# Reporte rurinas para comunicación SPI entre RPI y XMOS

Felipe M.N.

28 de Abril, 2017.

## 1 Objectivo

Desarrollar rutinas de simple uso para comunicar desde un xmosStartKIT(XMOS) a una Raspberry V2.0 B (Rpi), para poder enviar y recibir desde ambos dispositivos a través del protocolo SPI.

## 2 Idea principal

Dada la característica de la libreria WiringPi para manejar los GPIO y los distintos protocolos de comunicación con los que cuenta la Rpi, no se tienen disponibles las rutinas para comunicarse por SPI con la Rpi trabajando como esclavo, solo como maestro (Recordemos que en SPI el maestro es el que inicia la comunicación). Por lo tanto, se amarró un GPIO del XMOS con un GPIO de la Rpi para generar una interrupción que le avise al maestro que debe pedirle datos al esclavo.

De manera inversa, cuando se quiere enviar información desde la Rpi a la XMOS, se deben enviar los datos de forma ordenada y recibirlos en un vector.

# 3 Topología

La figura 2 ejemplifica el conexionado necesario entre las placas para poder comunicarse, recordemos que la asignación de GPIO de la XMOS es aleatoria, pues puede ser cualquier puerto de tamaño 1, no así en la Rpi, que cada GPIO es específico (al menos en el bus SPI).

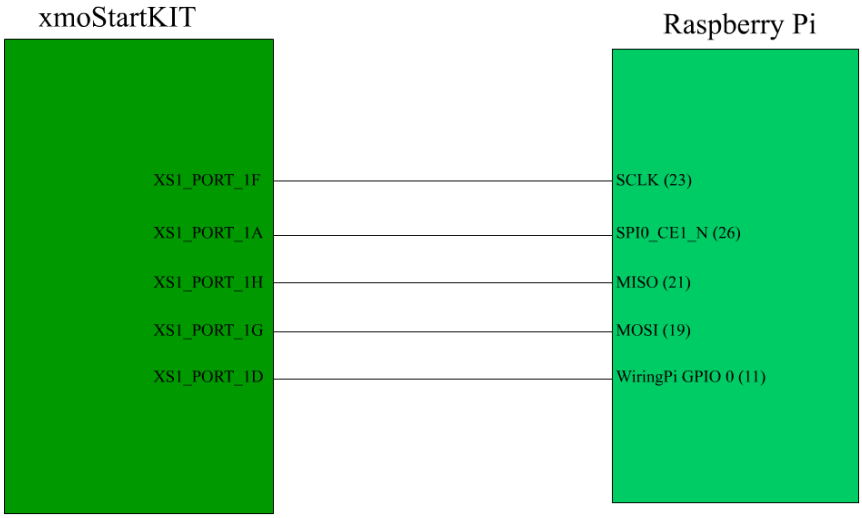


Figure 1: Conexiones.

# 4 Implementación

La forma en la que se implementaron las rutinas para enviar desde la XMOS a la Rpi de forma ordenada fue la siguiente:

- Primero se interrumpe la Rpi desde la XMOS.
- Luego se envía un byte con el tamaño de los datos.
- Finalmente se envían los datos con el tamaño ya recibido.

Esta forma se debe a que las rutinas de la libreria WiringPi trabajan con la cantidad de bytes enviados, no así la librería de la XMOS. Con respecto al

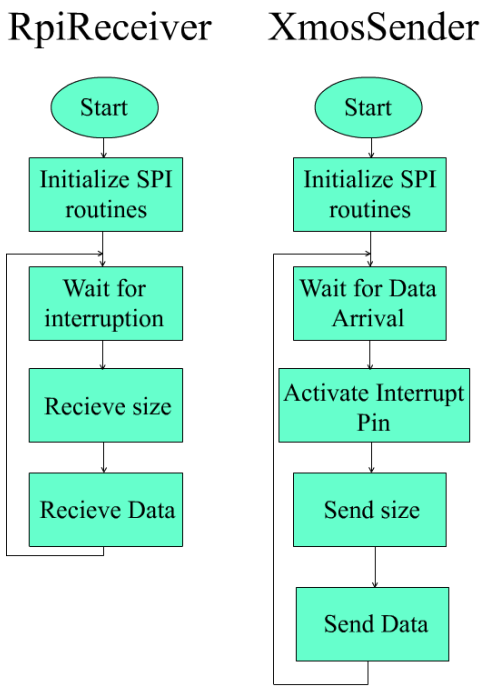


Figure 2: Diagrama de flujo, XMOS sender, Rpi receiver.

envío desde la Rpi, es mas transparente, pues es el maestro el que inicia la comunicación y se modificó el ejemplo de [3], para recibir datos en la XMOS por SPI como esclavo.

# 5 Repositorio

El código de las rutinas se encuentra en https://github.com/gopimn/xmosRpiSenderReceiver bajo licencia GPL 3.0, y se compone de dos directorios:

## 5.1 src

En esta capeta se encuentran los *headers* para poder usar las diferentes rutinas:

- *rpiReceiver.h*,para recibir con la Rpi, contiene la función rpiReceiver, que recibe como argumento el vector de *unsigned char* donde guardaremos la data y retorna la cantidad de bytes recibidos.
- *rpiSender.h*, para enviar con la Rpi, contiene la Task rpiSend, que recibe como argumento el vector de *unsigned char* donde tenemos la información a enviar y la cantidad de bytes que se quiere enviar, retorna la cantidad de bytes exitosamente enviados.
- *xmosReceiver*, para recibir con la XMOS, contiene la función spiSender, que tiene como argumentos la interfaz SPI(esclavo) y la interfaz *receiverSPI_if*para recibir datos de esta rutina definida en el mismo header, permite crear una Task para llamar a la función xmosRecieve, que ejecuta la adquisición de datos como cliente (revisar documentación [], para más información acerca de los conceptos de interfaz y channend de lenguaje xc).
- *xmosSender.h*, para enviar con la XMOS, contiene la Task spiSender, que tiene como argumentos la interfaz de comunicacion SPI (esclavo), la interfaz *senderSPI_if* y el puerto (GPIO) con el cuál interrumpimos a la Rpi. Esta interfaz permite usar como cliente la función xmosSend, que tiene como argumentos el vector de datos y la cantidad de bytes a enviar.

## 5.2 examples

En esta carpeta se encuentran los dos ejemplos para ambas aplicaciones, enviar desde la XMOS y recibir desde la XMOS, recordar colocar en el mismo directorio del codigo fuente el header correspondiente definido en el encabezado de cada archivo en esta carpeta.

# 6 Notas Finales

- Es importante conocer los conceptos de interfaz cliente y servidor usados en [1].
- Existe una inconsistencia con la rutina de la Rpi al tratar de recibir mas de 15 Bytes (ocurre un corrimiento) dada la pobre calidad del reloj del Rpi. Para contrarrestar este efecto, se debe usar la rutina con 10 bytes o menos y llamarla varias veces en caso de tener mucha data, con intervalos de 1 [ms].
- Es importante tener instalada la librería wiring Pi en la Rpi.
- Es importante importar en el makefile de la XMOS a las librerias spi y debug_printf.
- Recordar colocar los header en la misma carpeta donde se ejecuta los codigos de ejemplo.
- En la Rpi compilar con g++ y ejecutar como superusuario.

# 7 Bibliografía

Link en el número.


- [1] XMOS programing guide.
- [2] XMOSSTARTKIT hardware manual.
- [3] XMOS spi slave documentation.
- [4] Libreria Wiring pi.