



UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO

Diplomado en Software Embebido

Módulo 4: Bases de Electrónica Digital

Tema: 4.1.9. Comunicaciones

José Pérez
Marcos Peña
Carlos Reyes

ÍNDICE GENERAL

4.1.8 COMUNICACIONES	1
4.1.8.1.1. Introducción	1
4.1.8.1.2. Tipos de Protocolos de Comunicación.....	1
4.1.8.1.3. Periféricos de Comunicación	3
4.1.8.1.4. UART	3
4.1.8.1.5. SPI	5
4.1.8.1.6. I2C	7
4.1.8.1.7. SCI con Renesas Synergy	9
4.1.8.1.8. SPI con Renesas Synergy	17
4.1.8.1.9. I2C con Renesas Synergy	28
4.1.8.1.10. Consideraciones de diseño.....	36
4.1.8.1.11. Ejercicios	38
4.1.8.1.12. Referencias.....	40
4.1.8.1.13. Evaluación	41

TABLA DE FIGURAS

Figura 1. Inter System Protocol	2
Figura 2. Intra System Protocol	2
Figura 3. Flujo de datos del UART	4
Figura 4. Diagrama de Bloques SCI (UART)	4
Figura 5. Flujo de datos de SPI	5
Figura 6. Diagrama de bloques SPI	6

Figura 7. Flujo de datos I2C	8
Figura 8. Diagrama de bloques I2C	8
Figura 9. Selección de SCI	9
Figura 10. Modo de operación SCI	10
Figura 11. Configuración pin TX de SCI	11
Figura 12. Configuración de PIN RX de SCI	13
Figura 13. Agregar driver de SCI	15
Figura 14. Generar código del driver de SCI	15
Figura 15. Selección de SPI	17
Figura 16. Configuración de modo de operación de SPI	18
Figura 17. Modo de operación del pin MISO	19
Figura 18. Modo de operación del pin MOSI	21
Figura 19. Modo de operación del pin SCK	23
Figura 20. Modo de operación del pin SS	24
Figura 21. Agregar driver de SPI	26
Figura 22. Generar código del driver de SPI	27
Figura 23. Selección de interfaz I2C	28
Figura 24. Modo de operación del I2C	29
Figura 25. Modo de operación del pin SDA	30

Figura 26. Modo de operación del pin SCL	32
Figura 27. Agregar driver de I2C	34
Figura 28. Generar código de I2C	35
Figura 29. Función para cálculo de CRC	37

4.1.8 COMUNICACIONES

4.1.8.1.1. Introducción

Comunicación:

El intercambio de información de un sistema a otro sistema con un medio se denomina comunicación.

Protocolo:

Un conjunto de reglas y regulaciones se llama protocolo.

Protocolo de Comunicación:

Es un conjunto de reglas y regulaciones que permiten que dos dispositivos electrónicos se conecten para intercambiar los datos entre uno y otro.

4.1.8.1.2. Tipos de Protocolos de Comunicación

Inter System Protocol:

El protocolo Inter System se utiliza para comunicar dos dispositivos diferentes, el similar a la comunicación que se da entre una computadora y un microcontrolador o entre dos sistemas embebidos, la comunicación se realiza a través de un bus (arnés) para conectar los dos dispositivos.

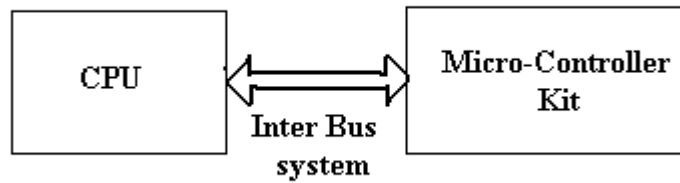


Figura 1. Inter System Protocol

Intra System Protocol:

El protocolo Intra System se utiliza normalmente para comunicar dos dispositivos dentro de la misma placa de circuito. En un Intra System aumenta la complejidad de la tarjeta electrónica y por lo tanto el consumo de corriente, pero a su vez se convierte en un sistema más seguro ya que dentro de la misma tarjeta electrónica se tiene acceso a los datos.

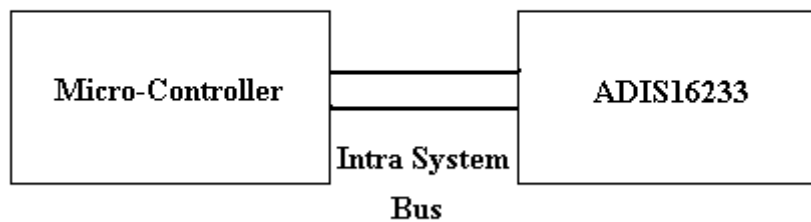


Figura 2. Intra System Protocol

4.1.8.1.3. Periféricos de Comunicación

La función de estos periféricos es establecer una interfaz de comunicación entre dispositivos a través de un protocolo de comunicación estándar. Entre los periféricos de comunicación más comunes en un microcontrolador se encuentran:

- UART
- SPI
- I2C
- CAN
- ETHERNET

4.1.8.1.4. UART

UART significa transmisor y receptor asíncrono universal. El UART es un protocolo de comunicación alámbrica, las señales de datos están etiquetadas como Rx y Tx.

La comunicación en serie se usa comúnmente para transmitir y recibir señales. Se transfieren y reciben los datos en serie bit a bit.

El UART toma bytes de datos en paralelo y envía los bits individuales de manera serial secuencial. UART es un protocolo half dúplex que significa transferir y recibir los datos, pero no al mismo tiempo.

La mayoría de los microcontroladores tienen hardware UART incluido. Se utiliza una sola línea de datos para transmitir y otra para recibir. Tiene un bit de inicio, datos de 8 bits y un bit de parada.

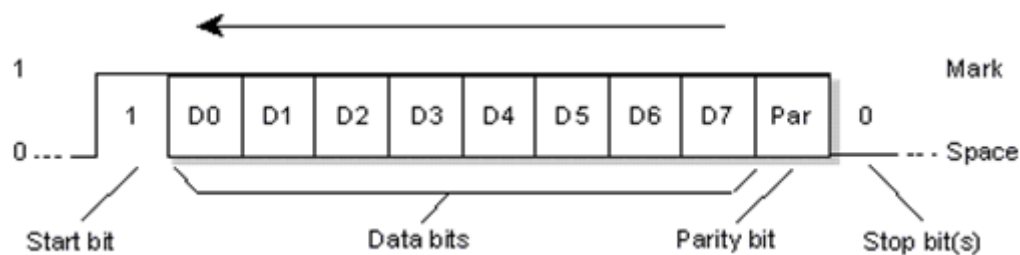


Figura 3. Flujo de datos del UART

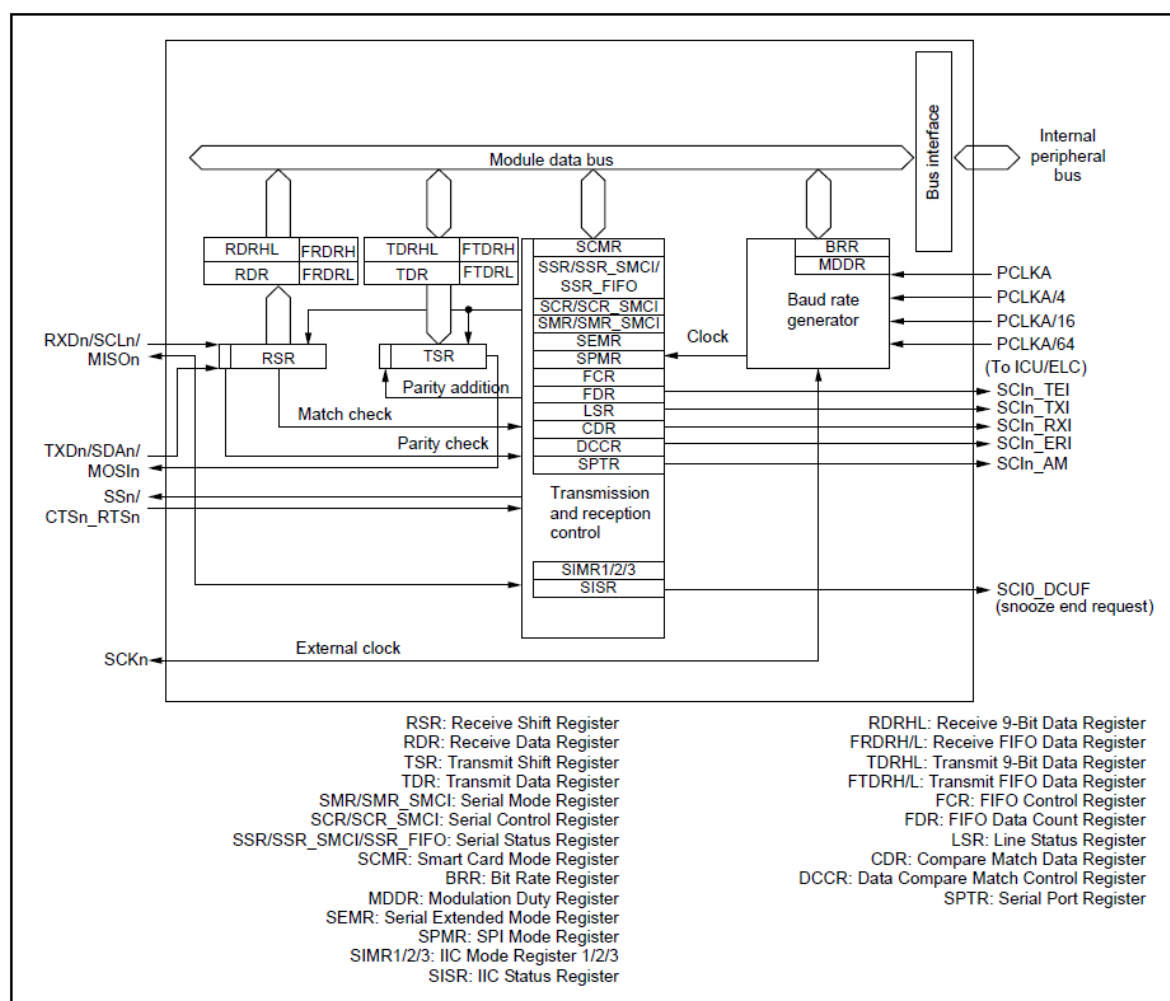


Figura 4. Diagrama de Bloques SCI (UART)

4.1.8.1.5. SPI

SPI significa interfaz periférica en serie. Es uno de los protocolos de comunicación en serie desarrollado por Motorola. Algunas veces, el protocolo SPI también se denomina protocolo de 4 hilos. Requiere cuatro señales: MOSI, MISO, SS y SCLK.

SPI es utilizado para comunicar los dispositivos maestro y esclavo. El maestro primero configura el reloj usando una frecuencia, luego selecciona el dispositivo esclavo particular para la comunicación y así inicia la comunicación entre el maestro y ese esclavo.

El maestro selecciona solo un esclavo a la vez. Es un protocolo de comunicación full dúplex no limitado a palabras de 8 bits de transferencia.

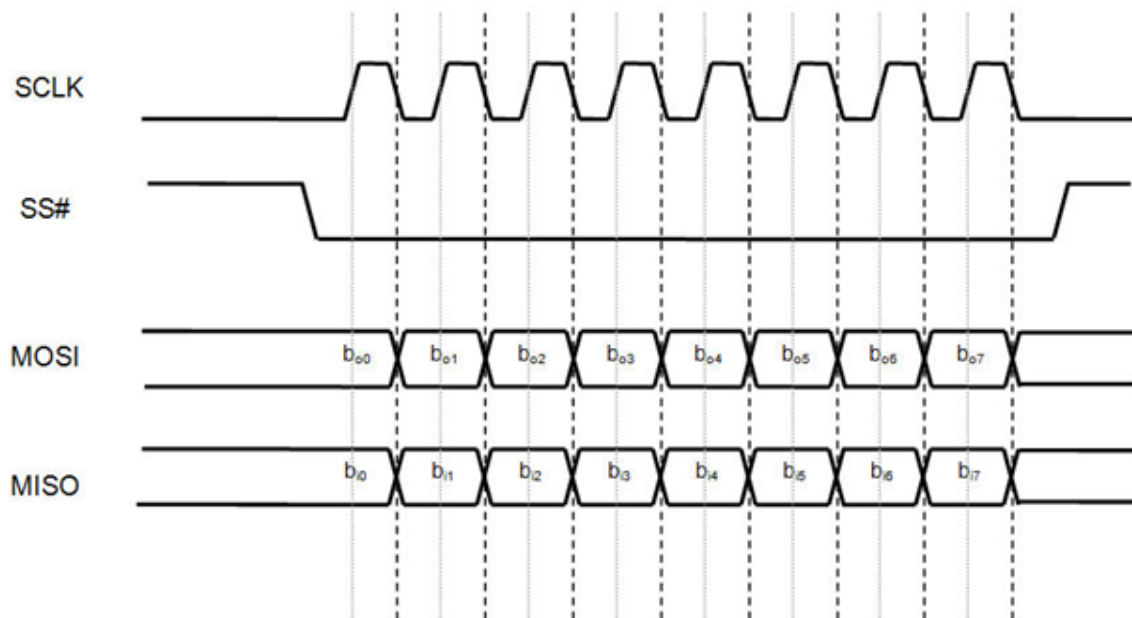


Figura 5. Flujo de datos de SPI

4.1.8.1.6. I2C

I2C significa circuito inter integrado. I2C requiere solo dos cables que conectan todos los periféricos al microcontrolador, las líneas SDA (datos en serie) y SCL (reloj en serie) sirven para llevar información entre los dispositivos.

I2C es un protocolo de comunicación maestro a esclavo, cada esclavo tiene una dirección única, el dispositivo maestro envía la dirección del dispositivo esclavo de destino y el indicador de lectura / escritura. La dirección debe coincidir con cualquier dispositivo esclavo que esté encendido, los dispositivos esclavos restantes permanecen en modo desactivado. Una vez que la dirección coincide, la comunicación se realiza entre el dispositivo maestro y el dispositivo esclavo para la transmisión y recepción de los datos. El transmisor envía datos de 8 bits, el receptor responde 1 bit de acknowledge, cuando se completa la comunicación, el maestro emite la condición de parada.

El bus I2C fue desarrollado por Philips Semiconductors, su propósito original es proporcionar una manera fácil de conectar un CPU a los chips periféricos. Los dispositivos periféricos en sistemas integrados a menudo se conectan al microcontrolador como dispositivos mapeados en memoria. I2C requiere solo dos cables para conectar todos los periféricos al microcontrolador. Estos cables activos, llamados SDA y SCL, son bidireccionales. La línea SDA es una línea de datos serie y la línea SCL es una línea de reloj serie.

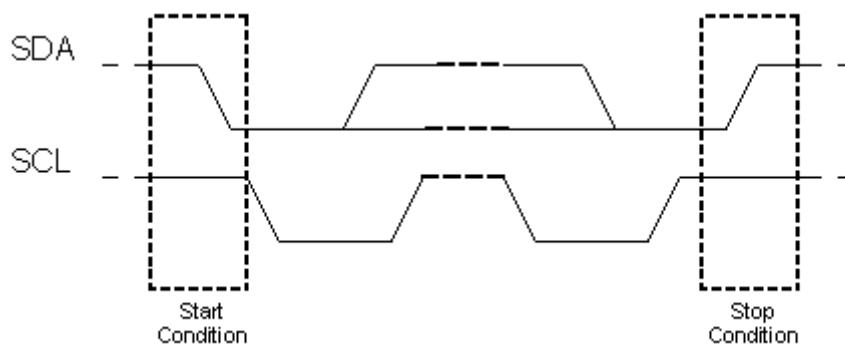


Figura 7. Flujo de datos I2C

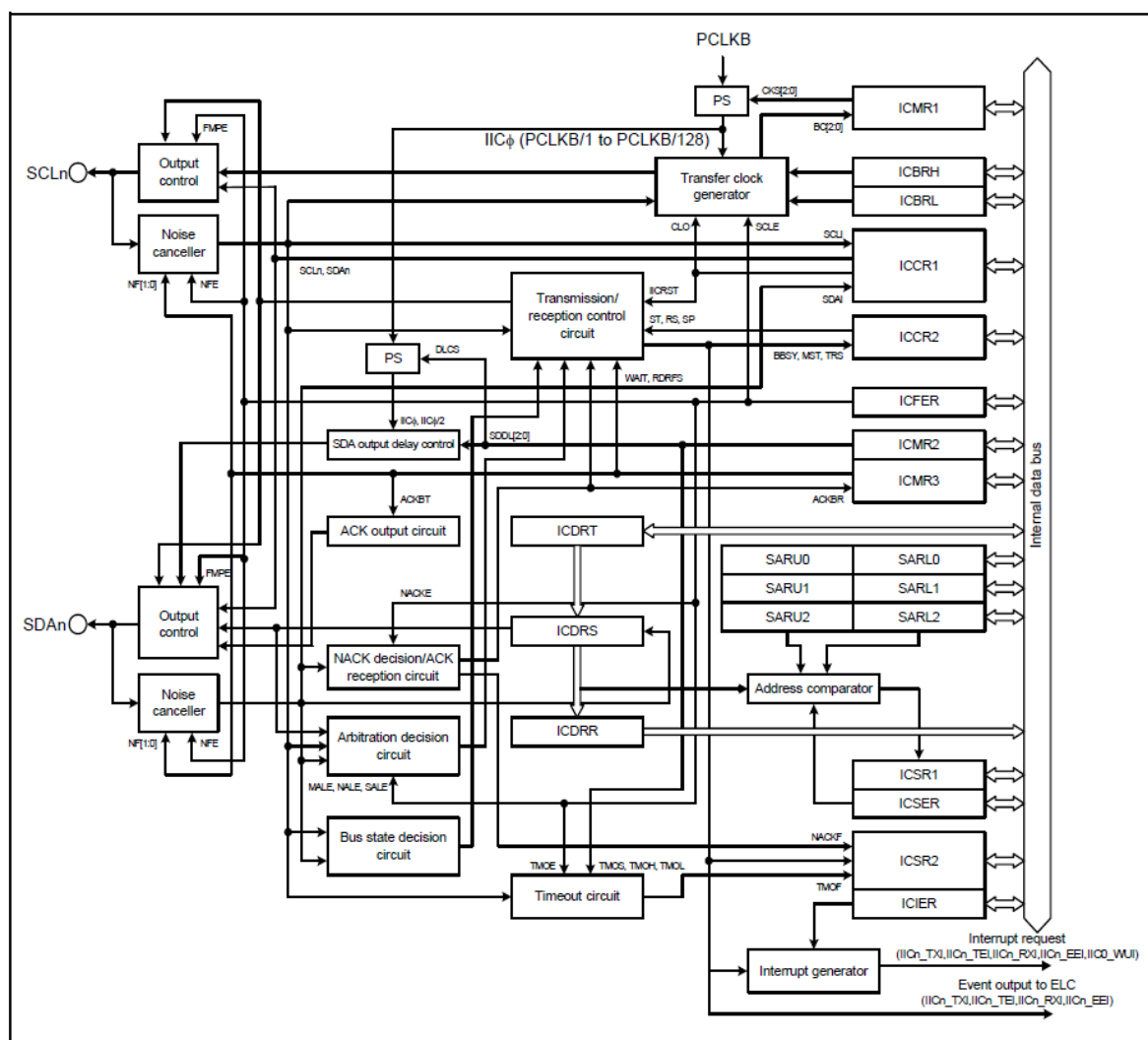


Figura 8. Diagrama de bloques I2C

4.1.8.1.7. SCI con Renesas Synergy

La configuración de un puerto SCI (UART) con Renesas Synergy se realiza bajo los siguientes pasos:

- Seleccionar la interfaz SCI que vamos a usar

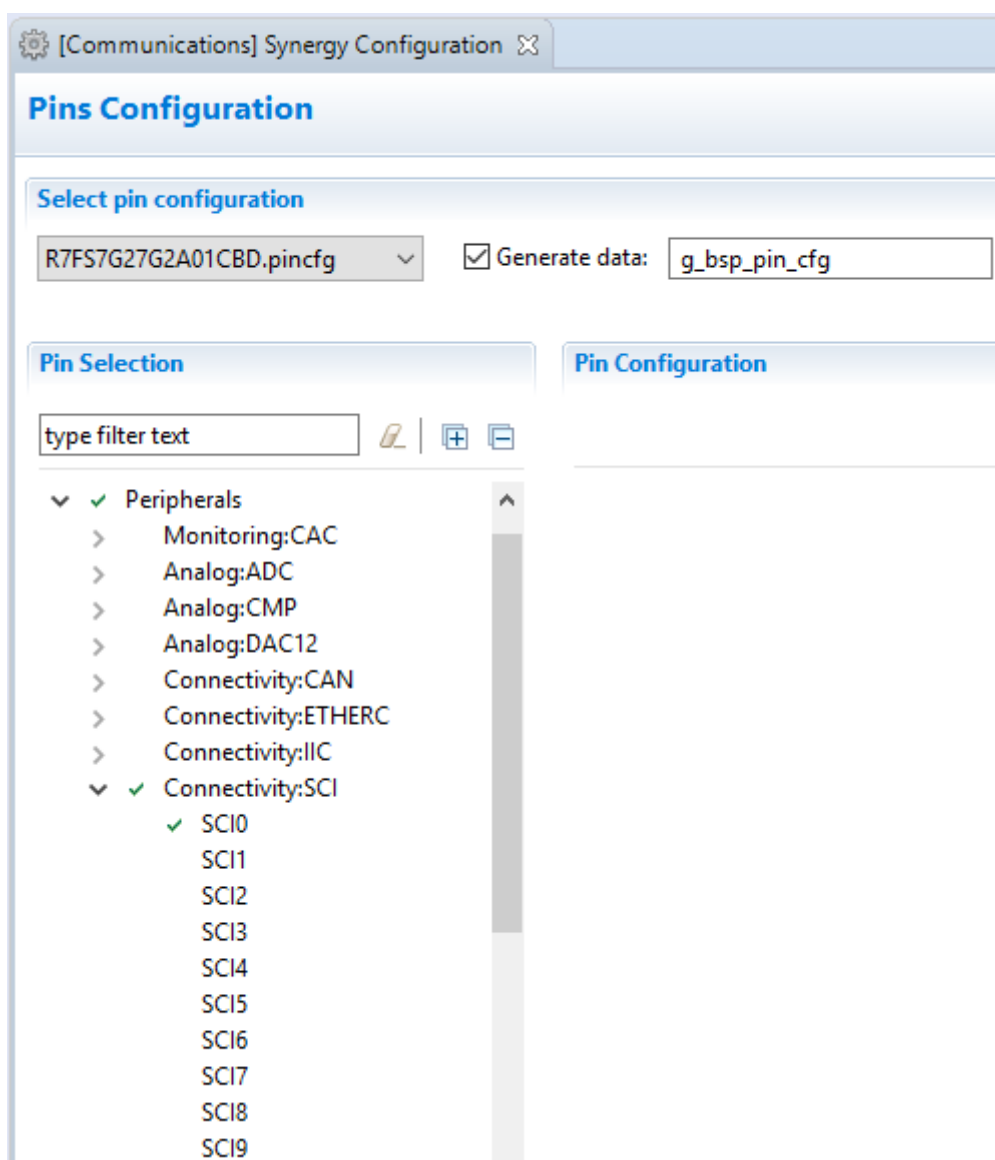


Figura 9. Selección de SCI

- Configurar el modo de operación del módulo correspondiente:

Pin Configuration

Module name:
SCI0
Usage:
When using Simple I2C mode, ensure port pins output type is n-ch open drain.
When switching between I2C and other modes, first disable.

Pin Group Selection:
_A only

Operation Mode:
Asynchronous UART

Input/Output

TXD_MOSI:	✓	P101		→
RXD_MISO:	✓	P100		→
SCK:		None		→
CTS_RTS_SS:		None		→
SDA:		None		→
SCL:		None		→

Figura 10. Modo de operación SCI

- **Pin Group Selection:**
 - Mixed
 - _A only
 - _B only
- **Operation Mode:**
 - Disabled
 - Custom
 - Asynchronous UART
 - Simple SPI

- Simple I2C
 - Synchronous UART
 - Smart Card
- Configurar el modo de operación de pin de transmisión TX:

Pin Configuration

Module name:

Symbolic Name:

Comment:

Port Capabilities:

P101

P_TX

AGT0: AGTEE
BUS0: D1_DQ1
GLCDC0: LCD_CLK
IIC1: SDA
IRQ0: IRQ01
KINT0: KRM1
POEG1: GTETRG
SCI0: SDA
SCI0: TXD_MOSI
SCI1: CTS_RTS_SS
SPI0: MOSI

P101 Configuration

Mode:

Pull up:

IRQ:

Drive Capacity:

Output type:

Peripheral mode

None

None

Low

CMOS

Chip input/output

P101:

✓

SCI0_TXD_MOSI

➔

Figura 11. Configuración pin TX de SCI

- **Symbolic Name:**
 - P_TX

- **Mode:**
 - Peripheral mode
- **Pull up:**
 - None
 - input pull-up
- **IRQ:**
 - IRQ1
- **Drive Capacity:**
 - Low
 - Medium
 - High
- **Output type:**
 - CMOS
 - n-ch open drain

Pin Configuration

Module name:

P100

Symbolic Name:

P_RX

Comment:

Port Capabilities:

AGT0: AGTIO
BUS0: D0_DQ0
GLCDC0: LCD_EXTCLK
IIC1: SCL
IRQ0: IRQ02
KINT0: KRM0
POEG0: GTETRQ
SCI0: RXD_MISO
SCI0: SCL
SCI1: SCK
SPI0: MISO

P100 Configuration

Mode:

Peripheral mode

Pull up:

None

IRQ:

IRQ2

Drive Capacity:

Low

Output type:

CMOS

Chip input/output

P100:

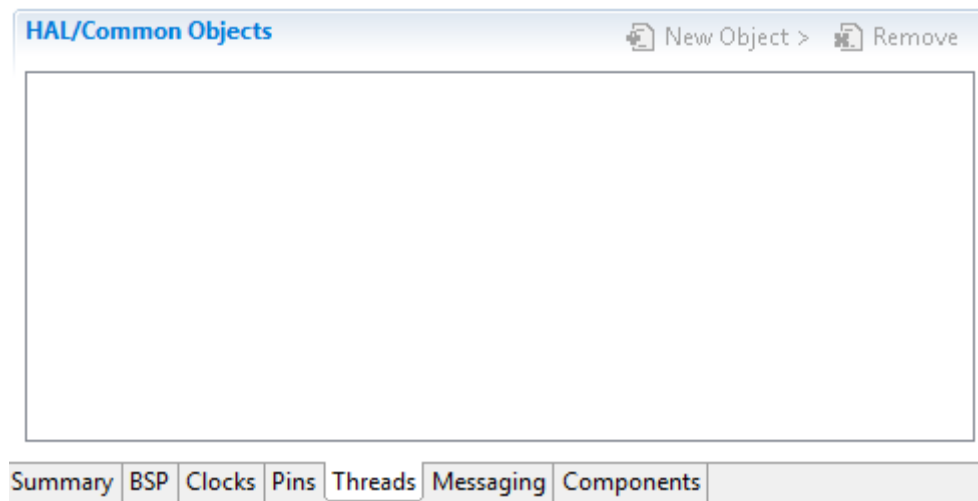
✓

SCI0_RXD_MISO

Figura 12. Configuración de PIN RX de SCI

- **Symbolic Name:**
 - P_RX
- **Mode:**
 - Peripheral mode
- **Pull up:**

- None
- input pull-up
- **IRQ:**
 - None
 - IRQ2
- **Drive Capacity:**
 - Low
 - Medium
 - High
- **Output type:**
 - CMOS
 - n-ch open drain
- En la pestaña de Threads seleccionar el stack driver de UART



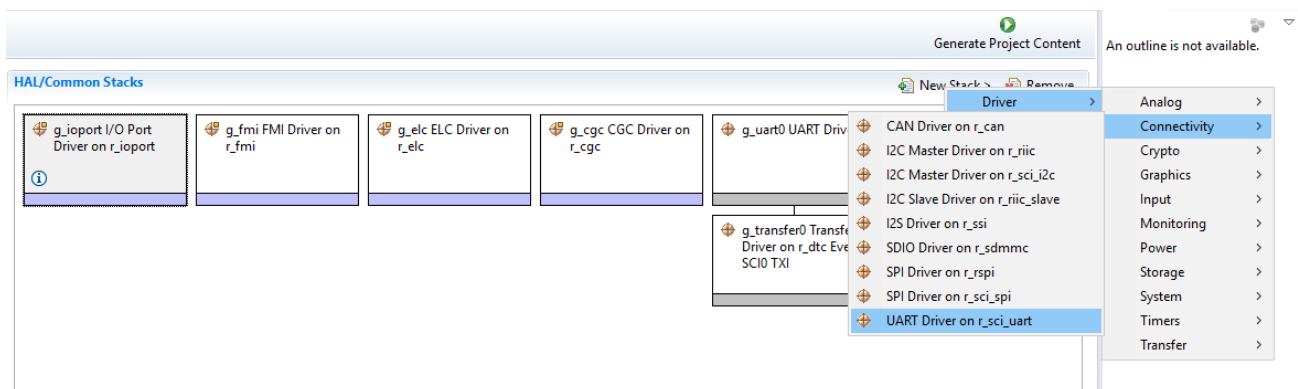


Figura 13. Agregar driver de SCI

- Por último, una vez configurado el UART se debe generar el código de las estructuras o API mediante el ícono Generate Project Content:

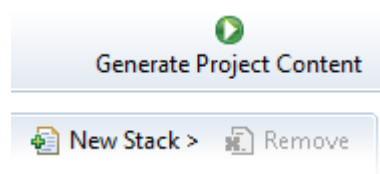


Figura 14. Generar código del driver de SCI

Las funciones principales de la estructura `p_cfg` se muestran a continuación:

- **baud:** Configura el baud rate con el cual vamos a transmitir y recibir datos.
- **channel:** Configura el canal de UART que vamos a usar.
- **data_bits:** Establece el número de bits a enviar / recibir.
- **parity:** Establece el tipo de paridad a usar.
- **rx_ipl:** Habilita las interrupciones de recepción.
- **stop_bits:** Establece el número de bits de parada.

- **tei_ipl:** Habilita las interrupciones de transmisión.

Las funciones principales de la estructura `p_api` se muestran a continuación:

- **baudSet:** Regresa el valor del baud rate configurado.
- **close:** Cierra la comunicación UART.
- **open:** Abre el puerto de comunicación UART.
- **read:** Lee los datos recibidos en el UART.
- **write:** Envía datos a través del UART.

.

4.1.8.1.8. SPI con Renesas Synergy

La configuración de un puerto SPI con Renesas Synergy se realiza bajo los siguientes pasos:

- Seleccionar la interfaz SPI que vamos a usar

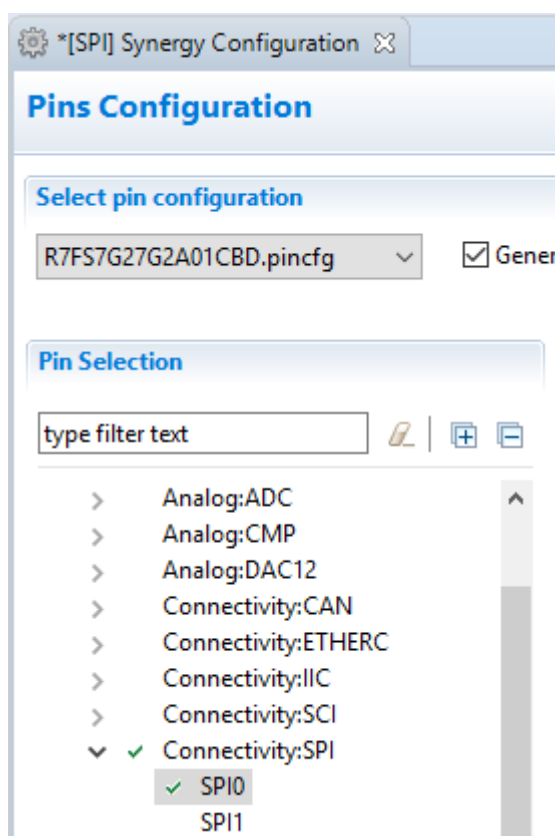


Figura 15. Selección de SPI

- Configurar el modo de operación del módulo correspondiente:

Pin Configuration

Module name:	SPI0		
Usage:	For SPI, same Pin Group recommended		
Pin Group Selection:	_A only		
Operation Mode:	Enabled		

Input/Output

MISO:	✓	P100		→
MOSI:	✓	P101		→
RSPCK:	✓	P102		→
SSL0:		None		→
SSL1:		None		→
SSL2:		None		→
SSL3:		None		→

Figura 16. Configuración de modo de operación de SPI

- **Pin Group Selection:**

- Mixed
- _A only
- _B only

- **Operation Mode:**

- Disabled
- Custom
- Enabled

- Configurar el modo de operación de los pines del módulo SPI

Pin Configuration

Module name: P100

Symbolic Name: P_MISO

Comment:

Port Capabilities:

AGT0: AGTIO
 BUS0: D0_DQ0
 GLCDC0: LCD_EXTCLK
 IIC1: SCL
 IRQ0: IRQ02
 KINT0: KRM0
 POEG0: GTETRQ
 SCI0: RXD_MISO
 SCI0: SCL
 SCI1: SCK
 SPI0: MISO

P100 Configuration

Mode: Peripheral mode

Pull up: None

IRQ: None

Drive Capacity: Low

Output type: CMOS

Chip input/output

P100: ✓ SPI0_MISO

Figura 17. Modo de operación del pin MISO

- **Symbolic Name:**
 - P_MISO
- **Mode:**
 - Peripheral mode
- **Pull up:**

- None
- input pull-up
- **IRQ:**
 - None
 - IRQ2
- **Drive Capacity:**
 - Low
 - Medium
 - High
- **Output type:**
 - CMOS
 - n-ch open drain

Pin Configuration

Module name:

P101

Symbolic Name:

P_MOSI

Comment:

Port Capabilities:

AGT0: AGTEE
BUS0: D1_DQ1
GLCDC0: LCD_CLK
IIC1: SDA
IRQ0: IRQ01
KINT0: KRM1
POEG1: GTETRG
SCI0: SDA
SCI0: TXD_MOSI
SCI1: CTS_RTS_SS
SPI0: MOSI

P101 Configuration

Mode:

Peripheral mode

Pull up:

None

IRQ:

None

Drive Capacity:

Low

Output type:

CMOS

Chip input/output

P101:

✓

SPI0_MOSI

Figura 18. Modo de operación del pin MOSI

- **Symbolic Name:**
 - P_MOSI
- **Mode:**
 - Peripheral mode
- **Pull up:**

- None
- input pull-up
- **IRQ:**
 - None
 - IRQ1
- **Drive Capacity:**
 - Low
 - Medium
 - High
- **Output type:**
 - CMOS
 - n-ch open drain

Pin Configuration

Module name: P102
Symbolic Name: P_SCK
Comment:
Port Capabilities:

ADC0: ADTRG
AGT0: AGTO
BUS0: D2_DQ2
GLCDC0: LCD_TCON0
GPT2: GTIOCB
KINT0: KRM2
OP50: GTOWLO
SCI0: SCK
SPI0: RSPCK

P102 Configuration

Mode: Peripheral mode
Pull up: None
Drive Capacity: Low
Output type: CMOS

Chip input/output

P102: ☒ SPI0_RSPCK

Figura 19. Modo de operación del pin SCK

- **Symbolic Name:**
 - P_SCK
- **Mode:**
 - Peripheral mode
- **Pull up:**
 - None
 - input pull-up

- **Drive Capacity:**
 - Low
 - Medium
 - High
- **Output type:**
 - CMOS
 - n-ch open drain

Pin Configuration

Module name:

P103

Symbolic Name:

Comment:

Port Capabilities:

BUS0: D3_DQ3
GLCDC0: LCD_TCON1
GPT2: GTIOCA
KINT0: KRM3
OPS0: GTOWUP
SCI0: CTS_RTS_SS
SPI0: SSL0

P103 Configuration

Mode:

Peripheral mode

Pull up:

None

Drive Capacity:

Low

Output type:

CMOS

Chip input/output

P103:

✓

SPI0_SSL0

➔

Figura 20. Modo de operación del pin SS

- **Symbolic Name:**

- P_SS
 - **Mode:**
 - Peripheral mode
 - **Pull up:**
 - None
 - input pull-up
 - **Drive Capacity:**
 - Low
 - Medium
 - High
 - **Output type:**
 - CMOS
 - n-ch open drain
- En la pestaña de Threads seleccionar el stack driver de SPI

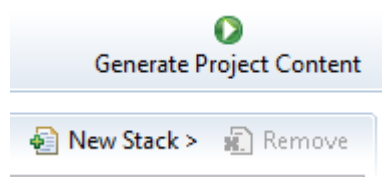


Figura 22. Generar código del driver de SPI

Las funciones principales de la estructura `p_cfg` se muestran a continuación:

- **bit_order**: Configura el orden de envío de los bits msb o lsb
- **bitrate**: Configura la velocidad de transmisión.
- **clk_phase**: Configura la fase del reloj.
- **clk_polarity**: Configura la polaridad del reloj.
- **operating_mode**: Modo de operación.
- **p_transfer_rx**: Habilita recepción de datos.
- **p_transfer_tx**: Habilita transmisión de datos.
- **rx_ipl**: Habilita interrupción de recepción de datos.
- **tx_ipl**: Habilita interrupción de transmisión de datos.

Las funciones principales de la estructura `p_api` se muestran a continuación:

- **close**: Cierra la comunicación SPI.
- **open**: Abre el puerto de comunicación SPI.
- **read**: Lee los datos recibidos en el puerto SPI.
- **write**: Envía datos a través del puerto SPI.
- **writeRead**: Envía y recibe datos a través del puerto SPI.

4.1.8.1.9. I2C con Renesas Synergy

La configuración de un puerto I2C con Renesas Synergy se realiza bajo los siguientes pasos:

- Seleccionar la interfaz I2C que vamos a usar

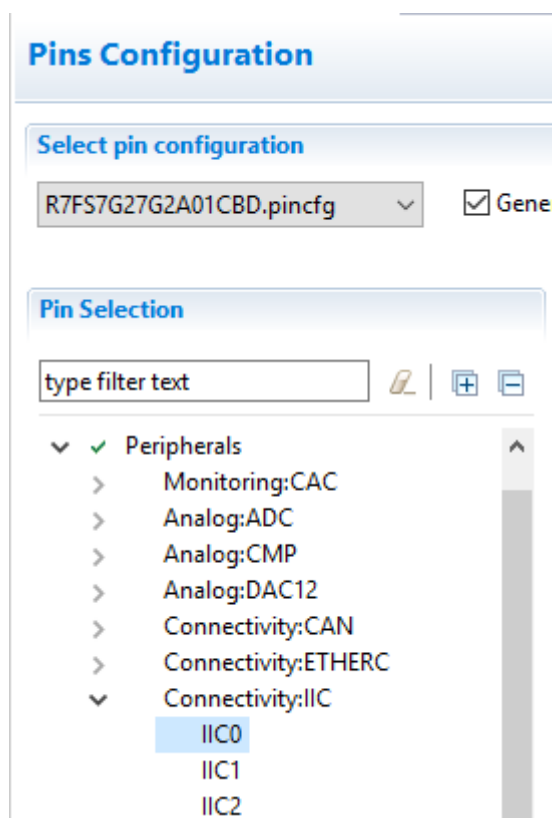


Figura 23. Selección de interfaz I2C

- Configurar el modo de operación del módulo correspondiente:

Pin Configuration

Module name:	IIC0		
Usage:	For IIC, use same Pin Group for SDA/SCL signals -Please refer to the MCU User's Manual.		
Pin Group Selection:	_A only		
Operation Mode:	Enabled		

Input/Output

SDA:	✓	P401		➡
SCL:	✓	P400		➡

Figura 24. Modo de operación del I2C

- **Pin Group Selection:**

- _A only
- _B only

- **Operation Mode:**

- Disabled
- Enabled

- Configurar el modo de operación de los pines del módulo I2C

Pin Configuration

Module name: P401

Symbolic Name: P_SDA

Comment:

Port Capabilities:

CAN0: CTX
 ETHERC0: MDC
 ETHERC0: MDC
 GPT6: GTIOCB
 IIC0: SDA
 IRQ0: IRQ05
 POEG0: GTETRQ
 SCI4: CTS_RTS_SS
 SCI7: SDA
 SCI7: TXD_MOSI

P401 Configuration

Mode: Peripheral mode

Pull up: None

IRQ: None

Output type: n-ch open drain

Chip input/output

P401: ✓ IIC0_SDA

Figura 25. Modo de operación del pin SDA

- **Symbolic Name:**
 - P_SDA
- **Mode:**
 - Peripheral mode
- **Pull up:**
 - None

- input pull-up
- **IRQ:**
 - None
 - IRQ5-DS
- **Output type:**
 - CMOS
 - n-ch open drain

Pin Configuration

Module name:

P400

Symbolic Name:

P_SCL

Comment:

Port Capabilities:

ADC1: ADTRG
ETHERC1: TX_CLK
GPT6: GTIOCA
IIC0: SCL
IRQ0: IRQ00
SCI4: SCK
SCI7: SCK
SSI: AUDIO_CLK

P400 Configuration

Mode:

Peripheral mode

Pull up:

None

IRQ:

None

Output type:

n-ch open drain

Chip input/output

P400:

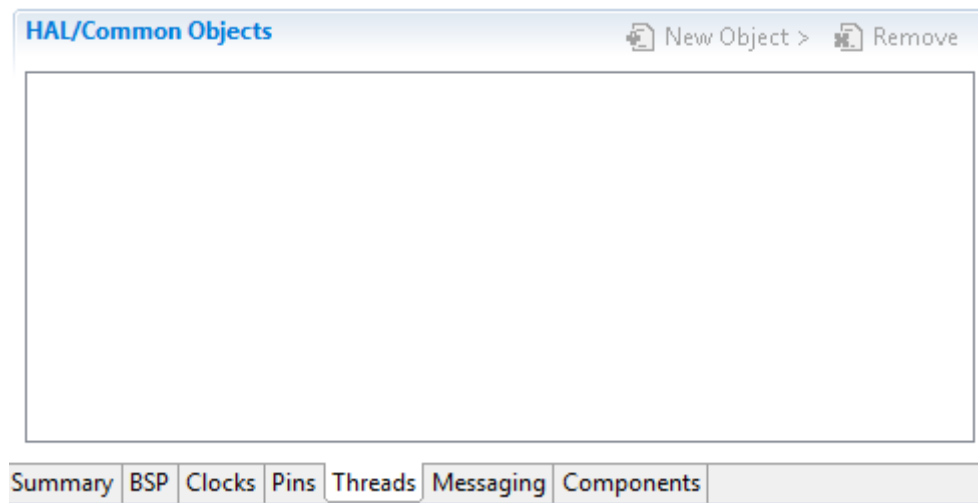
✓

IIC0_SCL

Figura 26. Modo de operación del pin SCL

- **Symbolic Name:**
 - P_SCL
- **Mode:**
 - Peripheral mode
- **Pull up:**
 - None
 - input pull-up

- **IRQ:**
 - None
 - IRQ0
 - **Output type:**
 - CMOS
 - n-ch open drain
- En la pestaña de Threads seleccionar el stack driver de I2C



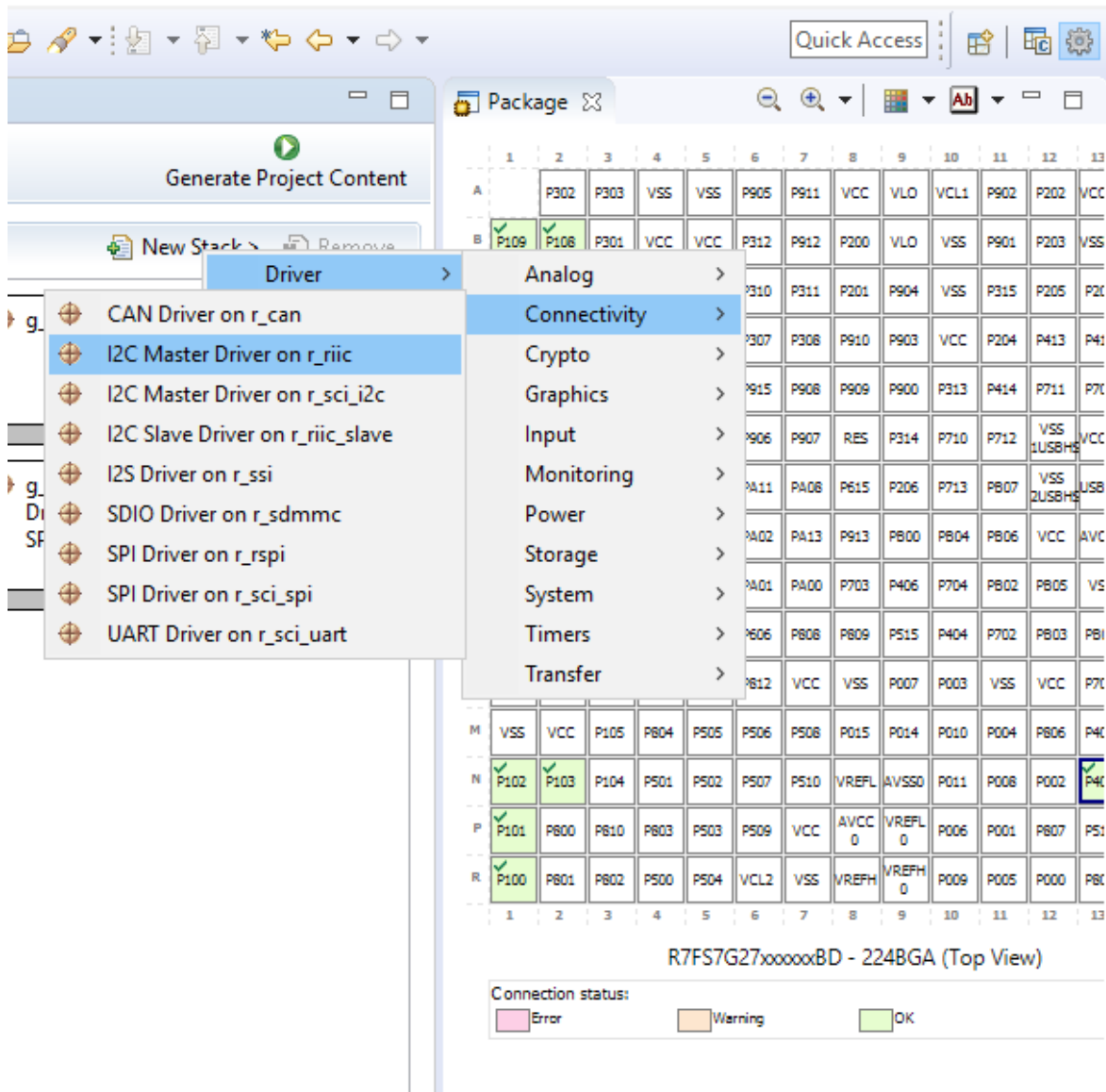


Figura 27. Agregar driver de I2C

- Por último, una vez configurado el módulo I2C se debe generar el código de las estructuras o API mediante el ícono Generate Project Content:

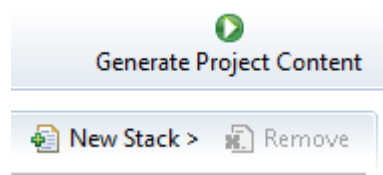


Figura 28. Generar código de I2C

Las funciones principales de la estructura `p_cfg` se muestran a continuación:

- **rate:** Configura la velocidad de transmisión / recepción de datos.
- **slave:** Configura la dirección en modo esclavo.
- **rx_ipl:** Configura la interrupción de recepción.
- **tx_ipl:** Configura la interrupción de transmisión.
- **rx_ipl:** Habilita las interrupciones de recepción.

Las funciones principales de la estructura `p_api` se muestran a continuación:

- **close:** Cierra la comunicación I2C.
- **open:** Abre el puerto de comunicación I2C.
- **read:** Lee los datos recibidos en el I2C.
- **write:** Envía datos a través del I2C.
- **reset:** Reinicia el módulo I2C.
- **slaveAddressSet:** Asigna la dirección en modo esclavo.

4.1.8.1.10. Consideraciones de diseño

Para el uso de comunicación a través de UART se recomienda de acuerdo a la longitud entre los módulos que se están comunicando, el uso de cable blindado o par trenzado para evitar la interferencia electromagnética.

Otra buena práctica es tener un método de validación de integridad de los datos recibidos, para ello podemos utilizar cualquiera de las siguientes alternativas.

- CheckSum
- CRC16
- CRC32

Para el caso de comunicación SPI e I2C se recomienda que la distancia entre el microcontrolador y el periférico no sea muy grande, sabemos que éstos normalmente están dentro de la misma tarjeta electrónica.

El Checksum es posiblemente el método más antiguo usado para la detección de errores y está basado en la suma de todos los bytes transmitidos, el checksum se agrega normalmente al final de la trama a enviar.

El receptor tiene que hacer el cálculo nuevamente y así comprobar la integridad de los datos.

El CRC (Cyclic Redundancy Check) es también un concepto empleado para la validación de datos, el principio es similar al Checksum, pero en lugar de hacer sumas, utiliza división de polinomios para la determinación del CRC, el cual es generalmente de 16 o 32 bits de longitud.


```

unsigned short update_crc(unsigned short crc_accum, unsigned char *data_blk_ptr, unsigned short data_blk_size)
{
    unsigned short i, j;
    unsigned short crc_table[256] = {
        0x0000, 0x8005, 0x800F, 0x000A, 0x801B, 0x001E, 0x0014, 0x8011,
        0x8033, 0x0036, 0x003C, 0x8039, 0x0028, 0x802D, 0x8027, 0x0022,
        0x8063, 0x0066, 0x006C, 0x8069, 0x0078, 0x807D, 0x8077, 0x0072,
        0x0050, 0x8055, 0x805F, 0x005A, 0x804B, 0x004E, 0x0044, 0x8041,
        0x80C3, 0x00C6, 0x00CC, 0x80C9, 0x00D8, 0x80DD, 0x80D7, 0x00D2,
        0x00F0, 0x80F5, 0x80FF, 0x00FA, 0x80EB, 0x00EE, 0x00E4, 0x80E1,
        0x00A0, 0x80A5, 0x80AF, 0x00AA, 0x80BB, 0x00BE, 0x00B4, 0x80B1,
        0x8093, 0x0096, 0x009C, 0x8099, 0x0088, 0x808D, 0x8087, 0x0082,
        0x8183, 0x0186, 0x018C, 0x8189, 0x0198, 0x819D, 0x8197, 0x0192,
        0x01B0, 0x81B5, 0x81BF, 0x01BA, 0x81AB, 0x01AE, 0x01A4, 0x81A1,
        0x01E0, 0x81E5, 0x81EF, 0x01EA, 0x81FB, 0x01FE, 0x01F4, 0x81F1,
        0x81D3, 0x01D6, 0x01DC, 0x81D9, 0x01C8, 0x81CD, 0x81C7, 0x01C2,
        0x0140, 0x8145, 0x814F, 0x014A, 0x815B, 0x015E, 0x0154, 0x8151,
        0x8173, 0x0176, 0x017C, 0x8179, 0x0168, 0x816D, 0x8167, 0x0162,
        0x8123, 0x0126, 0x012C, 0x8129, 0x0138, 0x813D, 0x8137, 0x0132,
        0x0110, 0x8115, 0x811F, 0x011A, 0x810B, 0x010E, 0x0104, 0x8101,
        0x8303, 0x0306, 0x030C, 0x8309, 0x0318, 0x831D, 0x8317, 0x0312,
        0x0330, 0x8335, 0x833F, 0x033A, 0x832B, 0x032E, 0x0324, 0x8321,
        0x0360, 0x8365, 0x836F, 0x036A, 0x837B, 0x037E, 0x0374, 0x8371,
        0x8353, 0x0356, 0x035C, 0x8359, 0x0348, 0x834D, 0x8347, 0x0342,
        0x03C0, 0x83C5, 0x83CF, 0x03CA, 0x83DB, 0x03DE, 0x03D4, 0x83D1,
        0x83F3, 0x03F6, 0x03FC, 0x83F9, 0x03E8, 0x83ED, 0x83E7, 0x03E2,
        0x83A3, 0x03A6, 0x03AC, 0x83A9, 0x03B8, 0x83BD, 0x83B7, 0x03B2,
        0x0390, 0x8395, 0x839F, 0x039A, 0x838B, 0x038E, 0x0384, 0x8381,
        0x0280, 0x8285, 0x828F, 0x028A, 0x829B, 0x029E, 0x0294, 0x8291,
        0x82B3, 0x02B6, 0x02BC, 0x82B9, 0x02A8, 0x82AD, 0x82A7, 0x02A2,
        0x82E3, 0x02E6, 0x02EC, 0x82E9, 0x02F8, 0x82FD, 0x82F7, 0x02F2,
        0x02D0, 0x82D5, 0x82DF, 0x02DA, 0x82CB, 0x02CE, 0x02C4, 0x82C1,
        0x8243, 0x0246, 0x024C, 0x8249, 0x0258, 0x825D, 0x8257, 0x0252,
        0x0270, 0x8275, 0x827F, 0x027A, 0x826B, 0x026E, 0x0264, 0x8261,
        0x0220, 0x8225, 0x822F, 0x022A, 0x823B, 0x023E, 0x0234, 0x8231,
        0x8213, 0x0216, 0x021C, 0x8219, 0x0208, 0x820D, 0x8207, 0x0202
    };
};

for(j = 0; j < data_blk_size; j++)
{
    i = ((unsigned short)(crc_accum >> 8) ^ data_blk_ptr[j]) & 0xFF;
    crc_accum = (crc_accum << 8) ^ crc_table[i];
}

return crc_accum;
}

```

Figura 29. Función para cálculo de CRC

<http://ww1.microchip.com/downloads/en/appnotes/00730a.pdf>

4.1.8.1.11. Ejercicios

Ejercicio 1		Comunicación Tarjeta Synergy S7G2 - PC			
Duración	120 min.	Participantes:	Equipo	Lugar:	Aula
Descripción: Configure la tarjeta de evaluación S7G2 para leer un potenciómetro a través de un ADC y enviar el valor leído por el puerto SCI3 para ser desplegado en una PC.					
Materiales: <ul style="list-style-type: none">• Tarjeta Synergy S7G2• Laptop o PC• Convertidor TTL a USB• Potenciómetro 1K o 10K• Cables Dupont• Software de uso libre (Tera Term, Hyperterminal o similar) para captura de datos seriales.					

Ejercicio 2		Comunicación entre tarjetas Synergy S7G2			
Duración	120 min.	Participantes:	Equipo	Lugar:	Aula
Descripción: Implemente un protocolo de comunicación maestro-esclavo entre dos tarjetas de evaluación S7G2 usando el puerto SCI3, el esclavo leerá los botones y el maestro encenderá leds correspondientes a los botones presionados en el esclavo.					
Materiales: <ul style="list-style-type: none"> • Dos tarjetas Synergy S7G2 • Cables Dupont 					

4.1.8.1.12. Referencias

Renesas Electronics Corporation. (Aug 24, 2017).

<https://www.renesas.com/eu/en/doc/products/renesas-synergy/apn/r11an0111eu0101-synergy-spi-hal-mod-guide.pdf>

Renesas Electronics Corporation. (Aug 30, 2017).

<https://www.renesas.com/eu/en/doc/products/renesas-synergy/apn/r11an0117eu0101-synergy-spi-fw-mod-guide.pdf>

Renesas Electronics Corporation. (Sep 5, 2017).

<https://www.renesas.com/eu/en/doc/products/renesas-synergy/apn/r11an0148eu0101-synergy-i2c-master-hal-mod-guide.pdf>

Renesas Electronics Corporation. (31 de August de 2018). Synergy Software Package User's Manual.

Overview on Electronic Communication Protocols. (2013-2018).

<https://www.elprocus.com/communication-protocols/>

4.1.8.1.13. Evaluación

Proyecto		Comunicación I2C y SPI con tarjeta Synergy S7G2			
Duración		Participantes:	2 personas	Lugar:	
Descripción: Crear un software para leer las coordenadas del Touch-screen sensing (Semtech SX8656) y mostrar la posición de la coordenada con una “X” en el LCD (HaoRan HT024K5QV50T con driver Ilitek ILI9341V).					
Materiales: <ul style="list-style-type: none">• Tarjeta Synergy S7G2					