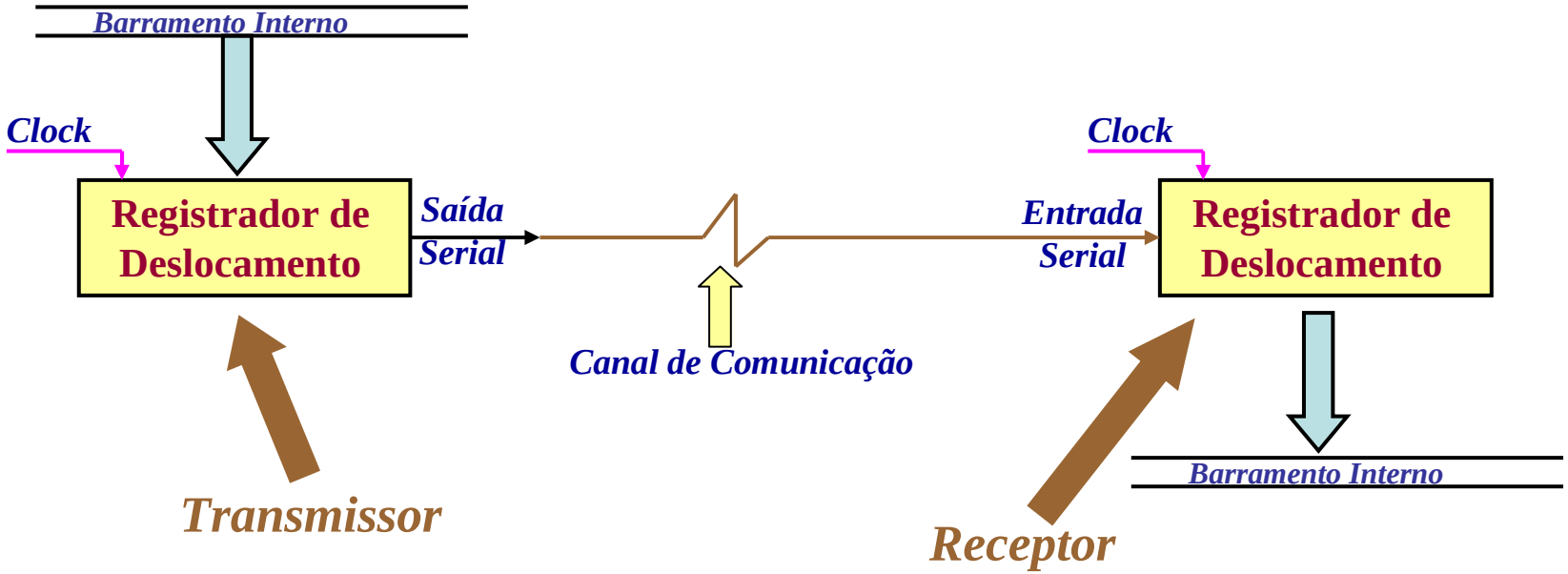
Configurar a Interface Serial (UART):

***Freq.* DCO = 8 MHZ**
Taxa = 9.600 bps
Nº de bits = 8 (sem paridade)

Após a recepção do primeiro byte, enviar um texto inicial, e retornar (ecoar) os bytes recebidos pela UART:

```
>Interface Serial:
>1
>2
>3
>A
>B
>C
>D
>
```

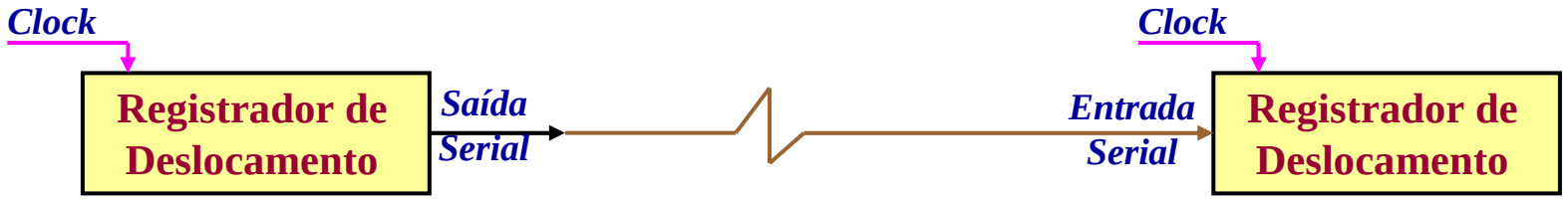
Interface Serial (UART)



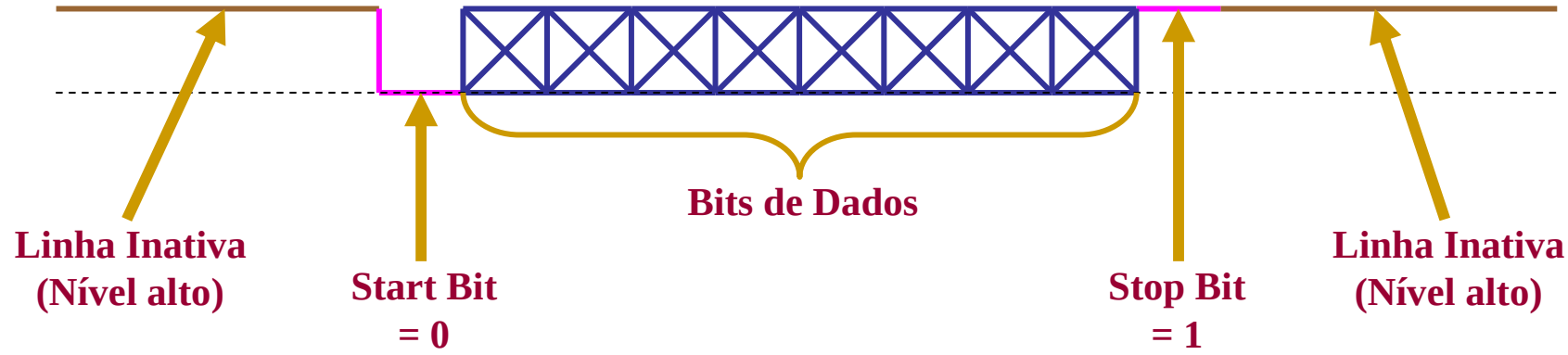
***Taxa de Transmissão*:** **Freqüência de *clock* dos registradores de deslocamento**

***Taxa de Transmissão*: bps (*baud*)**

Interface Serial (UART)



Transmissão de um byte:



USCI_Ax:

UART Mode (UCSYNC = 0)

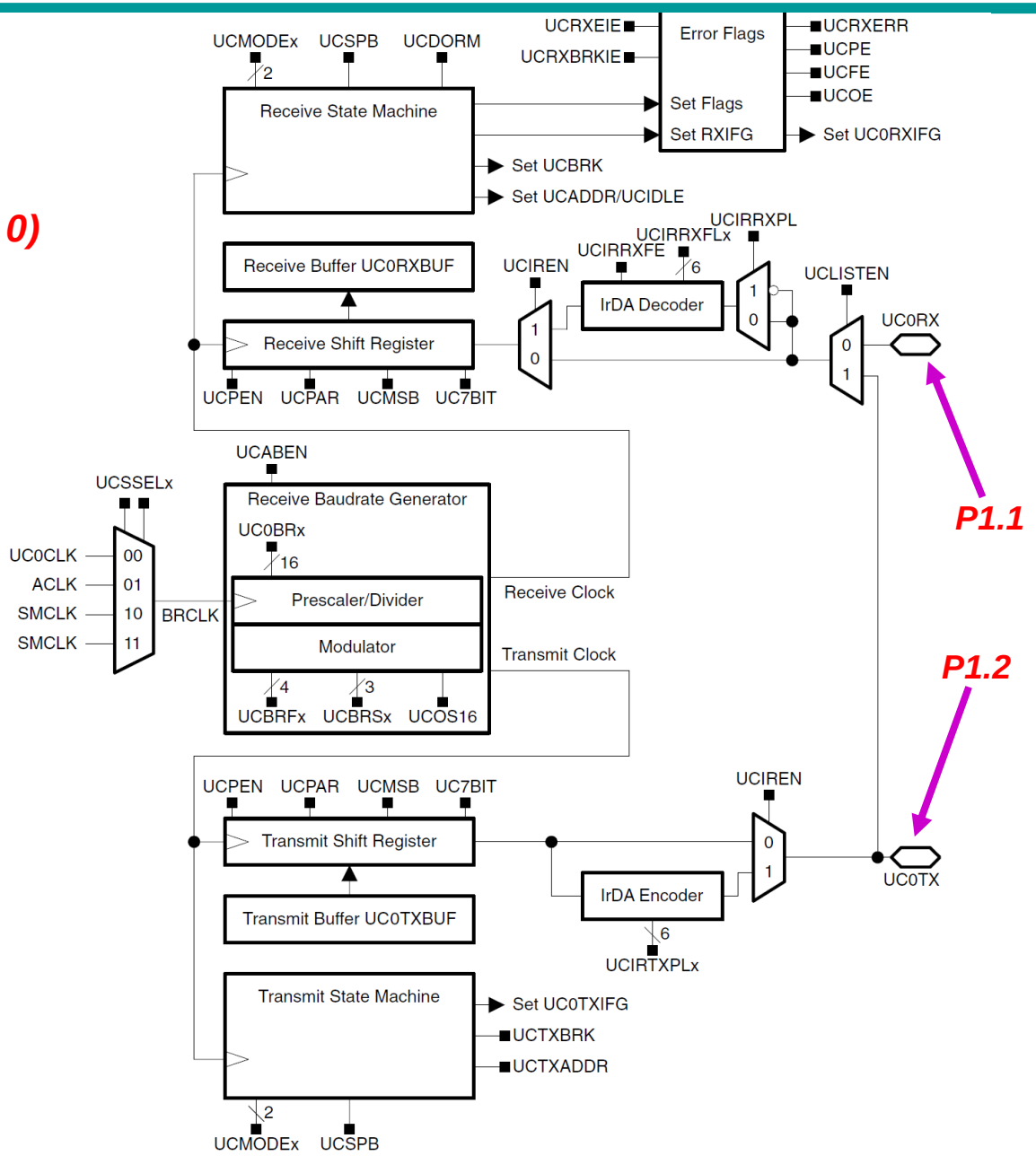


Table 16. Port P1 (P1.0 to P1.2) Pin Functions

PIN NAME (P1.x)	x	FUNCTION	CONTROL BITS / SIGNALS (1)				
			P1DIR.x	P1SEL.x	P1SEL2.x	ADC10AE.x INCH.x=1 (2)	CAPD.y
P1.0/	0	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0CLK/		TA0.TACLK	0	1	0	0	0
ACLK/		ACLK	1	1	0	0	0
A0 (2)/		A0	X	X	X	1 (y = 0)	0
CA0/		CA0	X	X	X	0	1 (y = 0)
Pin Osc		Capacitive sensing	X	0	1	0	0
P1.1/	1	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0.0/		TA0.0	1	1	0	0	0
		TA0.CCI0A	0	1	0	0	0
UCA0RXD/		UCA0RXD	from USCI	1	1	0	0
UCA0SOMI/		UCA0SOMI	from USCI	1	1	0	0
A1 (2)/		A1	X	X	X	1 (y = 1)	0
CA1/		CA1	X	X	X	0	1 (y = 1)
Pin Osc		Capacitive sensing	X	0	1	0	0
P1.2/	2	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0.1/		TA0.1	1	1	0	0	0
		TA0.CCI1A	0	1	0	0	0
UCA0TXD/		UCA0TXD	from USCI	1	1	0	0
UCA0SIMO/		UCA0SIMO	from USCI	1	1	0	0
A2 (2)/		A2	X	X	X	1 (y = 2)	0
CA2/		CA2	X	X	X	0	1 (y = 2)
Pin Osc		Capacitive sensing	X	0	1	0	0

UCAxCTL0, USCI_Ax Control Register 0

7	6	5	4	3	2	1	0
UCPEN	UCPAR	UCMSB	UC7BIT	UCSPB	UCMODEx		UCSYNC=0
rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–0

UCPEN Bit 7 Parity enable
- 0 Parity disabled.
- 1 Parity enabled. Parity bit is generated (UCAxTXD) and expected (UCAxRXD). In address-bit multiprocessor mode, the address bit is included in the parity calculation.

UCPAR Bit 6 Parity select. UCPAR is not used when parity is disabled.
- 0 Odd parity
- 1 Even parity

UCMSB Bit 5 MSB first select. Controls the direction of the receive and transmit shift register.
- 0 LSB first
- 1 MSB first

UC7BIT Bit 4 Character length. Selects 7-bit or 8-bit character length.
- 0 8-bit data
- 1 7-bit data

UCSPB Bit 3 Stop bit select. Number of stop bits.
- 0 One stop bit
- 1 Two stop bits

UCMODEx Bits 2–1 USCI mode. The UCMODEx bits select the asynchronous mode when UCSYNC = 0.
- 00 UART Mode.
- 01 Idle-Line Multiprocessor Mode.
- 10 Address-Bit Multiprocessor Mode.
- 11 UART Mode with automatic baud rate detection.

UCSYNC Bit 0 Synchronous mode enable
- 0 Asynchronous mode
- 1 Synchronous Mode

UCAxCTL1, USCI_Ax Control Register 1

7	6	5	4	3	2	1	0
UCSSELx		UCRXEIE	UCBRKIE	UCDORM	UCTXADDR	UCTXBRK	UCSWRST
rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–1

UCSSELx Bits 7-6 — USCI clock source select. These bits select the BRCLK source clock.
- 00 UCLK
- 01 ACLK
- 10 SMCLK
- 11 SMCLK

UCRXEIE Bit 5 — Receive erroneous-character interrupt-enable
- 0 Erroneous characters rejected and UCAxRXIFG is not set
- 1 Erroneous characters received will set UCAxRXIFG

UCBRKIE Bit 4 — Receive break character interrupt-enable
- 0 Received break characters do not set UCAxRXIFG.
- 1 Received break characters set UCAxRXIFG.

UCDORM Bit 3 — Dormant. Puts USCI into sleep mode.
- 0 Not dormant. All received characters will set UCAxRXIFG.
- 1 Dormant. Only characters that are preceded by an idle-line or with address bit set will set UCAxRXIFG. In UART mode with automatic baud rate detection only the combination of a break and synch field will set UCAxRXIFG.

UCTXADDR Bit 2 — Transmit address. Next frame to be transmitted will be marked as address depending on the selected multiprocessor mode.
- 0 Next frame transmitted is data
- 1 Next frame transmitted is an address

UCTXBRK Bit 1 — Transmit break. Transmits a break with the next write to the transmit buffer. In UART mode with automatic baud rate detection 055h must be written into UCAxTXBUF to generate the required break/synch fields. Otherwise 0h must be written into the transmit buffer.
- 0 Next frame transmitted is not a break
- 1 Next frame transmitted is a break or a break/synch

UCSWRST Bit 0 — Software reset enable
- 0 Disabled. USCI reset released for operation.
- 1 Enabled. USCI logic held in reset state.

UCAxBR0, USCI_Ax Baud Rate Control Register 0

7	6	5	4	3	2	1	0
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw

UCAxBR1, USCI_Ax Baud Rate Control Register 1

7	6	5	4	3	2	1	0
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw

UCBRx	Clock prescaler setting of the Baud rate generator. The 16-bit value of (UCAxBR0 + UCAxBR1 × 256) forms the prescaler value.

UCAxMCTL, USCI_Ax Modulation Control Register

7	6	5	4	3	2	1	0
UCBRFx				UCBRSx			UCOS16
rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–0	rw–0

UCBRFx	Bits 7–4	First modulation stage select. These bits determine the modulation pattern for BITCLK16 when UCOS16 = 1. Ignored with UCOS16 = 0. Table 15–3 shows the modulation pattern.
UCBRSx	Bits 3–1	Second modulation stage select. These bits determine the modulation pattern for BITCLK. Table 15–2 shows the modulation pattern.
UCOS16	Bit 0	Oversampling mode enabled 0 Disabled 1 Enabled

15.3.10 Setting a Baud Rate

For a given BRCLK clock source, the baud rate used determines the required division factor N:

$$N = \frac{f_{BRCLK}}{Baudrate}$$

The division factor N is often a non-integer value thus at least one divider and one modulator stage is used to meet the factor as closely as possible.

If N is equal or greater than 16 the oversampling baud rate generation mode can be chosen by setting UCOS16.

f_{BRCLK} = 8 MHz

$Baudrate$ = 9.600 bps

N = 8000000 / 9600 = 833,33333 . . .

15.3.10 Setting a Baud Rate

For a given BRCLK clock source, the baud rate used determines the required division factor N:

$$N = \frac{f_{BRCLK}}{Baudrate}$$

The division factor N is often a non-integer value thus at least one divider and one modulator stage is used to meet the factor as closely as possible.

If N is equal or greater than 16 the oversampling baud rate generation mode can be chosen by setting UCOS16.

Low-Frequency Baud Rate Mode Setting

In the low-frequency mode, the integer portion of the divisor is realized by the prescaler:

UCBRx = INT(N)

and the fractional portion is realized by the modulator with the following nominal formula:

UCBRSx = round((N – INT(N)) * 8)

IE2, Interrupt Enable Register 2

7	6	5	4	3	2	1	0
				UCB0TXIE	UCB0RXIE	UCA0TXIE	UCA0RXIE
				rw-0	rw-0	rw-0	rw-0

	Bits 7-4	These bits may be used by other modules (see the device-specific data sheet).
UCB0TXIE	Bit 3	USCI_B0 transmit interrupt enable 0 Interrupt disabled 1 Interrupt enabled
UCB0RXIE	Bit 2	USCI_B0 receive interrupt enable 0 Interrupt disabled 1 Interrupt enabled
UCA0TXIE	Bit 1	USCI_A0 transmit interrupt enable 0 Interrupt disabled 1 Interrupt enabled
UCA0RXIE	Bit 0	USCI_A0 receive interrupt enable 0 Interrupt disabled 1 Interrupt enabled

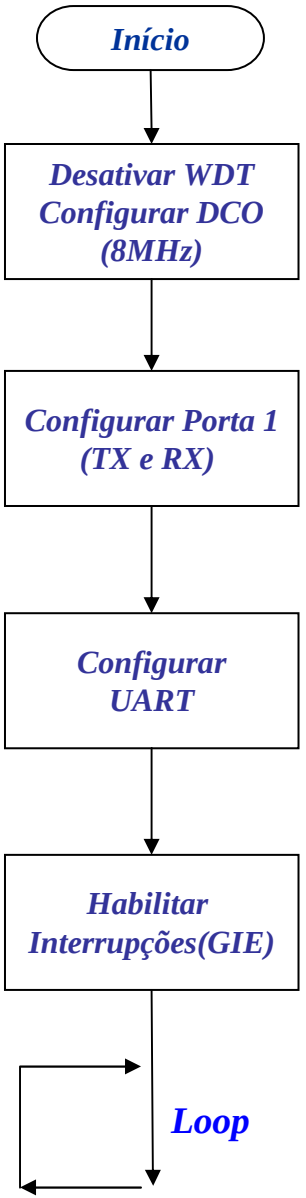
IFG2, Interrupt Flag Register 2

7	6	5	4	3	2	1	0
				UCB0 TXIFG	UCB0 RXIFG	UCA0 TXIFG	UCA0 RXIFG
				rw-1	rw-0	rw-1	rw-0

	Bits 7-4	These bits may be used by other modules (see the device-specific data sheet).
UCB0 TXIFG	Bit 3	USCI_B0 transmit interrupt flag. UCB0TXIFG is set when UCB0TXBUF is empty. 0 No interrupt pending 1 Interrupt pending
UCB0 RXIFG	Bit 2	USCI_B0 receive interrupt flag. UCB0RXIFG is set when UCB0RXBUF has received a complete character. 0 No interrupt pending 1 Interrupt pending
UCA0 TXIFG	Bit 1	USCI_A0 transmit interrupt flag. UCA0TXIFG is set when UCA0TXBUF empty. 0 No interrupt pending 1 Interrupt pending
UCA0 RXIFG	Bit 0	USCI_A0 receive interrupt flag. UCA0RXIFG is set when UCA0RXBUF has received a complete character. 0 No interrupt pending 1 Interrupt pending

Interrupções

```
#define PORT1_VECTOR          (2 * 2u)   /* 0xFFE4 Port 1 */
#define PORT2_VECTOR          (3 * 2u)   /* 0xFFE6 Port 2 */
#define ADC10_VECTOR          (5 * 2u)   /* 0xFFEA ADC10 */
#define USCIAB0TX_VECTOR      (6 * 2u)   /* 0xFFEC USCI A0/B0 Transmit */
#define USCIAB0RX_VECTOR      (7 * 2u)   /* 0xFFEE USCI A0/B0 Receive */
#define TIMER0_A1_VECTOR      (8 * 2u)   /* 0xFFF0 Timer0)A CC1, TA0 */
#define TIMER0_A0_VECTOR      (9 * 2u)   /* 0xFFF2 Timer0_A CC0 */
#define WDT_VECTOR            (10 * 2u)   /* 0xFFF4 Watchdog Timer */
#define COMPARATORA_VECTOR    (11 * 2u)   /* 0xFFF6 Comparator A */
#define TIMER1_A1_VECTOR      (12 * 2u)   /* 0xFFF8 Timer1_A CC1-4, TA1 */
#define TIMER1_A0_VECTOR      (13 * 2u)   /* 0xFFFA Timer1_A CC0 */
#define NMI_VECTOR            (14 * 2u)   /* 0xFFFC Non-maskable */
#define RESET_VECTOR          (15 * 2u)   /* 0xFFFE Reset [Highest Priority] */
```



P1SEL1

P1SEL2

UCA0CTL0: → *UART Mode, Assíncrono, 8 bits, sem paridade, LSB first, 1 stop bit*

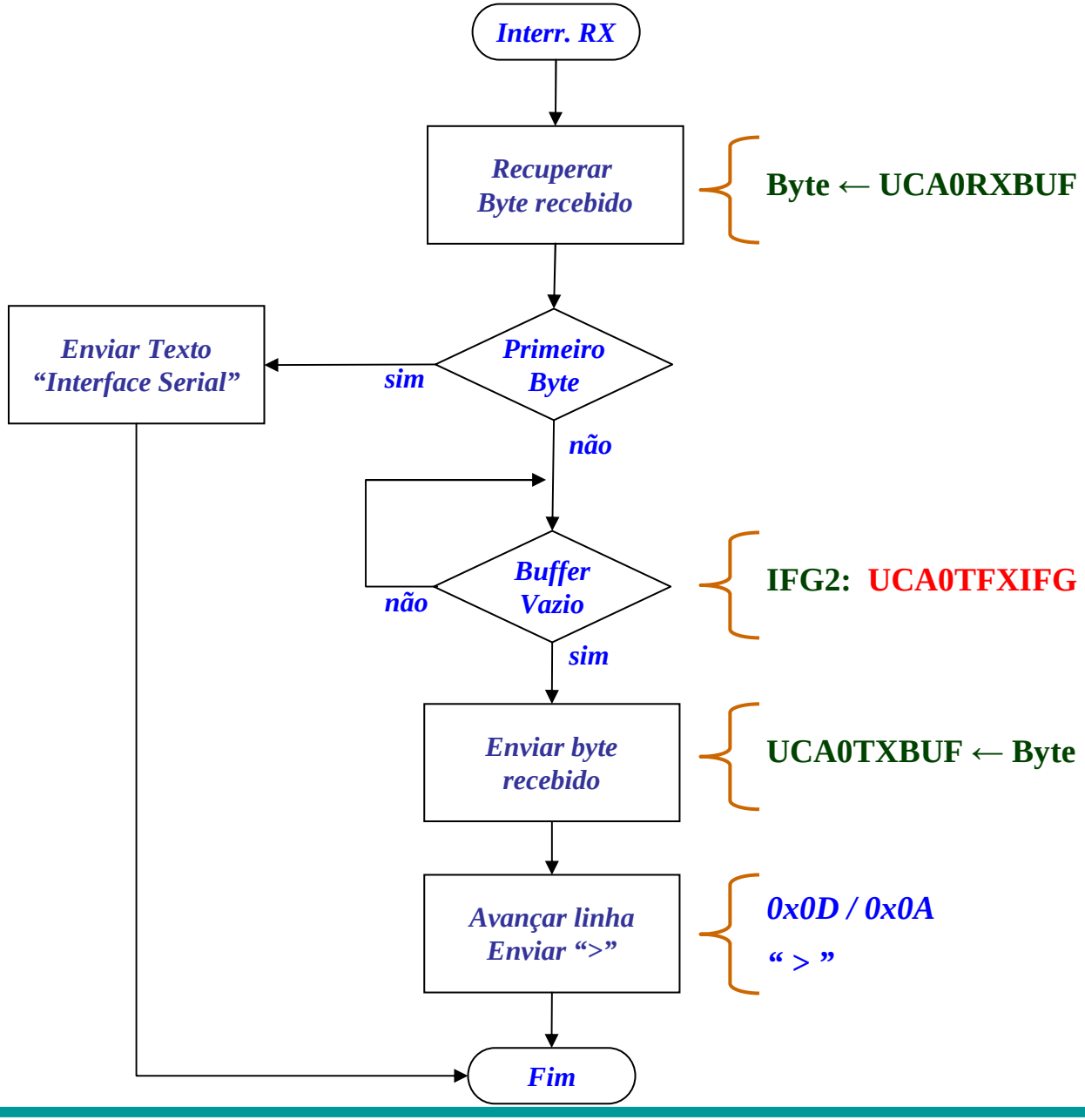
UCA0CTL1: UCSSEL*x* → *SMCLK*

UCA0CTL1: UCSWRST → *USCI liberada para operação*

UCAMCTL: → *UCBRFx = 0, UCBRSx = xx, UCOS16 = 0*

UCABR1/UCABR0: → *9600 bps*

IE2: UCA0RXIE → *Habilita interrupção de recepção*


Interr. RX
Recuperar Byte recebido
Byte ← UCA0RXBUF
Primeiro Byte
sim
Enviar Texto "Interface Serial"
não
Buffer Vazio
não
IFG2: UCA0TFXIFG
sim
Enviar byte recebido
UCA0TXBUF ← Byte
Avançar linha Enviar ">"
0x0D / 0x0A
" > "
Fim