# CORTEX NXP1769: HOJAS DE DATOS PARA LA REALIZACIÓN DE EXÁMENES DE INFO II

# Acceso a Hardware

| PINSEL0 to PINSEL9 Values | Function |
|---|---|
| 00 | Primary (default) function, typically GPIO port |
| 01 | First alternate function |
| 10 | Second alternate function |
| 11 | Third alternate function |

| PINMODE_OD0 to PINMODE_OD4 Values | Function |
|---|---|
| 0 | Pin is in the normal (not open drain) mode. |
| 1 | Pin is in the open drain mode. |

| PINMODE0 to PINMODE9 Values | Function |
|---|---|
| 00 | Pin has an on-chip pull-up resistor enabled. |
| 01 | Repeater mode |
| 10 | Pin has neither pull-up nor pull-down resistor enabled. |
| 11 | Pin has an on-chip pull-down resistor enabled. |

| Register | Controls | |
|---|---|---|
| PINSEL0 | P0[15:0] | |
| PINSEL1 | P0 [31:16] | |
| PINSEL2 | P1 [15:0] (Ethernet) | |
| PINSEL3 | P1 [31:16] | |
| PINSEL4 | P2 [15:0] | |
| PINSEL5 | P2 [31:16] | not used |
| PINSEL6 | P3 [15:0] | not used |
| PINSEL7 | P3 [31:16] | |
| PINSEL8 | P4 [15:0] | not used |
| PINSEL9 | P4 [31:16] | |
| PINSEL10 | Trace port enable | |

- `void SetPINSEL(uint8_t puerto, uint8_t pin, uint8_t modo)`
- `void SetPINMODE(uint8_t port, uint8_t pin ,uint8_t modo)`
- `void SetDIR(uint32_t* puerto,uint8_t pin,uint8_t direccion)`
- `void SetPIN(uint32_t* puerto,uint8_t pin,uint8_t estado)`
- `uint8_t GetPIN(uint8_t puerto , uint8_t pin , uint8_t actividad )`

| Name | Description | Access | Reset Value[1] | Address |
|---|---|---|---|---|
| PINSEL0 | Pin function select register 0. | R/W | 0 | 0x4002 C000 |
| PINSEL1 | Pin function select register 1. | R/W | 0 | 0x4002 C004 |
| PINSEL2 | Pin function select register 2. | R/W | 0 | 0x4002 C008 |
| PINSEL3 | Pin function select register 3. | R/W | 0 | 0x4002 C00C |
| PINSEL4 | Pin function select register 4 | R/W | 0 | 0x4002 C010 |
| PINSEL7 | Pin function select register 7 | R/W | 0 | 0x4002 C01C |
| PINSEL8 | Pin function select register 8 | R/W | 0 | 0x4002 C020 |
| PINSEL9 | Pin function select register 9 | R/W | 0 | 0x4002 C024 |
| PINSEL10 | Pin function select register 10 | R/W | 0 | 0x4002 C028 |
| PINMODE0 | Pin mode select register 0 | R/W | 0 | 0x4002 C040 |
| PINMODE1 | Pin mode select register 1 | R/W | 0 | 0x4002 C044 |
| PINMODE2 | Pin mode select register 2 | R/W | 0 | 0x4002 C048 |
| PINMODE3 | Pin mode select register 3. | R/W | 0 | 0x4002 C04C |
| PINMODE4 | Pin mode select register 4 | R/W | 0 | 0x4002 C050 |
| PINMODE5 | Pin mode select register 5 | R/W | 0 | 0x4002 C054 |
| PINMODE6 | Pin mode select register 6 | R/W | 0 | 0x4002 C058 |
| PINMODE7 | Pin mode select register 7 | R/W | 0 | 0x4002 C05C |
| PINMODE9 | Pin mode select register 9 | R/W | 0 | 0x4002 C064 |
| PINMODE_OD0 | Open drain mode control register 0 | R/W | 0 | 0x4002 C068 |
| PINMODE_OD1 | Open drain mode control register 1 | R/W | 0 | 0x4002 C06C |
| PINMODE_OD2 | Open drain mode control register 2 | R/W | 0 | 0x4002 C070 |
| PINMODE_OD3 | Open drain mode control register 3 | R/W | 0 | 0x4002 C074 |
| PINMODE_OD4 | Open drain mode control register 4 | R/W | 0 | 0x4002 C078 |
| I2CPADCFG | $I^2C$ Pin Configuration register | R/W | 0 | 0x4002 C07C |

# Mapa de las GPIO del LPC1769

Establezco Dirección (0 = entrada 1 = salida)

máscara (0 = enable)

Lectura/escritura de pin (FIOMASK = 0, todo habilitado)

Con un "1" Escribo un 1

Con un "1" escribo un 0

**Table 101. GPIO register map (local bus accessible registers - enhanced GPIO features)**

| Generic Name | Description | Access | Reset value[1] | PORTn Register Name & Address |
|---|---|---|---|---|
| FIODIR | Fast GPIO Port Direction control register. This register individually controls the direction of each port pin. | R/W | 0 | FIO0DIR - 0x2009 C000<br>FIO1DIR - 0x2009 C020<br>FIO2DIR - 0x2009 C040<br>FIO3DIR - 0x2009 C060<br>FIO4DIR - 0x2009 C080 |
| FIOMASK | Fast Mask register for port. Writes, sets, clears, and reads to port (done via writes to FIOPIN, FIOSET, and FIOCLR, and reads of FIOPIN) alter or return only the bits enabled by zeros in this register. | R/W | 0 | FIO0MASK - 0x2009 C010<br>FIO1MASK - 0x2009 C030<br>FIO2MASK - 0x2009 C050<br>FIO3MASK - 0x2009 C070<br>FIO4MASK - 0x2009 C090 |
| FIOPIN | Fast Port Pin value register using FIOMASK. The current state of digital port pins can be read from this register, regardless of pin direction or alternate function selection (as long as pins are not configured as an input to ADC). The value read is masked by ANDing with inverted FIOMASK. Writing to this register places corresponding values in all bits enabled by zeros in FIOMASK.<br>**Important:** if an FIOPIN register is read, its bit(s) masked with 1 in the FIOMASK register will be read as 0 regardless of the physical pin state. | R/W | 0 | FIO0PIN - 0x2009 C014<br>FIO1PIN - 0x2009 C034<br>FIO2PIN - 0x2009 C054<br>FIO3PIN - 0x2009 C074<br>FIO4PIN - 0x2009 C094 |
| FIOSET | Fast Port Output Set register using FIOMASK. This register controls the state of output pins. Writing 1s produces highs at the corresponding port pins. Writing 0s has no effect. Reading this register returns the current contents of the port output register. Only bits enabled by 0 in FIOMASK can be altered. | R/W | 0 | FIO0SET - 0x2009 C018<br>FIO1SET - 0x2009 C038<br>FIO2SET - 0x2009 C058<br>FIO3SET - 0x2009 C078<br>FIO4SET - 0x2009 C098 |
| FIOCLR | Fast Port Output Clear register using FIOMASK. This register controls the state of output pins. Writing 1s produces lows at the corresponding port pins. Writing 0s has no effect. Only bits enabled by 0 in FIOMASK can be altered. | WO | 0 | FIO0CLR - 0x2009 C01C<br>FIO1CLR - 0x2009 C03C<br>FIO2CLR - 0x2009 C05C<br>FIO3CLR - 0x2009 C07C<br>FIO4CLR - 0x2009 C09C |

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

# CORTEX NXP1769: HOJAS DE DATOS PARA LA REALIZACIÓN DE EXÁMENES DE INFO II

# Interrupciones

Table 51. NVIC register map

| Name | Description | Access | Reset value | Address |
|---|---|---|---|---|
| ISER0 to ISER1 | Interrupt Set-Enable Registers. These 2 registers allow enabling interrupts and reading back the interrupt enables for specific peripheral functions. | RW | 0 | ISER0 - 0xE000 E100<br>ISER1 - 0xE000 E104 |
| ICER0 to ICER1 | Interrupt Clear-Enable Registers. These 2 registers allow disabling interrupts and reading back the interrupt enables for specific peripheral functions. | RW | 0 | ICER0 - 0xE000 E180<br>ICER1 - 0xE000 E184 |
| ISPR0 to ISPR1 | Interrupt Set-Pending Registers. These 2 registers allow changing the interrupt state to pending and reading back the interrupt pending state for specific peripheral functions. | RW | 0 | ISPR0 - 0xE000 E200<br>ISPR1 - 0xE000 E204 |
| ICPR0 to ICPR1 | Interrupt Clear-Pending Registers. These 2 registers allow changing the interrupt state to not pending and reading back the interrupt pending state for specific peripheral functions. | RW | 0 | ICPR0 - 0xE000 E280<br>ICPR1 - 0xE000 E284 |
| IABR0 to IABR1 | Interrupt Active Bit Registers. These 2 registers allow reading the current interrupt active state for specific peripheral functions. | RO | 0 | IABR0 - 0xE000 E300<br>IABR1 - 0xE000 E304 |

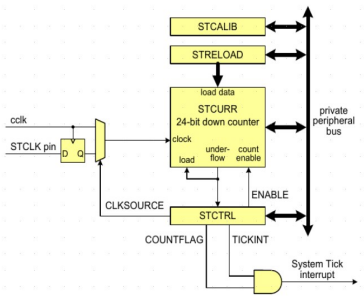
| Interrupción EXTERNA | Puerto |
|---|---|
| EINT0 | P210 |
| EINT1 | P211 |
| EINT2 | P212 |
| EINT3 | P213 |

```
    void ResetISR(void);
WEAK void NMI_Handler(void);
WEAK void HardFault_Handler(void);
WEAK void MemManage_Handler(void);
WEAK void BusFault_Handler(void);
WEAK void UsageFault_Handler(void);
WEAK void SVCall_Handler(void);
WEAK void DebugMon_Handler(void);
WEAK void PendSV_Handler(void);
WEAK void SysTick_Handler(void);
WEAK void IntDefaultHandler(void);
```

```
void TIMER0_IRQHandler(void) ALIAS(IntDefaultHandler);
void TIMER1_IRQHandler(void) ALIAS(IntDefaultHandler);
void TIMER2_IRQHandler(void) ALIAS(IntDefaultHandler);
void TIMER3_IRQHandler(void) ALIAS(IntDefaultHandler);
void UART0_IRQHandler(void) ALIAS(IntDefaultHandler);
void UART1_IRQHandler(void) ALIAS(IntDefaultHandler);
void UART2_IRQHandler(void) ALIAS(IntDefaultHandler);
void UART3_IRQHandler(void) ALIAS(IntDefaultHandler);
void PWM1_IRQHandler(void) ALIAS(IntDefaultHandler);
void I2C0_IRQHandler(void) ALIAS(IntDefaultHandler);
void I2C1_IRQHandler(void) ALIAS(IntDefaultHandler);
void I2C2_IRQHandler(void) ALIAS(IntDefaultHandler);
void SPI_IRQHandler(void) ALIAS(IntDefaultHandler);
void SSP0_IRQHandler(void) ALIAS(IntDefaultHandler);
void SSP1_IRQHandler(void) ALIAS(IntDefaultHandler);
void PLL0_IRQHandler(void) ALIAS(IntDefaultHandler);
void RTC_IRQHandler(void) ALIAS(IntDefaultHandler);
void EINT0_IRQHandler(void) ALIAS(IntDefaultHandler);
void EINT1_IRQHandler(void) ALIAS(IntDefaultHandler);
void EINT2_IRQHandler(void) ALIAS(IntDefaultHandler);
void EINT3_IRQHandler(void) ALIAS(IntDefaultHandler);
void ADC_IRQHandler(void) ALIAS(IntDefaultHandler);
void BOD_IRQHandler(void) ALIAS(IntDefaultHandler);
void USB_IRQHandler(void) ALIAS(IntDefaultHandler);
void CAN_IRQHandler(void) ALIAS(IntDefaultHandler);
void DMA_IRQHandler(void) ALIAS(IntDefaultHandler);
void I2S_IRQHandler(void) ALIAS(IntDefaultHandler);
void ENET_IRQHandler(void) ALIAS(IntDefaultHandler);
void RIT_IRQHandler(void) ALIAS(IntDefaultHandler);
void MCPWM_IRQHandler(void) ALIAS(IntDefaultHandler);
void QEI_IRQHandler(void) ALIAS(IntDefaultHandler);
void PLL1_IRQHandler(void) ALIAS(IntDefaultHandler);
void USBActivity_IRQHandler(void) ALIAS(IntDefaultHand
```

# Systick

## Timer descendente de 24 bits



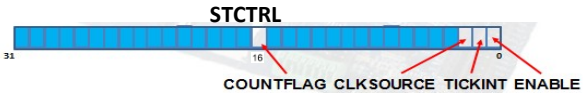




Table 438. System Tick Timer register map

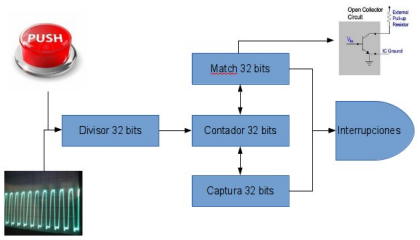
| Name | Description | Access | Reset value[1] | Address |
|---|---|---|---|---|
| STCTRL | System Timer Control and status register | R/W | 0x4 | 0xE000 E010 |
| STRELOAD | System Timer Reload value register | R/W | 0 | 0xE000 E014 |
| STCURR | System Timer Current value register | R/W | 0 | 0xE000 E018 |
| STCALIB | System Timer Calibration value register | R/W | 0x000F 423F | 0xE000 E01C |

Clock por defecto: Interno (CCLK)
Si se elige clock externo: STCLK (P3.26) => Seleccionar al pin STCLK en el registro pinMode

STRELOAD = ( STCALIB / N ) - 1 ☛ Si N=1 ➔ tick cada 10ms. (si clock=100Mhz)

# CORTEX NXP1769: HOJAS DE DATOS PARA LA REALIZACIÓN DE EXÁMENES DE INFO II

## Timers




- 2 canales de captura, que sacan una foto del timer ante una transición de una entrada. Puede generar, en forma opcional, una interrupción.
- Cuatro registros de match de 32-bit que permiten:
  - Operación contínua con generación opcional de interrupción on match.
  - Frena timer on match con interrupción opcional.
  - Reset timer on match interrupción opcional.
- Hasta cuatro salidas externas correspondientes a los registros de match con las siguientes capacidades:
  - Set low on match.
  - Set high on match.
  - Toggle on match.
  - No hace nada on match.

| Generic Name | Description | Access | Reset Value[1] | TIMERn Register/ Name & Address |
|---|---|---|---|---|
| IR | Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending. | R/W | 0 | T0IR - 0x4000 4000<br>T1IR - 0x4000 8000<br>T2IR - 0x4009 0000<br>T3IR - 0x4009 4000 |
| TCR | Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR. | R/W | 0 | T0TCR - 0x4000 4004<br>T1TCR - 0x4000 8004<br>T2TCR - 0x4009 0004<br>T3TCR - 0x4009 4004 |
| TC | Timer Counter. The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR. | R/W | 0 | T0TC - 0x4000 4008<br>T1TC - 0x4000 8008<br>T2TC - 0x4009 0008<br>T3TC - 0x4009 4008 |
| MCR | Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs. | R/W | 0 | T0MCR - 0x4000 4014<br>T1MCR - 0x4000 8014<br>T2MCR - 0x4009 0014<br>T3MCR - 0x4009 4014 |
| MR0 | Match Register 0. MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC. | R/W | 0 | T0MR0 - 0x4000 4018<br>T1MR0 - 0x4000 8018<br>T2MR0 - 0x4009 0018<br>T3MR0 - 0x4009 4018 |
| MR1 | Match Register 1. See MR0 description. | R/W | 0 | T0MR1 - 0x4000 401C<br>T1MR1 - 0x4000 801C<br>T2MR1 - 0x4009 001C<br>T3MR1 - 0x4009 401C |
| MR2 | Match Register 2. See MR0 description. | R/W | 0 | T0MR2 - 0x4000 4020<br>T1MR2 - 0x4000 8020<br>T2MR2 - 0x4009 0020<br>T3MR2 - 0x4009 4020 |
| MR3 | Match Register 3. See MR0 description. | R/W | 0 | T0MR3 - 0x4000 4024<br>T1MR3 - 0x4000 8024<br>T2MR3 - 0x4009 0024<br>T3MR3 - 0x4009 4024 |
| CCR | Capture Control Register. The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place. | R/W | 0 | T0CCR - 0x4000 4028<br>T1CCR - 0x4000 8028<br>T2CCR - 0x4009 0028<br>T3CCR - 0x4009 4028 |

TCR

| Bit | Symbol | Description | Reset Value |
|---|---|---|---|
| 0 | Counter Enable | When one, the Timer Counter and Prescale Counter are enabled for counting. When 1, the counters are disabled. | 0 |
| 1 | Counter Reset | When one, the Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero. | 0 |
| 31:2 | - | Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined. | NA |

IR

| Bit | Symbol | Description | Reset Value |
|---|---|---|---|
| 0 | MR0 Interrupt | Interrupt flag for match channel 0. | 0 |
| 1 | MR1 Interrupt | Interrupt flag for match channel 1. | 0 |
| 2 | MR2 Interrupt | Interrupt flag for match channel 2. | 0 |
| 3 | MR3 Interrupt | Interrupt flag for match channel 3. | 0 |
| 4 | CR0 Interrupt | Interrupt flag for capture channel 0 event. | 0 |
| 5 | CR1 Interrupt | Interrupt flag for capture channel 1 event. | 0 |
| 31:6 | - | Reserved | - |

MCR

| Bit | Symbol | Value | Description | Reset Value |
|---|---|---|---|---|
| 0 | MR0I | 1 | Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC. | 0 |
| | | 0 | This interrupt is disabled | |
| 1 | MR0R | 1 | Reset on MR0: the TC will be reset if MR0 matches it. | 0 |
| | | 0 | Feature disabled. | |
| 2 | MR0S | 1 | Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC. | 0 |
| | | 0 | Feature disabled. | |
| 3 | MR1I | 1 | Interrupt on MR1: an interrupt is generated when MR1 matches the value in the TC. | 0 |
| | | 0 | This interrupt is disabled | |
| 4 | MR1R | 1 | Reset on MR1: the TC will be reset if MR1 matches it. | 0 |
| | | 0 | Feature disabled. | |
| 5 | MR1S | 1 | Stop on MR1: the TC and PC will be stopped and TCR[0] will be set to 0 if MR1 matches the TC. | 0 |
| | | 0 | Feature disabled. | |

PCLK0

Table 40. Peripheral Clock Selection register 0 (PCLKSEL0 - address 0x400F C1A8) bit description

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 1:0 | PCLK_WDT | Peripheral clock selection for WDT. | 00 |
| 3:2 | PCLK_TIMER0 | Peripheral clock selection for TIMER0. | 00 |
| 5:4 | PCLK_TIMER1 | Peripheral clock selection for TIMER1. | 00 |
| 7:6 | PCLK_UART0 | Peripheral clock selection for UART0. | 00 |
| 9:8 | PCLK_UART1 | Peripheral clock selection for UART1. | 00 |
| 11:10 | - | Reserved. | NA |
| 13:12 | PCLK_PWM1 | Peripheral clock selection for PWM1. | 00 |
| 15:14 | PCLK_I2C0 | Peripheral clock selection for I²C0. | 00 |
| 17:16 | PCLK_SPI | Peripheral clock selection for SPI. | 00 |
| 19:18 | - | Reserved. | NA |
| 21:20 | PCLK_SSP1 | Peripheral clock selection for SSP1. | 00 |
| 23:22 | PCLK_DAC | Peripheral clock selection for DAC. | 00 |
| 25:24 | PCLK_ADC | Peripheral clock selection for ADC. | 00 |

PCONP

Table 46. Power Control for Peripherals register (PCONP - address 0x400F C0C4) bit description

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 0 | - | Reserved. | NA |
| 1 | PCTIM0 | Timer/Counter 0 power/clock control bit. | 1 |
| 2 | PCTIM1 | Timer/Counter 1 power/clock control bit. | 1 |
| 3 | PCUART0 | UART0 power/clock control bit. | 1 |
| 4 | PCUART1 | UART1 power/clock control bit. | 1 |
| 5 | - | Reserved. | NA |
| 6 | PCPWM1 | PWM1 power/clock control bit. | 1 |
| 7 | PCI2C0 | The I²C0 interface power/clock control bit. | 1 |
| 8 | PCSPI | The SPI interface power/clock control bit. | 1 |
| 9 | PCRTC | The RTC power/clock control bit. | 1 |
| 10 | PCSSP1 | The SSP 1 interface power/clock control bit. | 1 |
| 11 | - | Reserved. | NA |
| 12 | PCADC | A/D converter (ADC) power/clock control bit.<br>**Note:** Clear the PDN bit in the AD0CR before clearing this bit, and set this bit before setting PDN. | 0 |
| 13 | PCCAN1 | CAN Controller 1 power/clock control bit. | 0 |
| 14 | PCCAN2 | CAN Controller 2 power/clock control bit. | 0 |
| 15 | PCGPIO | Power/clock control bit for IOCON, GPIO, and GPIO interrupts. | 1 |
| 16 | PCRIT | Repetitive Interrupt Timer power/clock control bit. | 0 |

PCONP

Table 46. Power Control for Peripherals register (PCONP - address 0x400F C0C4) bit description

| Bit | Symbol | Description | Reset value |
|---|---|---|---|
| 17 | PCMCPWM | Motor Control PWM | 0 |
| 18 | PCQEI | Quadrature Encoder Interface power/clock control bit. | 0 |
| 19 | PCI2C1 | The I²C1 interface power/clock control bit. | 1 |
| 20 | - | Reserved. | NA |
| 21 | PCSSP0 | The SSP0 interface power/clock control bit. | 1 |
| 22 | PCTIM2 | Timer 2 power/clock control bit. | 0 |
| 23 | PCTIM3 | Timer 3 power/clock control bit. | 0 |
| 24 | PCUART2 | UART 2 power/clock control bit. | 0 |
| 25 | PCUART3 | UART 3 power/clock control bit. | 0 |
| 26 | PCI2C2 | I²C interface 2 power/clock control bit. | 1 |
| 27 | PCI2S | I²S interface power/clock control bit. | 0 |
| 28 | - | Reserved. | NA |
| 29 | PCGPDMA | GPDMA function power/clock control bit. | 0 |
| 30 | PCENET | Ethernet block power/clock control bit. | 0 |
| 31 | PCUSB | USB interface power/clock control bit. | 0 |

# CORTEX NXP1769: HOJAS DE DATOS PARA LA REALIZACIÓN DE EXÁMENES DE INFO II

## Puerto Serie

$$UART1_{baudrate} = \frac{PCLK}{16 \times (256 \times U1DLM + U1DLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)}$$

```
#define     DIR_UART0     ( ( volatile uint32_t  * ) 0x4000C000UL )
#define     DIR_UART1     ( ( volatile uint32_t  * ) 0x40010000UL )
```

**PUERTO SERIE**: La dirección de los puertos serie del sistema, se encuentra a partir de la dirección 0x0400 de memoria

**Line Control Register: LCR – BASE + 3 (activo alto)**

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|---|---|

**Word Length (WLS0 – WLS1)**

| B1 | B0 | Character length |
|---|---|---|
| 0 | 0 | 5 |
| 0 | 1 | 6 |
| 1 | 0 | 7 |
| 1 | 1 | 8 |

**Number of Stop Bits (STB)**

| B2 | Word length | Stop bits |
|---|---|---|
| 0 | 5,6,7 o 8 | 1 |
| 1 | 5 | 1,5 |
| 1 | 6,7 o 8 | 2 |

**Party Enable (PEN)**

| B3 | Paridad |
|---|---|
| 0 | OFF |
| 1 | ON |

**Even Parity Select (EPS)**

| B4 | Parity |
|---|---|
| 0 | Odd |
| 1 | Even |

**Stick Parity**

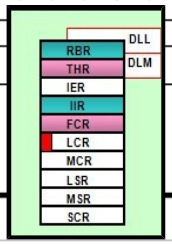
| B5 | B4 | B3 | Logic of parity bit |
|---|---|---|---|
| 1 | 1 | 1 | Logic 0 |
| 1 | 0 | 1 | Logic 1 |
| 0 | X | X | Disable |

**Set Break**

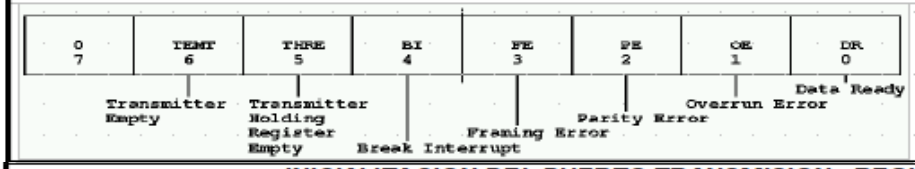
| B6 | Causes break conditions (SOUT) |
|---|---|
| 0 | Disable |
| 1 | Enable |

**Divisor Latch Access Bit (DLAB)**

| B7 | Access |
|---|---|
| 0 | Receiver Buffer and Interrupt Enable Register |
| 1 | Divisor Latches of the Baud Generator |




**Line Status Register: LSR – BASE + 5 (activo alto)**




**Reset by HW**: Por el solo hecho de leerlo se resetea

**INICIALIZACION DEL PUERTO TRANSMISION - REGISTROS**

(2) Serializador
(1) Registro Tx

Cuando pasa de (1) a (2) se activa THRE (BASE+5, bit 5)
Cuando sale de (2) se activa TEMT (BASE + 5 , bit 6)

**INICIALIZACION DEL PUERTO RECEPCION - REGISTROS**

(2) Paralelizador
(1) Registro Rx

Cuando pasa de (2) a (1) se activa DR (BASE + 5 , bit 0)

**Interrupt Enable Register: IER– BASE + 1 (activo alto)**

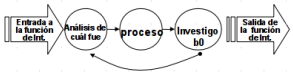
**Reset by HW**: Por el solo hecho de leerlo se resetea


IER0: A 1 si habilitar interrupción de dato disponible.
IER1: A 1 si habilitar interrupción de registro de retención de transmisión vacío.
IER2: A 1 si habilitar interrupción de error de recepción (bits 1 al 4 del LSR).
IER3: A 1 si habilitar interrupción ante el cambio del MSR (Registro de estado del modem).

**Interrupt Ident Register: IIR– BASE + 2 (activo alto)**

| | | 0 | 0 | | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|

**Interrupt Pending**




b0 = 0 => Hay una interrupción pendiente de atención
b0 = 1 => No hay una interrupción pendiente

| b2 | b1 | Descripción |
|---|---|---|
| 1 | 1 | Errores y break (>) |
| 1 | 0 | Dato disponible |
| 0 | 1 | THR disponible |
| 0 | 0 | Estado del MODEM (<) |