Microcontroladores

Semana 13

Profesor Kalun José Lau Gan

Preguntas previas

- Consultas con respecto al desarrollo de TF

Sobre el indicador de temperatura

2C0024h-2C0025h	TSLR1(1)	Gain = $\frac{0.1C \times 256}{count}$ (low range setting)
2C0026h-2C0027h	TSLR2(1)	Temperature indicator ADC reading at 90°C (low range setting)
2C0028h-2C0029h	TSLR3(1)	Offset (low range setting)
2C002Ah-2C002Bh	TSHR1(2)	Gain = $\frac{0.1C \times 256}{count}$ (high range setting)
2C002Ch-2C002Dh	TSHR2(2)	Temperature indicator ADC reading at 90°C (high range setting)
2C002Eh-2C002Fh	TSHR3(2)	Offset (high range setting)
2C0030h-2C0031h	FVRA1X	ADC FVR1 Output voltage for 1x setting (in mV)
2C0032h-2C0033h	FVRA2X	ADC FVR1 Output Voltage for 2x setting (in mV)
2C0034h-2C0035h	FVRA4X	ADC FVR1 Output Voltage for 4x setting (in mV)
2C0036h-2C0037h	FVRC1X	Comparator FVR2 output voltage for 1x setting (in mV)
2C0038h-2C0039h	FVRC2X	Comparator FVR2 output voltage for 2x setting (in mV)
2C003Ah-2C003Bh	FVRC4X	Comparator FVR2 output voltage for 4x setting (in mV)

Notes:
1. TSLR: Address 2C0024h-2C0029h store the measurements for the low range setting of the temperature sensor at V_{DD} = 3V, V_{REF}+ = 2.048V from FVR1.
2. TSHR: Address 2C002Ah-2C002Fh store the measurements for the high range setting of the temperature sensor at V_{DD} = 3V, V_{REF}+ = 2.048V from FVR1.

DIA – Se ubican los valores de ganancia y offset para el indicador de temperatura

Estos datos están ubicados en la memoria de programa por lo que para acceder a estos valores se requiere emplear el puntero de tabla (TBLPTR)

3

Statement de switch/case con rango de valores

```
//LATC = 0x3A cuando valor se encuentre entre 50 y 100

unsigned int valor = 50;

void main(void){
        while(1){
                switch(valor){
                        case 50 ... 100:
                                LATC = 0x02;
                                break;
                }
        }
}


void main(void){
        while(1){
                if(valor >= 50 && valor <=100){
                LATC = 0x3A;
        }
}
```

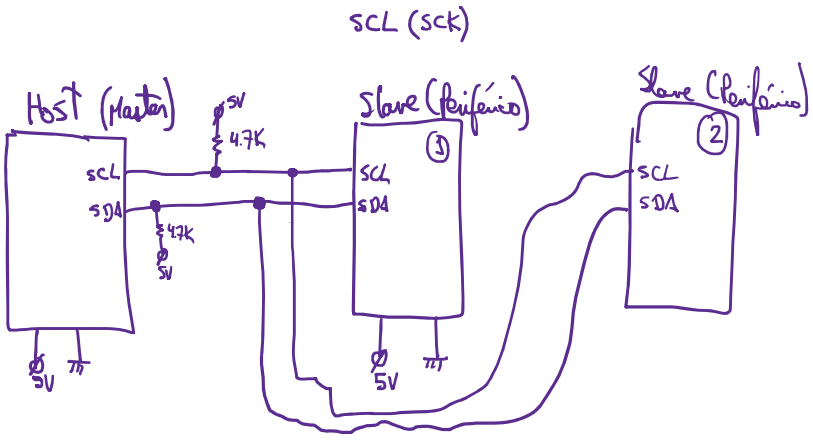
4

Agenda:

- Semana 13:
  - Comunicación serial:
    - I2C (teoría)
    - UART (laboratorio)
    - Protocolo para el DHT11/DHT22
    - 1-wire (DS18B20) – Parasite power
- Semana 14:
  - PC2 (teoría)
  - I2C (laboratorio)
- Semana 15:
  - Microcontroladores de 32bits (teoría)
  - DD (laboratorio)
- Semana 16:
  - TF

Comunicación I2C

- Comunicación desarrollada para la comunicación entre circuitos integrados. Phillips Semiconductor (NXP)
- I2C – I2S -> Sonido
- Podemos construir una red de dispositivos (topología bus)

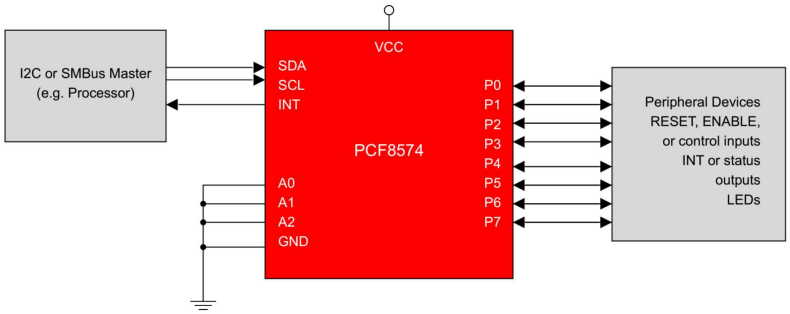


Comunicación I2C

- Direcciones de 7 bits ó 10 bits
- En el I2C tenemos condiciones:
  - Condición **START**: Inicio de la comunicación I2C.
  - Condición **RESTART**: Resetear la línea de comunicación, generalmente se emplea para cambiar entre escritura y lectura dentro de un evento de comunicación de un periférico.
  - Condición **STOP**: Término de la comunicación I2C.
  - **ACK** (Aknowledge): Evento de confirmación.
  - **NOACK** (No Acknowledge): Evento de no confirmación.
- Se debe de revisar la hoja técnica del periférico I2C para ver la trama I2C que necesita para comunicarse

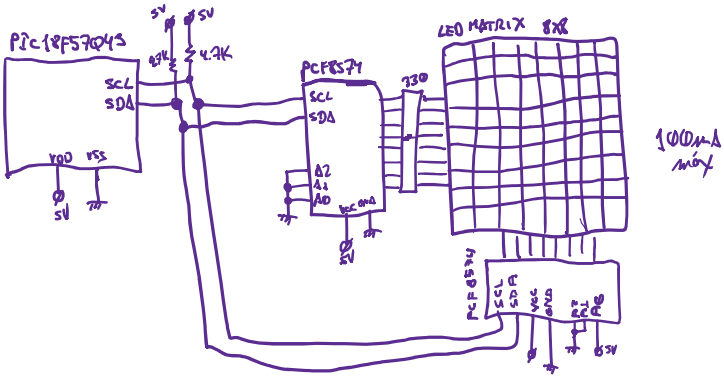
Caso: PCF8574

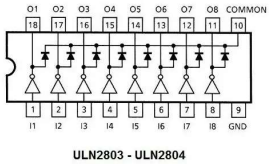
- Expansor de 8 bits de E/S I2C
- Hasta 8 dispositivos PCF8574 en un bus I2C



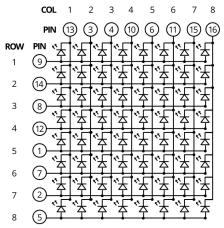
Caso: PCF8574

- Expansor de 8 bits de E/S I2C
- Hasta 8 dispositivos PCF8574 en un bus I2C





ULN2803 - ULN2804



9

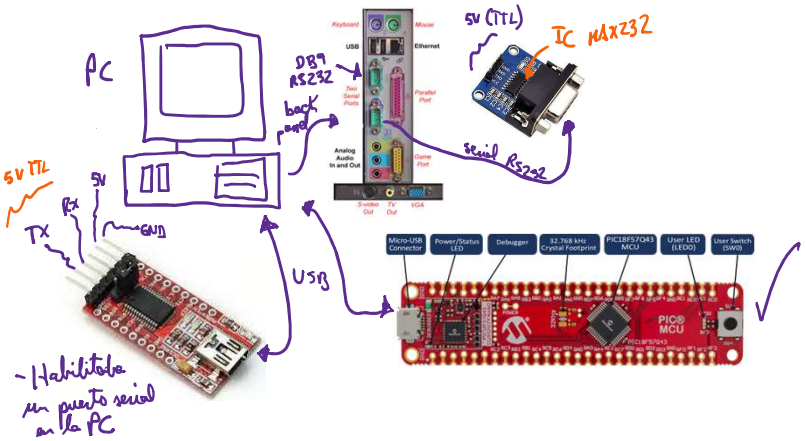
Caso: PCF8574

- Tarjeta I2C para display LCD 16x2 HD44780
- Posee un conector de 4 pines (Vcc, Gnd, SDA y SCL)
- Utiliza un PCF8574!

10

Ejemplo: Comunicación serial UART

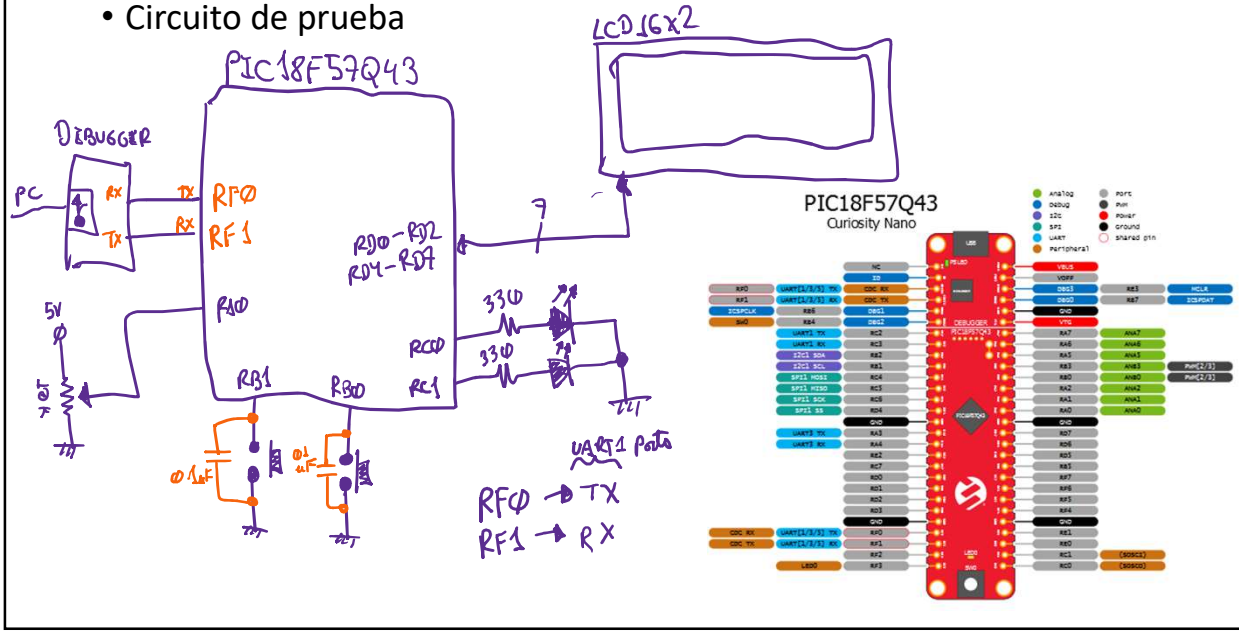
- Curiosity Nano -> PC



11

Ejemplo: Comunicación serial UART

- Circuito de prueba



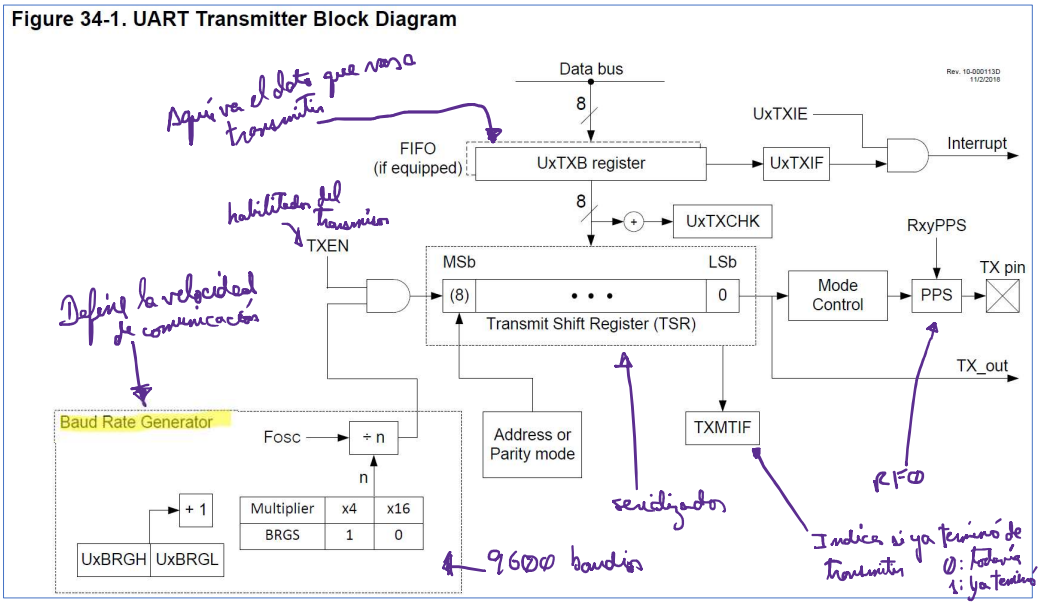
12

Sobre el UART en el PIC!8F57Q43

- Hay 5 módulos UART:
  - 3 full featured (U1, U3, U5)
  - 2 non full featured (U2, U4)

Diagrama de bloques del transmisor del UART

Figure 34-1. UART Transmitter Block Diagram



Fórmula para calcular el BRG:

The UART Baud Rate equals [Fosc*(1+(BRGS*3)]/[(16*(BRG+1))]

Tenemos que despejar BRG

Baud rate = 9600

Fosc = 32MHz

U1BRGH:U1BRGL

BRGH:BRGL

0 207

BRGS = 0

$9600 = \frac{[32E^6 (1 + 0 \times 3)]}{16 \times (BRG+1)}$

$BRG+1 = \frac{[32E^6 (1)]}{16 \times 9600}$

$BRG = \left(\frac{32000000}{153600}\right) - 1 = 207.3333$

Para 19200:

$BRG+1 = \frac{[32E^6 (1)]}{16 \times 19200}$

$BRG = \left(\frac{32000000}{307200}\right) - 1$

BRG = 103

Para 38400:

$BRG = \left(\frac{32000000}{16 \times 38400}\right) - 1$

$BRG = \left(\frac{32000000}{614400}\right) - 1$

BRG = 51

15

Registro U1CON0

- Se configura el BRGS, los habilitadores de TX y RX y el modo de trabajo del UART1

Name: U1CON0
Address: 0x2AB,0x2BE,0x2D1,0x2E4,0x2F7

UART Control Register 0

Bit	7	6	5	4	3	2	1	0
	BRGS	ABDEN	TXEN	RXEN		MODE[3:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0 0 1 0 0 0 0 0 = 0x20

Bit 7 – BRGS Baud Rate Generator Speed Select

Value	Description
1	Baud Rate Generator is high speed with 4 baud clocks per bit
0	Baud Rate Generator is normal speed with 16 baud clocks per bit

Bit 6 – ABDEN Auto-Baud Detect Enable(3)

Value	Description
1	Auto-baud is enabled. Receiver is waiting for Sync character (0x55).
0	Auto-baud is not enabled or auto-baud is complete

Bit 5 – TXEN Transmit Enable Control(2)

Value	Description
1	Transmit is enabled. TX output pin drive is forced on when transmission is active, and controlled by PORT TRIS control when transmission is Idle.
0	Transmit is disabled. TX output pin drive is controlled by PORT TRIS control.

Bit 4 – RXEN Receive Enable Control(2)

Value	Description
1	Receiver is enabled
0	Receiver is disabled

Bits 3:0 – MODE[3:0] UART Mode Select(1)

Value	Description
1111 – 1101	Reserved
1100	LIN Host/Client mode(4)
1011	LIN Client Only mode(4)
1010	DMX mode(4)
1001	DALI Control Gear mode(4)
1000	DALI Control Device mode(4)
0111 – 0101	Reserved
0100	Asynchronous 9-bit UART Address mode, 9th bit: 1 = address, 0 = data
0011	Asynchronous 8-bit UART mode with 9th bit even parity
0010	Asynchronous 8-bit UART mode with 9th bit odd parity
0001	Asynchronous 7-bit UART mode
0000	Asynchronous 8-bit UART mode

16

Registro U1CON1

- Se configura el habilitador del UART1

Name: UxCON1
Address: 0x2AC,0x2BF,0x2D2,0x2E5,0x2F8

UART Control Register 1

Bit	7	6	5	4	3	2	1	0
	ON			WUE	RXBIMD		BRKOVR	SENDB
Access	R/W			R/W/HC	R/W		R/W	R/W/HC
Reset	0			0	0		0	0

Bit 7 – ON Serial Port Enable

Value	Description
1	Serial port enabled
0	Serial port disabled (held in Reset)

Bit 4 – WUE Wake-Up Enable

Value	Description
1	Receiver is waiting for falling RX input edge which will set the UxIF bit. Cleared by hardware on wake-up event. Also requires the UxIE bit of PIEx to enable wake.
0	Receiver operates normally

Bit 3 – RXBIMD Receive Break Interrupt Mode Select

Value	Description
1	Set RXBKIF immediately when RX in has been low for the minimum Break time
0	Set RXBKIF on rising RX input after RX in has been low for the minimum Break time

Bit 1 – BRKOVR Send Break Software Override

Value	Description
1	TX output is forced to non-Idle state
0	TX output is driven by transmit shift register

Bit 0 – SENDB Send Break Control(1)

Value	Description
1	Output Break upon UxTXB write. Written byte follows Break. Bit is cleared by hardware.
0	Break transmission completed or disabled

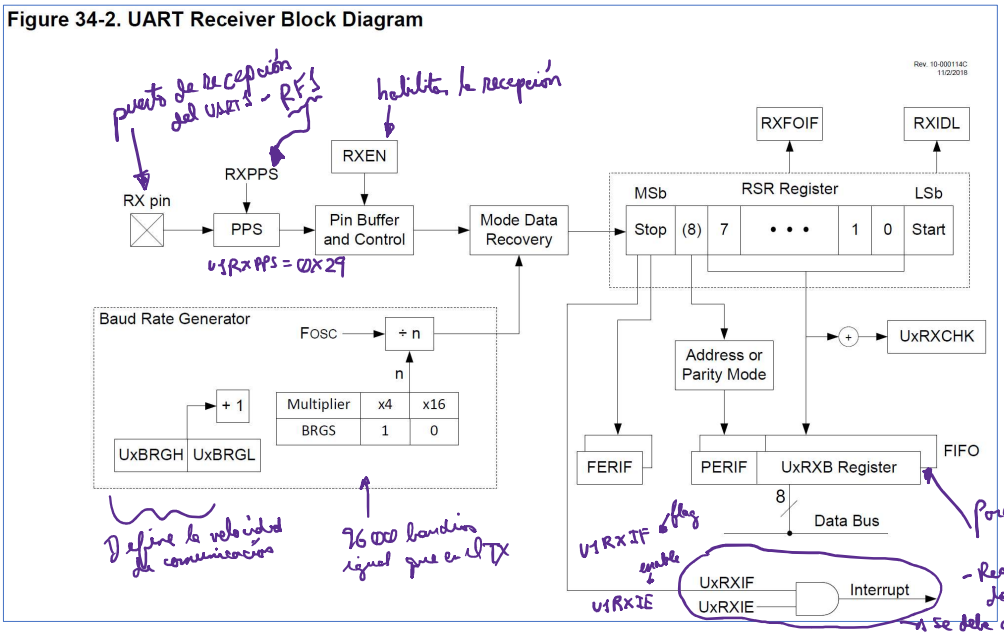
Registro U1CON2

- Lo vamos a dejar con los valores por defecto (0x00)

Configuración del UART1 para transmitir un dato:

- Primero: Puertos de E/S:
  - En el Curiosity Nano RF0 es el TX, RF1 es el RX
- Segundo: Establecer velocidad con U1BRG
- Tercero: Configurar registros U1CONx:
  - U1CON0: BRGS (normal/high), TX enable, RX enable, UART mode
  - U1CON1: ON (serial port enable)
  - U1CON2: déjalo con los valores por defecto
- Cuarto: El PPS para TX
  - En el Curiosity Nano: RF0PPS = 0x20;
- Quinto: Para transmitir un dato de 8 bits
  - Colocar el dato en U1TXB
  - Esperar a que se termine de transmitir (U1ERRIR bit TXMTIF = 1)

Diagrama de bloques del receptor del UART



Configuración del PPS de entrada del UART1 RX con el puerto RF1

- Esta por defecto el RC7 para UART1 RX

Peripheral	PPS Input Register	Default Pin Selection at POR	Register Reset Value at POR	Available Input Port											
				28-Pin Devices			40-Pin Devices				48-Pin Devices				
SPI1 Client Select	SPI1SSPPS	RA5	'b000 101	A	—	C	A	—	—	D	—	A	—	—	D	—	—
SPI2 Clock	SPI2SCKPPS	RB3	'b001 011	—	B	C	—	B	—	D	—	—	B	—	D	—	—
SPI2 Data	SPI2SDIPPS	RB2	'b001 010	—	B	C	—	B	—	D	—	—	B	—	D	—	—
SPI2 Client Select	SPI2SSPPS	RA4	'b000 100	A	—	C	A	—	—	D	—	A	—	—	D	—	—
I2C1 Clock	I2C1SCLPPS(1)	RC3	'b010 011	—	B	C	—	B	C	—	—	—	B	C	—	—	—
I2C1 Data	I2C1SDAPPS(1)	RC4	'b010 100	—	B	C	—	B	C	—	—	—	B	C	—	—	—
UART1 Receive	U1RXPPS	RC7	'b010 111	—	B	C	—	B	C	—	—	—	—	C	—	—	F
UART1 Clear to Send	U1CTSPPS	RC6	'b010 110	—	B	C	—	B	C	—	—	—	—	C	—	—	F
UART2 Receive	U2RXPPS	RB7	'b001 111	—	B	C	—	B	—	D	—	—	B	—	D	—	—
UART2 Clear to Send	U2CTSPPS	RB6	'b001 110	—	B	C	—	B	—	D	—	—	B	—	D	—	—
UART3 Receive	U3RXPPS	RA7	'b000 111	A	B	—	A	B	—	—	—	A	—	—	—	—	F
UART3 Clear to Send	U3CTSPPS	RA6	'b000 110	A	B	—	A	B	—	—	—	A	—	—	—	—	F
UART4 Receive	U4RXPPS	RB5	'b001 101	—	B	C	—	B	—	D	—	—	B	—	D	—	—
UART4 Clear to Send	U4CTSPPS	RB4	'b001 100	—	B	C	—	B	—	D	—	—	B	—	D	—	—
UART5 Receive	U5RXPPS	RA5	'b000 101	A	—	C	A	—	C	—	—	A	—	—	—	—	F
UART5 Clear to Send	U5CTSPPS	RA4	'b000 100	A	—	C	A	—	C	—	—	A	—	—	—	—	F

21.8.1 xxxPPS

Name: xxxPPS U1RXPPS = 0x29

Peripheral Input Selection Register

Bit	7	6	5	4	3	2	1	0
			PORT[2:0]			PIN[2:0]		
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			m	m	m	m	m	m

Bits 5:3 – PORT[2:0] Peripheral Input PORT Selection(1)
See the PPS Input Selection Table for the list of available Ports and default pin locations.

PORT	Selection
101	PORTF
100	PORTE
011	PORTD
010	PORTC
001	PORTB
000	PORTA

Reset States: POR = mmm
All other Resets = uuu

Bits 2:0 – PIN[2:0] Peripheral Input PORT Pin Selection(2)
Reset States: POR = mmm
All other Resets = uuu

Value	Description
111	Peripheral input is from PORTx Pin 7 (Rx7)
110	Peripheral input is from PORTx Pin 6 (Rx6)
101	Peripheral input is from PORTx Pin 5 (Rx5)
100	Peripheral input is from PORTx Pin 4 (Rx4)
011	Peripheral input is from PORTx Pin 3 (Rx3)
010	Peripheral input is from PORTx Pin 2 (Rx2)
001	Peripheral input is from PORTx Pin 1 (Rx1)
000	Peripheral input is from PORTx Pin 0 (Rx0)

21

Registro U1CON0

- Se configura el BRGS, los habilitadores de TX y RX y el modo de trabajo del UART1

Name: UxCON0
Address: 0x2AB,0x2BE,0x2D1,0x2E4,0x2F7

U1CON = 0x30

UART Control Register 0

Bit	7	6	5	4	3	2	1	0
	BRGS	ABDEN	TXEN	RXEN	MODE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – BRGS Baud Rate Generator Speed Select

Value	Description
1	Baud Rate Generator is high speed with 4 baud clocks per bit
0	Baud Rate Generator is normal speed with 16 baud clocks per bit

Bit 6 – ABDEN Auto-Baud Detect Enable(3)

Value	Description
1	Auto-baud is enabled. Receiver is waiting for Sync character (0x55).
0	Auto-baud is not enabled or auto-baud is complete

Bit 5 – TXEN Transmit Enable Control(2)

Value	Description
1	Transmit is enabled. TX output pin drive is forced on when transmission is active, and controlled by PORT TRIS control when transmission is Idle.
0	Transmit is disabled. TX output pin drive is controlled by PORT TRIS control.

Bit 4 – RXEN Receive Enable Control(2)

Value	Description
1	Receiver is enabled
0	Receiver is disabled

Bits 3:0 – MODE[3:0] UART Mode Select(1)

Value	Description
1111 – 1101	Reserved
1100	LIN Host/Client mode(4)
1011	LIN Client Only mode(4)
1010	DMX mode(4)
1001	DALI Control Gear mode(4)
1000	DALI Control Device mode(4)
0111 – 0101	Reserved
0100	Asynchronous 9-bit UART Address mode. 9th bit: 1 = address, 0 = data
0011	Asynchronous 8-bit UART mode with 9th bit even parity
0010	Asynchronous 8-bit UART mode with 9th bit odd parity
0001	Asynchronous 7-bit UART mode
0000	Asynchronous 8-bit UART mode

22

Ubicación de la interrupción de recepción del UART1:

0x04A1	PIE3	7:0	TMR0IE	CCP1IE	TMR1GIE	TMR1IE	TMR2IE	SPI1IE	SPI1TXIE	SPI1RXIE
0x04A2	PIE4	7:0	PWM1IE	PWM1PIE			U1IE	U1EIE	U1TXIE	U1RXIE
0x04A3	PIE5	7:0	PWM2IE	PWM2PIE	TMR3GIE	TMR3IE		SPI2IE	SPI2TXIE	SPI2RXIE
0x04A4	PIE6	7:0	DMA2AIE	DMA2ORIE	DMA2DCNTIE	DMA2SCNTIE	NCO1IE	CWG1IE	CLC2IE	INT1IE
0x04A5	PIE7	7:0	PWM3IE	PWM3PIE	CLC3IE		I2C1EIE	I2C1IE	I2C1TXIE	I2C1RXIE
0x04A6	PIE8	7:0	SCANIE	CCP2IE	TMR5GIE	TMR5IE	U2IE	U2EIE	U2TXIE	U2RXIE
0x04A7	PIE9	7:0			CLC4IE		U3IE	U3EIE	U3TXIE	U3RXIE
0x04A8	PIE10	7:0	DMA3AIE	DMA3ORIE	DMA3DCNTIE	DMA3SCNTIE	NCO2IE	CWG2IE	CLC5IE	INT2IE
0x04A9	PIE11	7:0	DMA4AIE	DMA4ORIE	DMA4DCNTIE	DMA4SCNTIE	TMR4IE	CWG3IE	CLC6IE	CCP3IE
0x04AA	PIE12	7:0	DMA5AIE	DMA5ORIE	DMA5DCNTIE	DMA5SCNTIE	U4IE	U4EIE	U4TXIE	U4RXIE
0x04AB	PIE13	7:0	DMA6AIE	DMA6ORIE	DMA6DCNTIE	DMA6SCNTIE	U5IE	U5EIE	U5TXIE	U5RXIE
0x04AC	PIE14	7:0					NCO3IE	CM2IE	CLC7IE	
0x04AD	PIE15	7:0					TMR6IE	CRCIE	CLC8IE	NVMIE
0x04AE	PIR0	7:0	IOCIF		CLC1IF		CSWIF	OSFIF	HLVDIF	SWIF
0x04AF	PIR1	7:0	SMT1PWAIF	SMT1PRAIF	SMT1IF	CM1IF	ACTIF	ADIF	ZCDIF	INT0IF
0x04B0	PIR2	7:0	DMA1AIF	DMA1ORIF	DMA1DCNTIF	DMA1SCNTIF				ADTIF
0x04B1	PIR3	7:0	TMR0IF	CCP1IF	TMR1GIF	TMR1IF	TMR2IF	SPI1IF	SPI1TXIF	SPI1RXIF
0x04B2	PIR4	7:0	PWM1IF	PWM1PIF			U1IF	U1EIF	U1TXIF	U1RXIF
0x04B3	PIR5	7:0	PWM2IF	PWM2PIF	TMR3GIF	TMR3IF		SPI2IF	SPI2TXIF	SPI2RXIF

Configuración adicional del UART1 para recepcionar un dato:

- Primero: Estas configuraciones son adicionales a lo que se establecieron en la transmisión
- Segundo: Cerciorarse que el RX esté enabled en el U1CON0
- Tercero: Habilitar la interrupción de la recepción del UART1:
  - Habilitador U1RXIE se encuentra en PIE4 y bandera U1RXIF se encuentra en PIR4
  - No olvidar habilitar el GIE que esta en INTCON0.
- Cuarto: El PPS para RX
  - En el Curiosity Nano: U1RXPPS = 0x29
- Quinto: Para recibir un dato de 8 bits
  - Definir la función de interrupción de la recepción del UART1

```
void __interrupt(irq(IRQ_U1RX)) U1RX_ISR(void){
    PIR4bits.U1RXIF = 0;        //bajamos bandera de RX de UART1
```

Fin de la sesión