φΠφ--**

**Các đề mục chính:**

<b>A. Chức năng:</b>	<b>3</b>
<b>B. Diễn tả các chân chức năng:</b>	<b>3</b>
<b>C. Diễn tả các thanh ghi chức năng:</b>	<b>3</b>
C.I. Thanh ghi đệm truyền thu <U0RBR - 0xE000C000>:	3
C.II. Thanh ghi giữ giá trị truyền phát <U0THR - 0xE000C000>:	3
C.III. Thanh ghi cho phép ngắt truyền thông nối tiếp <U0IER - 0xE000C004>:	4
C.IV. Thanh ghi nhận dạng ngắt truyền thông nối tiếp <U0IIR - 0xE000C008>:	4
C.V. Thanh ghi điều khiển FIFO cho truyền thông nối tiếp <U0FCR - 0xE000C008>:	5
C.VI. Thanh ghi điều khiển đường truyền cho truyền thông nối tiếp <U0LCR - 0xE000C00C>:	5
C.VII. Thanh ghi điều khiển trạng thái cho đường truyền cho truyền thông nối tiếp <U0LSR - 0xE000C014>:	6
<b>D. Kiến trúc của bộ truyền thông nối tiếp UART0:</b>	<b>7</b>
D.I. Hình vẽ:	7
D.II. Chú thích:	8
<b>E. Chương trình minh họa: “Hello World!”</b>	<b>8</b>
E.I. Mã nguồn:	8
I.1. Khai báo:	8
I.2. Chương trình chính:	9
Hàm Serial.c:	9
E.II. Kết quả mô phỏng:	9
<b>F. Kết luận:</b>	<b>10</b>

## Danh mục các bảng:

Bảng 1: Diễn tả các chân chức năng của UART0 .....	3
Bảng 2: Thanh ghi đệm truyền thụ (U0RBR - 0xE000C000 khi DLAB = 0, chỉ đọc).....	3
Bảng 3: Thanh ghi giữ giá trị truyền phát <U0THR-0xE000C000 khi DLAB = 0, chỉ ghi>.....	3
Bảng 4: Byte thấp bộ chia sinh tốc độ baud cho UART0 (U0DLL-0xE000C000 khi DLAB=1) .....	3
Bảng 5: Byte cao bộ chia sinh tốc độ baud cho UART0 (U0DLL-0xE000C004 khi DLAB=1) .....	4
Bảng 6: Giải thích các bit của thanh ghi cho phép ngắt UART0.....	4
Bảng 7: Thanh ghi nhận dạng các ngắt của truyền thông nối tiếp (U0IIR - 0xE000C008, chỉ đọc) 4	
Bảng 8: Thường trình ngắt UART0 .....	5
Bảng 9: Ý nghĩa các bit của thanh ghi điều khiển UART0 FIFO .....	5
Bảng 10: Ý nghĩa các bit của thanh ghi điều khiển đường truyền UART0(U0LCR - 0xE000C00C) ..	6
Bảng 11: Ý nghĩa các bit của thanh ghi trạng thái đường truyền truyền thông nối tiếp UART0 .....	7

## Danh mục hình vẽ:

Hình 1: Sơ đồ khối bộ truyền thông nối tiếp bất đồng bộ UART0 .....	7
Hình 2: Kết quả mô phỏng .....	10
Hình 3: Kết quả thực tế .....	10

Khởi tạo: 200602221515

A. Chức năng:

- + Thu phát 16 bit FIFOs.
- + Vùng thanh ghi hợp chuẩn công nghiệp 550
- + Nhận điểm ngắt FIFO tại 1,4,8 và 14 bytes
- + Tích hợp sẵn bộ sinh tốc độ Baud.

B. Diễn tả các chân chức năng:

Tên chân	Hướng truyền	Chức năng
RxD0	Vào	<b>Serial Input.</b> Nhận dữ liệu nối tiếp.
TxD0	Ra	<b>Serial Output.</b> Truyền dữ liệu nối tiếp.

**Bảng 1: Diễn tả các chân chức năng của UART0**

C. Diễn tả các thanh ghi chức năng:

UART0 có 10 thanh ghi chức năng 8 bit.

C.I. Thanh ghi đệm truyền thu <U0RBR - 0xE000C000>:

U0RBR là byte trên cùng của UART0 Rx FIFO, byte này chứa các kí tự truyền nhận sau cùng và có thể đọc được nhờ bus giao tiếp. Bit 0 (LSB) là bit cuối cùng của dữ liệu nhận, nếu kí tự truyền nhận ít hơn 8 bit, các bit cao của thanh ghi này sẽ tự cập nhật giá trị '0'

Bit DLAB(Divisor Latch Access Bit) trong thanh ghi U0LCR phải bằng 0 để có thể truy cập U0RBR. Thanh ghi đệm của truyền thu là thanh ghi chỉ đọc.

U0RBR	Chức năng	Giải thích chức năng	Giá trị Reset
7:0	Thanh ghi đệm truyền thu	Thanh ghi đệm truyền thu chứa byte thu nhận sau nhất của UART0 Rx FIFO	Không biết

**Bảng 2: Thanh ghi đệm truyền thu (U0RBR - 0xE000C000 khi DLAB = 0, chỉ đọc)**

C.II. Thanh ghi giữ giá trị truyền phát <U0THR - 0xE000C000>:

Thanh ghi U0THR là thanh ghi chứa byte trên cùng của UART Tx FIFO. Thanh ghi này chứa kí tự gần nhất của quá trình truyền phát FIFO, có thể ghi giá trị vào thanh ghi này thông qua giao tiếp bus. Bit LSB là bit đầu tiên sẽ được truyền phát.

Bit DLAB trong thanh ghi U0LCR phải bằng 0 để có thể truy cập U0THR. Thanh ghi đệm của truyền thu là thanh ghi chỉ được phép ghi.

U0THR	Chức năng	Giải thích chức năng	Giá trị Reset
7:0	Thanh ghi giữ giá trị truyền phát	Ghi giá trị vào thanh ghi giữ giá trị truyền thu UART0 thì byte đó sẽ được xếp sau cùng vào FIFO.	N/A

**Bảng 3: Thanh ghi giữ giá trị truyền phát <U0THR-0xE000C000 khi DLAB = 0, chỉ ghi>**

+Thanh ghi UART0 Divisor Latch LSB (U0DLL - 0xE000C000) truy cập được khi DLAB = 1.

+Thanh ghi UART0 Divisor Latch MSB (U0DLM - 0xE000C004) truy cập được khi DLAB = 1.

+UART0 Divisor Latch là số chia của bộ tạo tốc độ Baud và giá trị của nó được dùng để chia xung clock VPB (PCLK-Xung nhịp hoạt động của ngoại vi) thành tốc độ baud, tốc độ xung nhịp ngoại vi bằng 16x tốc độ baud mong muốn. Hai thanh ghi U0DLL và U0DLM lập thành số chia 16 bit, U0DLL chứa 8 bit thấp và U0DLM chứa 8 bit cao. Giá trị U0DLM-U0DLL = 0x0000 được xem như 0x0001 < vì giá trị chia cho 0 không tồn tại > Bit DLAB=1 để có thể truy cập UART0 Divisor Latches.

U0DLL	Chức năng	Giá trị Reset
7:0	Byte thấp của bộ chia sinh tốc độ baud	0x01

**Bảng 4: Byte thấp bộ chia sinh tốc độ baud cho UART0 (U0DLL-0xE000C000 khi DLAB=1)**

U0DLM	Chức năng	Giá trị Reset
7:0	Byte cao của bộ chia sinh tốc độ baud	0x00

**Bảng 5: Byte cao bộ chia sinh tốc độ baud cho UART0 (U0DLL-0xE000C004 khi DLAB=1)**

Nói một cách vắn tắt, để thiết lập tốc độ baud mới, phải cho bit DLAB='1', truy cập các thanh ghi U0DLM- U0DLL, khi cho DLAB='0', khởi tạo bộ tạo tốc độ Baud. Và lúc đó, mới có thể truy cập các giá trị từ bộ đếm thu hoặc bộ giữ phát của UART0.

**C.III. Thanh ghi cho phép ngắt truyền thông nối tiếp <U0IER - 0xE000C004>:**

Thanh ghi U0IER được dùng để cho phép các ngắt của UART0(4 nguồn ngắt).

U0IER	Chức năng	Diễn giải các chức năng	Giá trị Reset
0	Cho phép ngắt RBR	<ul style="list-style-type: none"> <li>o U0IER[0] = '0': Không cho phép ngắt RDA.</li> <li>o U0IER[0] = '1': Cho phép ngắt RDA</li> </ul> Bit này cũng điều khiển ngắt do thời gian chờ truyền nhận kí tự bị Time-out.	0
1	Cho phép ngắt THRE	<ul style="list-style-type: none"> <li>o U0IER[1] = '0': Không cho phép ngắt THRE.</li> <li>o U0IER[1] = '1': Cho phép ngắt THRE</li> </ul> Trạng thái của bit ngắt này có thể được đọc từ thanh ghi U0LSR[5].	0
2	Cho phép ngắt từ đường trạng thái Rx	<ul style="list-style-type: none"> <li>o U0IER[2] = '0': Không cho phép ngắt trạng thái đường Rx</li> <li>o U0IER[2] = '1': Cho phép ngắt trạng thái đường Rx bị lỗi.</li> </ul> Trạng thái của bit ngắt này có thể được đọc từ thanh ghi U0LSR[4:1]	0
7:3	Dự trữ	Dự trữ, không được sử dụng	NA

**Bảng 6: Giải thích các bit của thanh ghi cho phép ngắt UART0**

**C.IV. Thanh ghi nhân dạng ngắt truyền thông nối tiếp<U0IIR - 0xE000C008>:**

U0IIR	Chức năng	Diễn giải các chức năng	Giá trị Reset
0	Chờ ngắt	<ul style="list-style-type: none"> <li>o U0IIR[0] = '0': Có ít nhất 1 ngắt đang chờ</li> <li>o U0IIR[0] = '1': không có ngắt nào đang chờ</li> </ul> Lưu ý rằng U0IIR0 tích cực ở mức thấp. Các trạng thái ngắt có thể được diễn tả bởi các giá trị trong 3 bit trạng thái U0IER3:1.	1
3:1	Nhận dạng ngắt	<ul style="list-style-type: none"> <li>o U0IIR[3:1] = '011': Có ngắt do trạng thái đường truyền UART lên mức tích cực(RLS).</li> <li>o U0IIR[3:1] = '010' 2a: Báo hiệu dữ liệu nhận sẵn sàng</li> <li>o U0IIR[3:1] = '110' 2b: Báo hiệu thời gian chờ truyền nhận kí tự bị Time-out. (CTI)</li> <li>o U0IIR[3:1] ='001' 3: Ngắt THRE</li> <li>o Các tổ hợp bit khác đều không hợp lệ.</li> </ul> U0IER[3:1] nhận dạng ngắt đáp ứng với UART0 Rx FIFO.	0
5:4	Dự trữ	Dự trữ, không được sử dụng	NA
7:6	Cho phép FIFO	2 bit này tương đương với U0FCR0.	0

**Bảng 7: Thanh ghi nhận dạng các ngắt của truyền thông nối tiếp (U0IIR - 0xE000C008, chỉ đọc)**

Các thường trình ngắt được đề cập ở bảng sau. Biết các giá trị của U0IIR[3:0], thường trình ngắt có thể biết nguồn sinh ngắt và cách để xóa ngắt đó. Thanh ghi U0IIR phải được đọc để xóa các ngắt ưu tiên và thoát khỏi thường trình ngắt.

Ngắt do đường truyền nhận UART bị lỗi là ngắt có độ ưu tiên cao nhất và nó có thể tích cực khi 1 trong 4 nguyên nhân gây lỗi Rx xuất hiện: lỗi tràn(OE), sai parity(PE), sai framing(FE) và break interrupt (BI). Điều kiện gây lỗi UART0 Rx tạo ngắt có thể được biết nhờ tổ hợp bit U0LSR[4:1]. Thường trình ngắt sẽ xóa các bit này ngay sau khi đọc U0LSR.

Ngắt RDA (U0IIR3:1=010) cùng mức ưu tiên (mức 2) với ngắt CTI(U0IIR3:1=110).

Ngắt RDA tích cực khi UART Rx FIFO bằng mức kích định nghĩa bởi U0FCR[7:6] và nó được xóa nhờ UART0 Rx FIFO xuống thấp hơn mức kích. Khi ngắt này tích cực, CPU có thể đọc khối dữ liệu được định nghĩa bởi mức kích.

Ngắt CTI tích cực khi khi UART0 Rx FIFO chứa ít nhất 1 ký tự và không có UART0 Rx FIFO tích cực xuất hiện trong khoảng thời gian truyền 3.5 đến 4.5 ký tự. Nguồn ngắt UART0 Rx FIFO tích cực (đọc hoặc ghi UART0 RSR) sẽ xóa ngắt này.

Thường trình phục vụ ngắt UART0 RBR sẽ hoàn tất khi dữ liệu truyền thông đã được nhận mà không vi phạm các mức kích.

*Vi dụ:* Ngoại vi cần nhận 105 ký tự và mức kích ở 10 ký tự, CPU sẽ nhận được 10 ngắt RDA (tức là 100 ký tự) còn lại 5 ký tự chưa được nhận, sẽ sinh ra ngắt CTI, và mức kích cần phải điều chỉnh (ở 1 đến 5 ký tự) để hoàn tất quá trình truyền nhận.

U0IIR[3:0]	Mức ưu tiên	Loại ngắt	Nguồn ngắt	Điều kiện để xóa ngắt
0001	-	không	không	-
0110	Cao nhất	Lỗi trên đường truyền nhận Rx	OE hoặc PE hoặc FE hoặc BI	U0LSR được đọc
0100	Mức 2	Dữ liệu Rx đã sẵn sàng	Dữ liệu Rx đã sẵn sàng hoặc (U0FCR0=1)	U0RBR được đọc hoặc UART0 FIFO xuống dưới mức kích
1100	Mức 2	Báo hiệu thời hạn truyền nhận bị Time-out	Thời gian Time-out = [(word length)x7 - 2]x8 + {(mức kích - số ký tự)x8 + 1} RCLKs	U0 RBR được đọc
0010	Mức 3	THRE	THRE	U0IIR được đọc (nếu là nguồn ngắt) hoặc ghi vào THR

Các giá trị U0IIR[3:0] = {"0000", "0011", "0101", "0111", "1000", "1001", "1010", "1011", "1101", "1110", "1111"} đều không hợp lệ

**Bảng 8: Thường trình ngắt UART0**

Ngắt UART0 THRE(U0IIR3:1=001) có mức ưu tiên thứ 3 và được tích cực khi UART0 THR FIFO được làm rỗng.

*C.V. Thanh ghi điều khiển FIFO cho truyền thông nối tiếp <U0FCR - 0xE000C008>:*  
 Thanh ghi này điều khiển hoạt động của bộ FIFO Rx và Tx.

U0FCR	Chức năng	Giải thích	Giá trị Reset
0	Cho phép FIFO	Tích cực mức cao sẽ cho phép truy cập UART0 Rx, Tx FIFOs và U0FCR[[7:1]. Bit này phải được cho bằng 1 để UART0 hoạt động đúng. Sự chuyển đổi trên bit này sẽ tự động xóa bộ UART0 FIFOs.	0
1	Reset FIFO Rx	Ghi mức 1 vào bit U0FCR[1] sẽ xóa tất cả các byte hiện đang lưu trữ trên UART0 Rx FIFO và Reset con trỏ về vị trí hợp lý. Bit này có thể tự xóa.	0
2	Reset FIFO Tx	Ghi mức 1 vào bit U0FCR[2] sẽ xóa tất cả các byte hiện đang lưu trữ trên UART0 Tx FIFO và Reset con trỏ về vị trí hợp lý. Bit này có thể tự xóa.	0
5:3	Dự trữ	Dự trữ, không được sử dụng	NA
7:6	Chọn mức ngắt Rx	00: ngắt ở mức 0 (mặc định= truyền nhận 1 ký tự) 01: ngắt ở mức 1 (mặc định= truyền nhận 4 ký tự) 10: ngắt ở mức 2 (mặc định= truyền nhận 8 ký tự) 11: ngắt ở mức 3 (mặc định= truyền nhận 14 ký tự) Tổ hợp 2 bit này cho biết có bao nhiêu ký tự được nhận từ UART FIFO sẽ gây ngắt. 4 mức kích này có thể được định nghĩa bởi người lập trình tùy vào số ký tự ở bộ FIFO.	0

**Bảng 9: Ý nghĩa các bit của thanh ghi điều khiển UART0 FIFO**

*C.VI. Thanh ghi điều khiển đường truyền cho truyền thông nối tiếp <U0LCR - 0xE000C00C>:*

Thanh ghi này cho biết dạng của ký tự dữ liệu truyền (phát hoặc nhận).

U0LCR	Chức năng	Giải thích	Giá trị Reset
1:0	Lựa chọn chiều dài từ	00: Kí tự dài 5 bit 01: Kí tự dài 6 bit 10: Kí tự dài 7 bit 11: Kí tự dài 8 bit	0
2	Chọn số Bit dừng	0: 1 stop bit 1: 2 stop bits (1.5 nếu U0LCR[1:0]=00)	0
3	Parity Enable	0: không cho phép tạo và kiểm tra parity 1: cho phép tạo và kiểm tra parity	0
5:4	Chọn Parity	00: parity lẻ 01: parity chẵn 10: gán "1" là parity 11: gán "0" là parity	0
6 l	Break Control	0: Không cho phép dừng truyền dữ liệu 1: Cho phép dừng truyền dữ liệu Ngõ ra của chân UART0 TxD sẽ bị gán = 0 khi U0LCR6 tích cực mức cao.	0
7	Divisor Latch Access Bit	0: Không cho phép truy cập Divisor Latches 1: Cho phép truy cập Divisor Latches	0

**Bảng 10: Ý nghĩa các bit của thanh ghi điều khiển đường truyền UART0(U0LCR - 0xE000C00C)**  
**C.VII. Thanh ghi điều khiển trạng thái cho đường truyền cho truyền thông nối tiếp <U0LSR - 0xE000C014>:**

Thanh ghi này cho biết trạng thái thông tin trên các khối UART0 Tx và Rx, thuộc tính chỉ đọc.

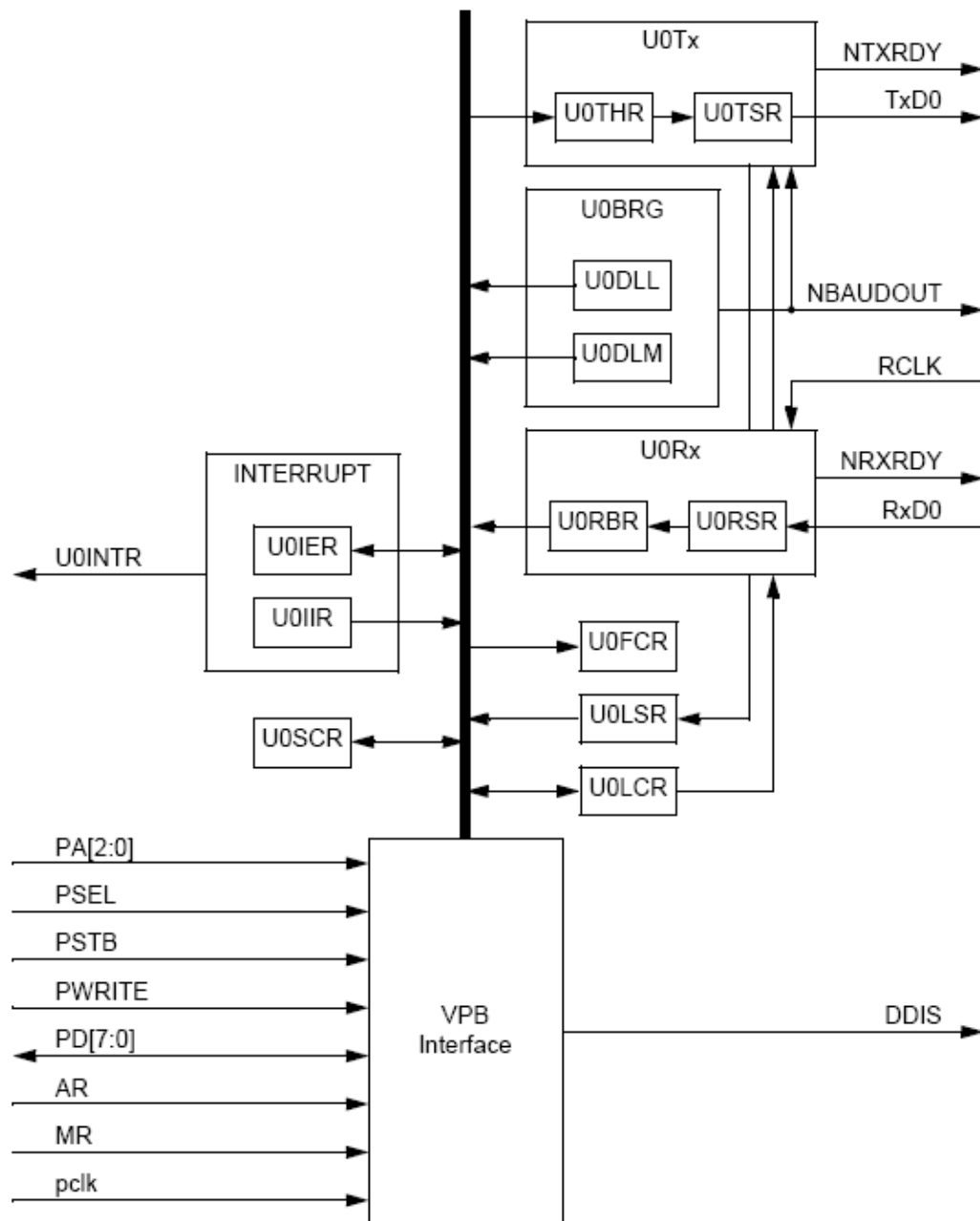
U0LSR	Chức năng	Giải thích	Giá trị Reset
0	Sẵn sàng nhận dữ liệu (RDR)	0: U0RBR rỗng. 1: U0RBR chứa các giá trị hợp lệ U0LSR0='1' khi U0RBR giữ các kí tự không đọc được và được xóa khi UART0 RBR FIFO được làm rỗng.	0
1	Lỗi tràn (OE)	0: Không bị tràn. 1: bị tràn Điều kiện lỗi tràn ngay lập tức sẽ đưa bit này lên tích cực, U0LSR được đọc sẽ xóa U0LSR[1]. U0LSR[1]='1' khi UART0 RSR nhận được kí tự mới trong khi bộ UART0 RBR FIFO đã đầy. Trong trường hợp này, UART0 RBR FIFO sẽ không ghi đè lên và giá trị muốn vào UART0 RSR sẽ bị mất.	0
2	Kiểm tra lỗi Parity(PE)	0: kiểm tra Parity không lỗi 1: kiểm tra Parity bị lỗi Khi bit parity từ kí tự nhận được bị lỗi, sẽ xuất hiện lỗi parity,U0LSR đọc và xóa U0LSR2.	0
3	Khung lỗi (FE)	0: Không có lỗi khung 1: Lỗi khung. Khi bit dừng của kí tự truyền nhận='0'. Lỗi khung xuất hiện.U0LSR đọc sẽ xóa bit này.	0
4	Ngắt dừng truyền thông nối tiếp(BI)	0: Trạng thái dừng ngắt truyền thông không tích cực 1: Trạng thái dừng ngắt truyền thông tích cực Khi RxD0 được giữ trong thời gian bằng 1 lần truyền kí tự (start, data, parity, stop), trạng thái ngắt truyền thông nối tiếp lên mức tích cực. Khi bit này lên mức tích cực, bộ nhận sẽ ở trạng thái rỗi (Idle). Bộ nhận sẽ bắt đầu hoạt động lại khi nào tất cả các bit(start, data, parity, stop) lên 1. U0LSR đọc sẽ xóa bit này.	0
5	Thanh ghi giữ giá trị truyền phát đã rỗng (THRE)	0: U0THR chứa dữ liệu hợp lệ 1: U0THR rỗng. THRE='1' ngay sau khi phát hiện UART0 THR rỗng và được xóa khi ghi dữ liệu vào U0THR	1
6	Quá trình truyền phát là rỗng (TEMT)	0: U0THR và/hoặc U0TSR chứa các dữ liệu hợp lệ 1: U0THR và U0TSR rỗng. TEMT='1' khi cả U0THR và U0TSR đều rỗng. TEMT được xóa khi U0TSR hoặc U0THR chứa các dữ liệu hợp lệ.	1

7	Lỗi do Rx FIFO (RxFE)	0: U0RBR không có lỗi UART0 Rx hoặc U0FCR0=0. 1: UART0 RBR có ít nhất 1 lỗi UART0 Rx . U0LSR[7]='1' khi kí tự với lỗi truyền nhận như lỗi khung, lỗi parity, hay ngắt dừng truyền thông được load vào U0RBR. Bit này sẽ được xóa khi thanh ghi U0LSR đọc và không có lỗi tương tự xuất hiện ở UART0 FIFO.	0
---	-----------------------	---	---

**Bảng 11: Ý nghĩa các bit của thanh ghi trạng thái đường truyền truyền thông nối tiếp UART0**

*D. Kiến trúc của bộ truyền thông nối tiếp UART0:*

*D.I. Hình vẽ:*



**Hình 1: Sơ đồ khối bộ truyền thông nối tiếp bất đồng bộ UART0**

D.II. Chú thích:

- ✚ Giao tiếp VPB cung cấp đường truyền thông giữa CPU hay host với UART0.
- ✚ Khối thu UART0: U0Rx, quản lý đường vào nối tiếp, RxD0, cho tín hiệu vào hợp lệ. thanh ghi dịch UART0 Rx (U0RSR) nhận các dữ liệu hợp lệ thông qua RxD0. Sau khi các kí tự hợp lệ vào U0RSR, nó sẽ qua thanh ghi đệm UART0 Rx FIFO và chờ CPU hay máy chủ truy cập thông qua giao tiếp chung.
- ✚ Khối nhận UART0: U0Tx, cho phép dữ liệu từ CPU hay host được giữ ở bộ đệm dữ liệu UART0 Tx FIFO (U0THR). Thanh ghi dịch UART0 Tx(U0TSR) đọc dữ liệu chứa trong U0THR và tập hợp chúng để gửi qua chân TxD.
- ✚ Khối sinh tốc độ Baud UART0: U0BRG, tạo thời gian cho phép sử dụng khối UART0 Tx. Xung nhịp của U0BRG bắt nguồn từ xung nhịp của VPB(PCLK),  $NBAUDOUT = PCCK/16x$  với  $x=[U0DLM-U0DLL]$ .
- ✚ Giao diện ngắt bao gồm 2 thanh ghi U0IER và U0IIR.

E. Chương trình minh họa: “Hello World!”E.I. Mã nguồn:I.1. Khai báo:

Để giao tiếp MCU-PC, ta cần khai báo hoạt động UART <giả sử sử dụng UART0>:

+Các chân sử dụng.

+Tốc độ Baud sử dụng.

+Kiểu truyền dữ liệu, các thông số của truyền thông nối tiếp bất đồng bộ.

+Cho DLAB=1.

Ta thực hiện quá trình đó, như sau:

+Các chân sử dụng: TxD0 và RxD0

*PINSEL0=0x0000 0001;* <bảng 58, trang 110 hướng dẫn sử dụng>

+Tốc độ Baud:  $NBAUDOUT = PCCK/16/[U0DLM-U0DLL]$

Giả sử dùng thạch anh 11.0592MHz,  $PCCK=5CCLK$ ,  $PCLK=CCLK$ , chọn tốc độ

$Baud=115200 \rightarrow [U0DLM-U0DLL]=0x1E$

*U0DLM=0x00;*

*U0DLL=0x1E;*

+Kiểu truyền dữ liệu, các thông số thiết lập: dữ liệu 8 bits, không parity, 1 bit dừng, xem bảng 8, ta có:

$U0LCR[1:0] = 11;$

$U0LCR[2] = 0;$

$U0LCR[3] = 0;$

Tức là:

*U0LCR=0x03;*

Đoạn code khởi tạo truyền thông nối tiếp như sau:

```
uart_init()
{
    PINSEL0=0x0000 0001;
    //Tích cực DLAB trước khi ghi giá trị vào các thanh ghi U0DLM và U0DLL
    U0LCR = 0x83;
    U0DLM=0x00;
    U0DLL=0x1E;
    // DLAB = 0 cho máy khởi tạo tốc độ baud hoạt động
    U0LCR = 0x03;
} //end uart_init()
```



*1.2. Chương trình chính:*

Ta dùng thư viện `stdio.h` sẵn có của Keil-uV3 viết chương trình Hello World như sau:

```
//Khai bao cac header su dung
#include <stdio.h>
#include <lpc22xx>
//vao chuong trinh chinh
void main (void)
{
    //Luu y phai thiet dat cac thong so MSEL, PSEL, PLLDIV truoac khi dung
    uart_init();
    printf("  Khan thuong nho ai?\n  Khan roi xuong dat\n  Khan thuong
nho ai?\n  Khan vat len vai\n  Khan thuong nho ai?\n  Khan chui nuoc mat\n  Den
thuong nho ai?\n  Ma den khong tat\n  Mat thuong nho ai?\n  Mat ngu khong yen!");
    while(1);    //Vong lap vo tan
}
```

Thực tế, các hàm khai báo trong thư viện `stdio.h` chỉ nhận dữ liệu vào bộ đệm hoặc đẩy dữ liệu ra bộ đệm mà không nhận về CPU hoặc gửi ra target, ta sử dụng thêm hàm khai báo `getchar(int ch)` và `putchar(int ch)` sẵn có trong ví dụ Hello của ARM để làm việc đây:

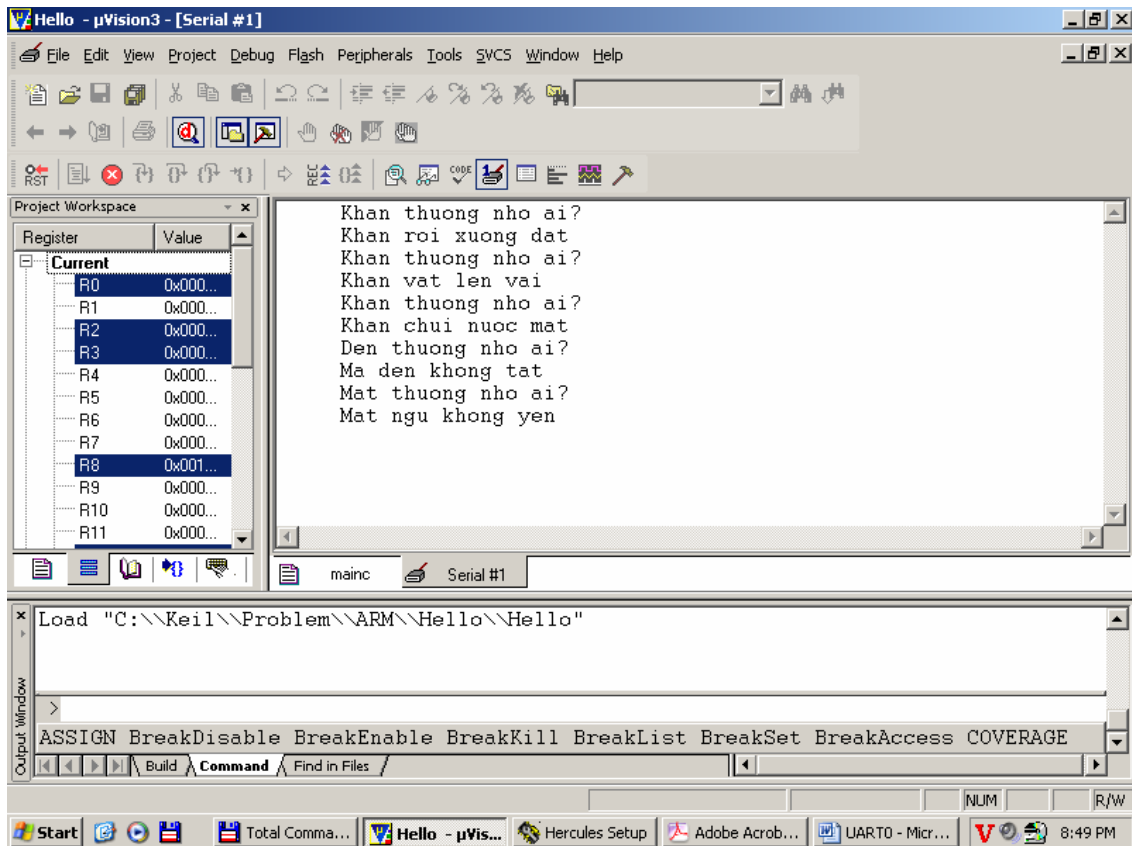
*Hàm Serial.c:*

```
/******
/* This file is part of the uVision/ARM development tools */
/* Copyright KEIL ELEKTRONIK GmbH 2002-2004 */
/******
/* */
/* SERIAL.C: Low Level Serial Routines */
/* */
/******
#include <LPC22xx.H>          /* Dinh nghia LPC22xx */
#define CR    0x0D

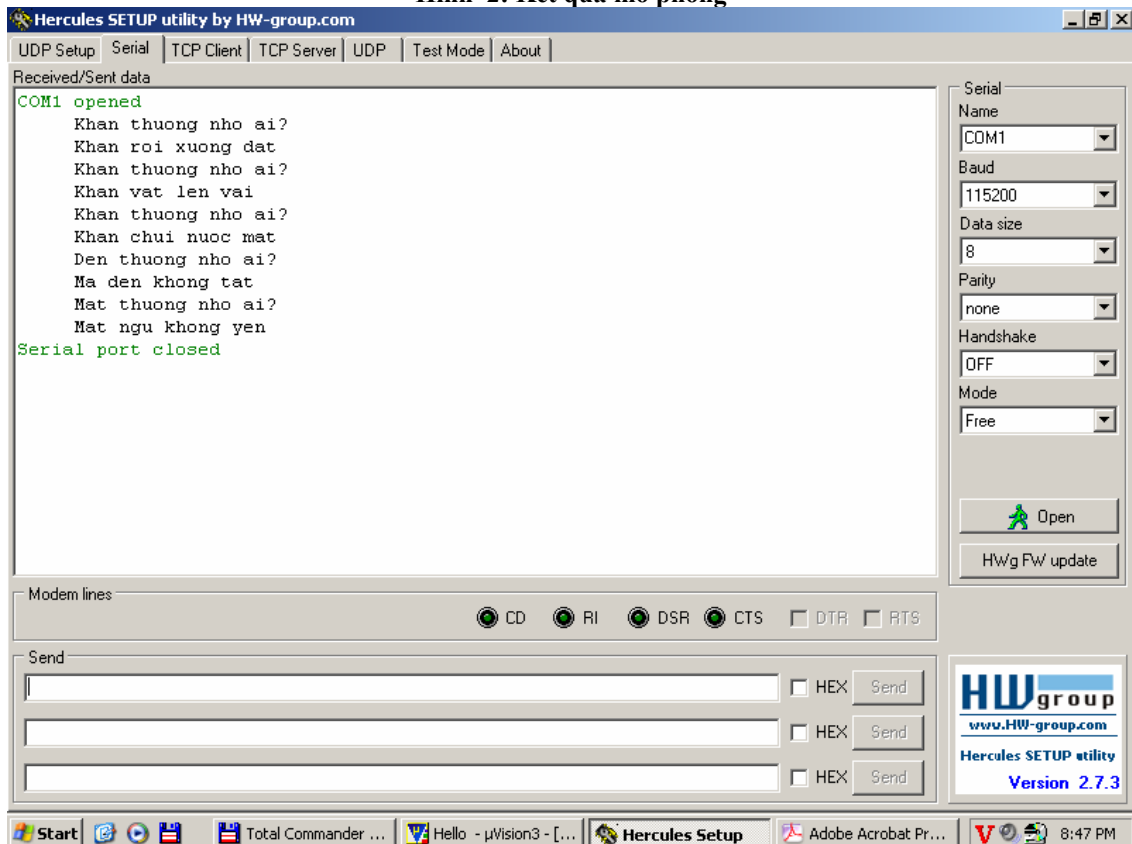
int putchar (int ch) {          /* Ghi ki tu ra cong noi tiep Serial Port */
//Neu ki tu la dau xuong hang, tra ve CR
    if (ch == '\n') {
        while (!(U0LSR & 0x20));
        U0THR = CR;          /* output CR */
    }
//Neu khong, nhan gia tri day vao thanh ghi U0THR
    while (!(U0LSR & 0x20));
    return (U0THR = ch);
}

int getchar (void) {          /* Doc ki tu tu cong noi tiep */
//Neu co du lieu hop le trong bo dem, ghi vao thanh ghi U0RBR
    while (!(U0LSR & 0x01));
    return (U0RBR);
}
```

*E.II. Kết quả mô phỏng:*



Hình 2: Kết quả mô phỏng



Hình 3: Kết quả thực tế

F. Kết luận:

Bài viết này tôi đã giới thiệu rất chi tiết cách dùng cổng truyền thông nối tiếp cho ARM7 LPC2214 của Philips Semiconductors. Các hình vẽ và nội dung được dịch từ tài liệu hướng dẫn sử dụng của LPC2214. Ví dụ minh họa có sử dụng chương trình viết sẵn của Keil-uV3. Chương trình giao tiếp qua cổng COM1 có thể dùng bất kì, tuy nhiên, phải lưu tần số các thiết đặt thuộc tính, với mạch tôi đang sử dụng, DTR=RTS=0.

Các kết quả đã được kiểm tra đúng như lý luận

Mọi liên hệ và thắc mắc khác <nếu có> xin trao đổi qua:

Bùi Trung Hiếu

Webmaster : <http://www.khvt.com>

Email : [buitrunghieu@khvt.com](mailto:buitrunghieu@khvt.com)

Cell : (+84)98.3210.906

Cập nhật lần cuối vào lúc: 2/25/2006 11:05:23 AM