

Sistemas Ubicuos, Empotrados y Móviles

Práctica 1: Laberinto “two players”

Máster Universitario en Ingeniería Informática

Curso 2020/2021



MÁSTER UNIVERSITARIO
INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Luis Blázquez Miñambres
Francisco Pinto Santos

Contenido

Tabla de ilustraciones	4
1. Introducción	5
2. Descripción del sistema.....	6
3. Componentes.....	7
3.1. Joystick.....	7
3.1.1. Configuración Kinetis	7
3.1.2. Aspectos del código	7
3.2. Buzzer	8
3.2.1. Configuración Kinetis	8
3.2.2. Aspectos del código	8
3.3. Temporizador	8
3.3.1. Configuración Kinetis	8
3.3.2. Aspectos del código	9
3.4. RGB LED y Botones.....	9
3.4.1. Configuración Kinetis	9
3.4.2. Aspectos del código	10
3.5. Puerto Serie	10
3.5.1. Configuración Kinetis	11
3.5.2. Aspectos del código	12
4. Conclusiones	13
Bibliografía.....	14

Tabla de ilustraciones

Ilustración 1. Visualización del juego en ejecución	6
Ilustración 2: Configuración del componente ADC para los joystick en Kinetis Studio	7
Ilustración 3: Configuración del componente buzzer en Kinetis Studio	8
Ilustración 4: Configuración del componente TimerInt en Kinetis Studio.....	9
Ilustración 5: Configuración del componente BytesIO de salida del RGB en Kinetis Studio	10
Ilustración 6: Configuración del componente BytesIO de entrada del RGB en Kinetis Studio	10
Ilustración 7: Configuración del componente Terminal en Kinetis Studio	11
Ilustración 8: Configuración del componente Wait en Kinetis Studio	11
Ilustración 9: Configuración de conexión a la terminal desplegada en Putty	12

1. Introducción

En este documento se recoge el informe acerca de la práctica primera de la asignatura “Sistemas Ubicuos, Empotrados y Móviles” del Máster en Ingeniería Informática por la Universidad de Salamanca en el Curso 2020-2021, relacionado con el desarrollo de un juego en el que , mediante el uso de los componentes y conceptos desarrollados y aprendidos en la asignatura (E/S digitales y analógicas, temporizadores y puerto serie, entre otros) realizar la implementación de un laberinto mediante el cual dos jugadores puedan utilizar dos joysticks para moverse a través del mismo hasta el punto de salida, todo ello, haciendo uso de los distintos periféricos y herramientas proporcionadas mediante el material de la asignatura.

2. Descripción del sistema

En la realización de la práctica se ha desarrollado el juego atendiendo a los criterios y requisitos mínimos establecidos y a tener en cuenta en el enunciado.

En primer lugar, se ha hecho uso de la terminal de Putty como programa cliente para mostrar y visualizar el laberinto mediante el uso de comunicación del puerto serie con la placa, mediante una matriz de 15x25.

Adicionalmente, se han añadido los dos joysticks, uno para cada jugador, de forma que la salida de cada uno de ellos se realiza desde un lado del laberinto con el objetivo de llegar a la salida en la parte superior. Se ha empleado un *buzzer* o zumbador para emitir sonidos distintos para determinadas situaciones: cada vez que los jugadores se chocaban entre ellos, cada vez que se chocaban con las paredes del laberinto y cuando llegaban a la meta. Además del uso del LED RGB para notificar:

- Rojo: notificará que el juego se encuentra esperando a que uno de los jugadores pulse el botón de comienzo de la partida.
- Verde: notificará que hay una partida en curso del juego
- Amarillo: el jugador se dibujará con color amarillo y el LED RGB se pondrá de este color para notificar la victoria de uno de los jugadores.

Por otro lado, se ha hecho uso de un temporizador para tener en cuenta el tiempo transcurrido desde el inicio de la partida y, en caso de que ninguno de los jugadores haya llegado a la meta cuando este llegue a 0, ceder la victoria a aquel jugador que haya realizado más movimientos correctos.

Finalmente, tal y como se puede ver en la Ilustración 1, la disposición del laberinto en la terminal de Putty una vez ejecutado el programa quedaría de la siguiente manera.

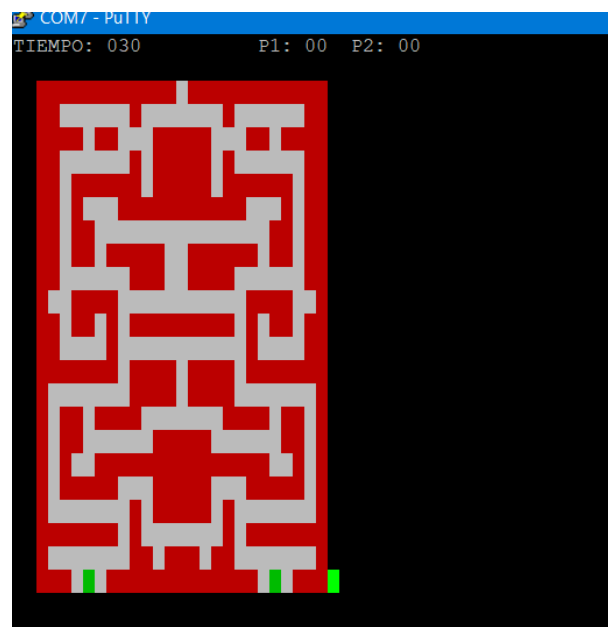


Ilustración 1. Visualización del juego en ejecución

3. Componentes

En este apartado se explicarán los componentes utilizados, así como una breve representación de su configuración y de la disposición de la lógica del código utilizada en cada uno.

3.1. Joystick

Como ejemplo de uso de entrada analógica se ha hecho uso de dos joysticks (uno para cada jugador) para emular el movimiento de cada uno de ellos dentro del laberinto.

3.1.1. Configuración Kinetis

Para configurar este componente en Kinetis Studio será necesario añadir el componente ADC al proyecto tal y como se indica en el material de las prácticas. Un vistazo a la configuración añadida se puede observar en la Ilustración 2. En cuanto a la conexión de los pines a la placa, se ha hecho uso de los pines **PTA7** y **PTA6** para la referencia a VRx y VRy para los movimientos en las direcciones del eje X y el eje Y del joystick 1 y los pines **PT** y **PT** para los movimientos en las direcciones del eje X y el eje Y del joystick 2.

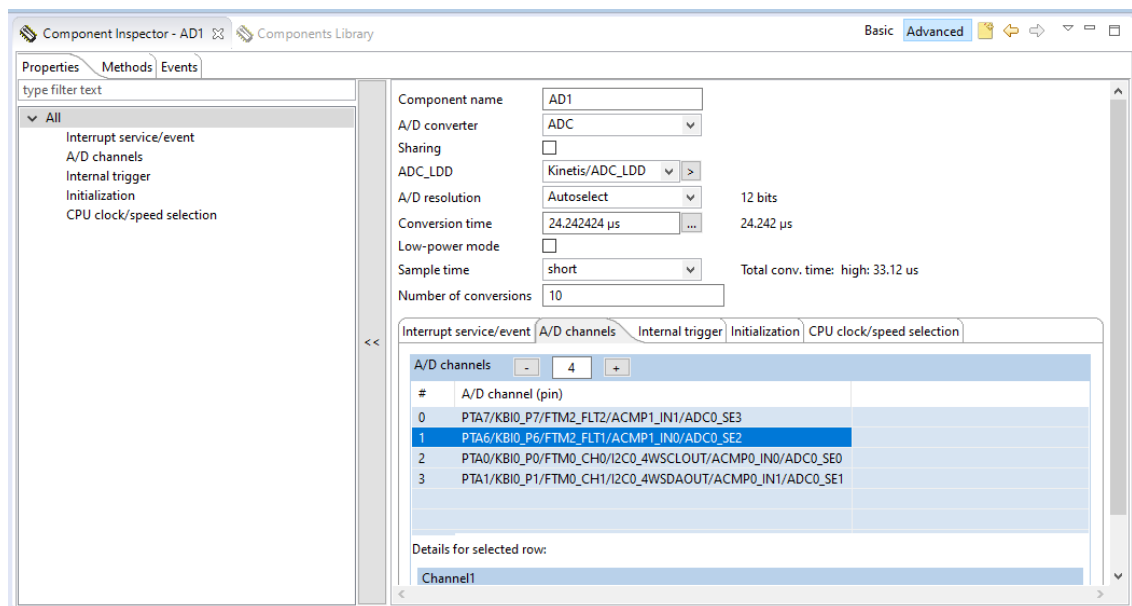


Ilustración 2: Configuración del componente ADC para los joystick en Kinetis Studio

3.1.2. Aspectos del código

En cuanto a los aspectos del código fuente, una vez configurado el componente, se hará uso de las funciones **GetValue16** para leer las medidas que proceden del joystick y transformarlas a voltaje mediante un factor de conversión proporcionado en los manuales de la asignatura. Para ello se han tomado las siguientes referencias:

- Valor entre 0 y 1.5
 - Eje X: Izquierda
 - Eje Y: Abajo
- Valor entre 1.5 y 3
 - Eje X: Derecha
 - Eje Y: Arriba

3.2. Buzzer

Como ejemplo de uso de un componente de generación de sonidos y frecuencias se ha hecho uso de un zumbador o *buzzer* para emitir sonidos cuando se dan distintas situaciones.

3.2.1. Configuración Kinetis

Para configurar este componente en Kinetis Studio será necesario añadir el componente ADC al proyecto tal y como se indica en el material de las prácticas. Un vistazo a la configuración añadida se puede observar en la Ilustración 3. En cuanto a la conexión de los pines a la placa, se ha hecho uso del pin **PTH0** para la conexión I/O del Buzzer por el puerto por defecto establecido por la placa.

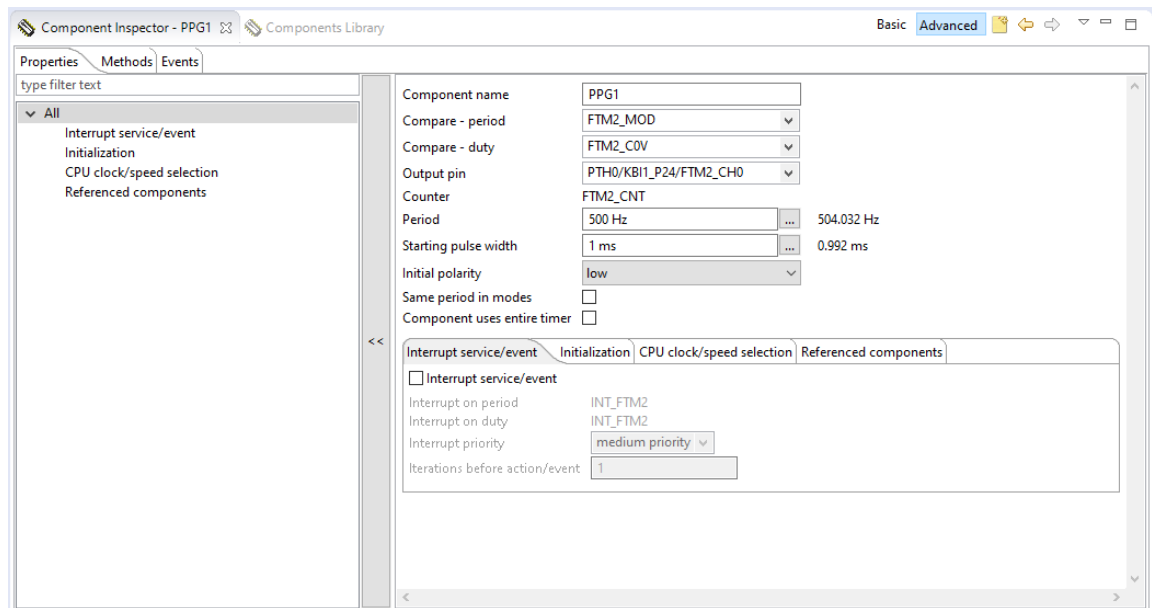


Ilustración 3: Configuración del componente buzzer en Kinetis Studio

3.2.2. Aspectos del código

En cuanto a los aspectos del código fuente, vez configurado el componente, se hará uso de las funciones **noTone()** para parar el sonido del *buzzer* y **Tone()** pasándole la frecuencia del sonido que se quiere emitir junto con la duración del mismo.

3.3. Temporizador

Como ejemplo de uso del componente de interrupciones y temporización se ha hecho uso del componente TimerInt configurado para que cada segundo se decremente una variable contadora con el tiempo transcurrido de juego.

3.3.1. Configuración Kinetis

Para configurar este componente en Kinetis Studio será necesario añadir el componente ADC al proyecto tal y como se indica en el material de las prácticas. Un vistazo a la configuración añadida se puede observar en la Ilustración 4. En la configuración del componente se ha establecido 1000 ms (1s) para que se genere la pulsación que indique al temporizador una actualización o cambio en el componente. Este aspecto se verá más en detalle e los aspectos del código en la sección siguiente.

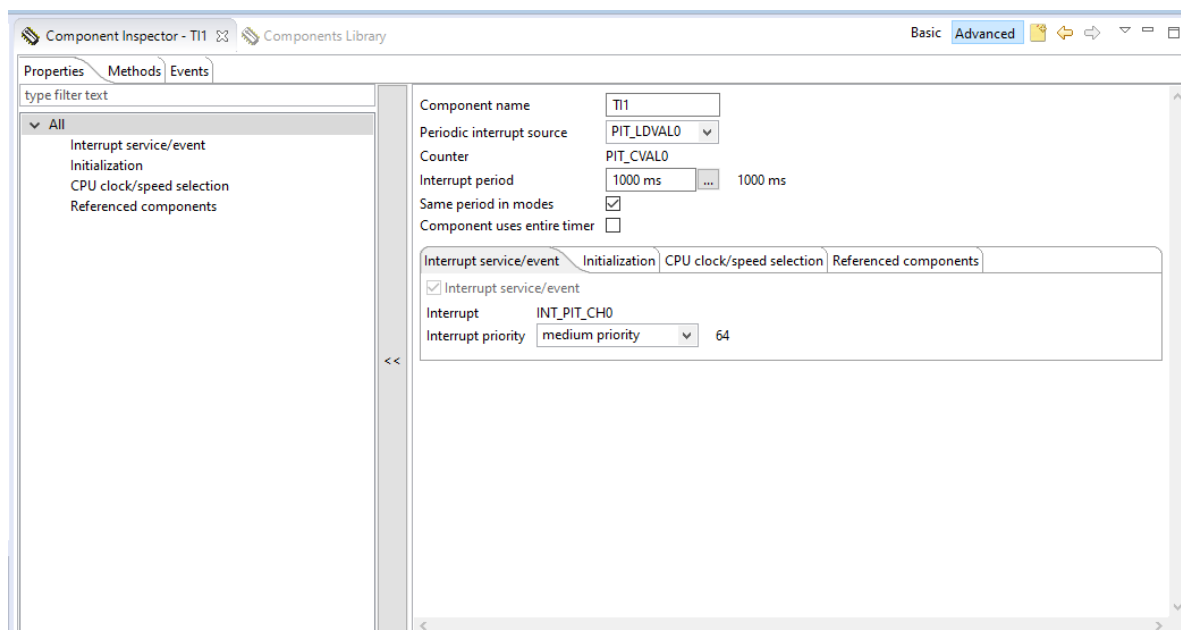


Ilustración 4: Configuración del componente TimerInt en Kinetis Studio

3.3.2. Aspectos del código

En cuanto a los aspectos del código fuente, una vez configurado el componente, se hará uso exclusivamente de una de las funciones generadas por el código del componente en el fichero 'Events.c', en la función **T1_OnInterrupt()**, donde se creará una variable externa global al código main del programa que llevará la cuenta del valor del tiempo transcurrido en el programa. De forma que, en la función del fichero de eventos, esta se ejecute cada segundo decrementando el contador hasta que se acabe el tiempo y se gestione el ganador de la partida.

3.4. RGB LED y Botones

Como ejemplo de uso del componente para representar mediante el LED central de la placa los colores representativos de cada una de las situaciones que se den en el juego se ha seguido la configuración de los pines y la generación del código mediante las funciones del componente BitsIO, creando uno para la entrada y otro para la salida.

3.4.1. Configuración Kinetis

Para configurar este componente en Kinetis Studio será necesario añadir el componente ADC al proyecto tal y como se indica en el material de las prácticas. Un vistazo a la configuración añadida se puede observar en la Ilustración 5 y en la Ilustración 6. En cuanto a la conexión de los pines a la placa, se ha hecho uso de tres pines (**PTG5, PTG6, PTG7**) para generar cada uno de los componentes de color (R, G y B) para mostrar la referencia de haz de luz del dispositivo.

En el caso del componente BitsIO de entrada (Input) se ha utilizado para el manejo del botón de la placa para el comienzo de la partida del juego. Por otro lado, el componente BitsIO de salida (Output) se ha utilizado para generar el haz de luz de color del LED central de la placa.

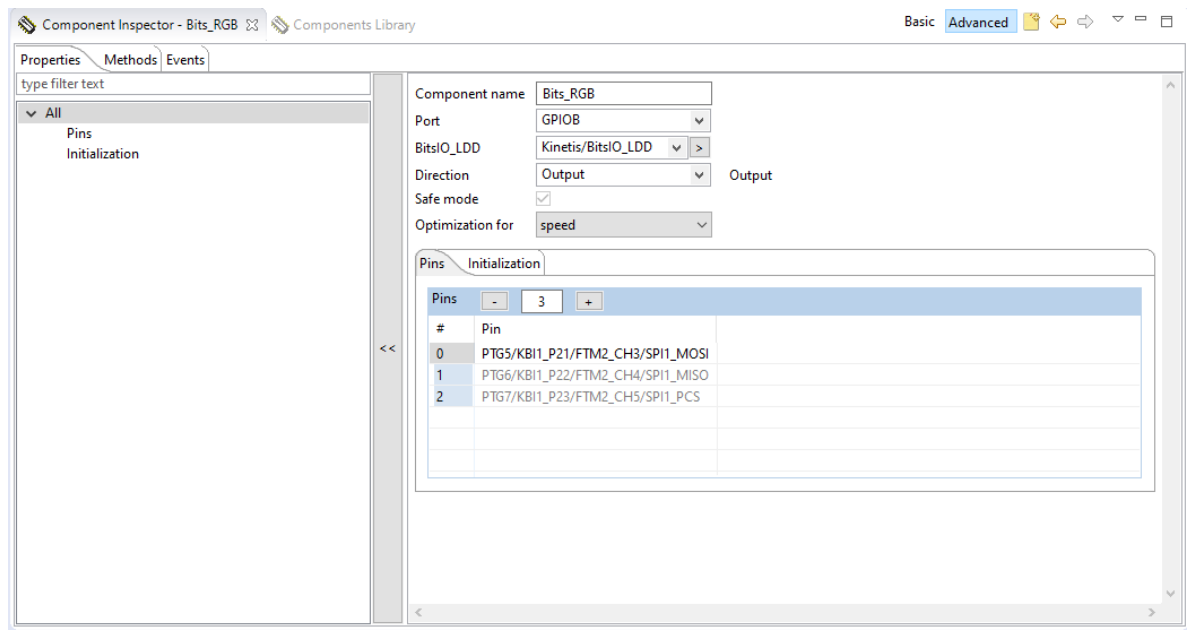


Ilustración 5: Configuración del componente BytesIO de salida del RGB en Kinetis Studio

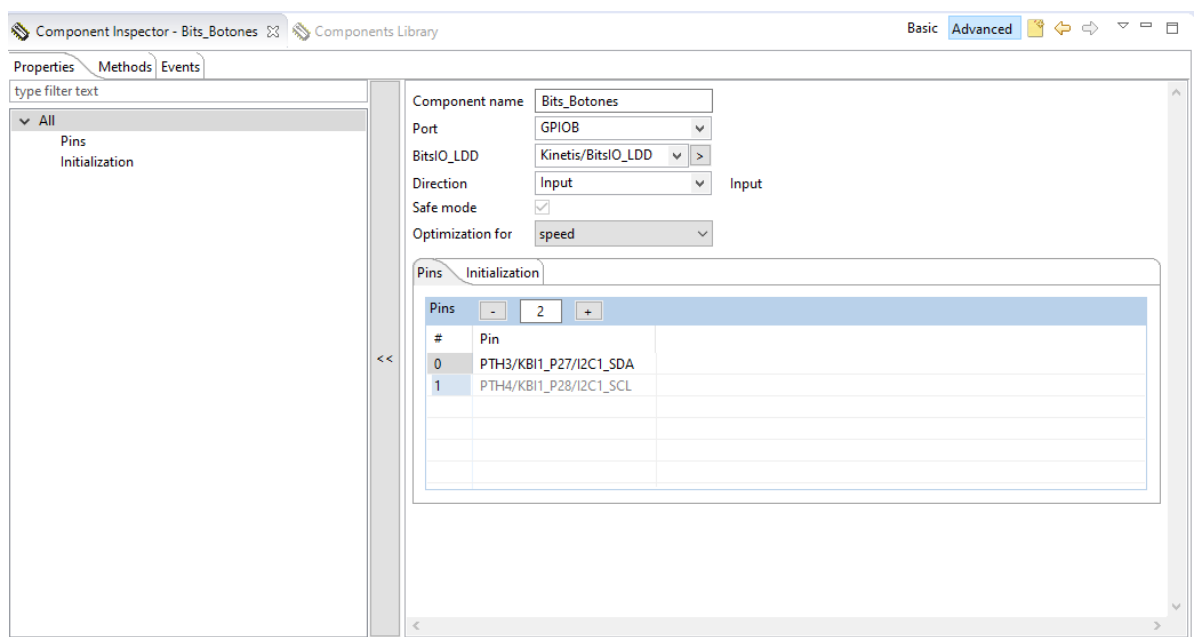


Ilustración 6: Configuración del componente BytesIO de entrada del RGB en Kinetis Studio

3.4.2. Aspectos del código

En cuanto a los aspectos del código fuente, una vez configurado el componente, se hará uso de las funciones **GetVal** y **SetVal** para asignarle por medio de macros el valor de los colores del LED, proporcionados en el material de la asignatura.

3.5. Puerto Serie

Como ejemplo de uso del componente para representar y visualizar mediante una terminal el entorno del juego con el laberinto que se dibuja y que se utilizará para manejar el juego entre los dos jugadores, los dos jugadores, el tiempo y la puntuación.

3.5.1. Configuración Kinetis

Para configurar este componente en Kinetis Studio será necesario añadir el componente ADC al proyecto tal y como se indica en el material de las prácticas. Un vistazo a la configuración añadida se puede observar en la Ilustración 7 e Ilustración 9. En cuanto a la conexión de los pines a la placa, se ha hecho uso de los pines **PTF3** y **PTF2** para la referencia a Rx y Tx para las conexiones del puerto serie.

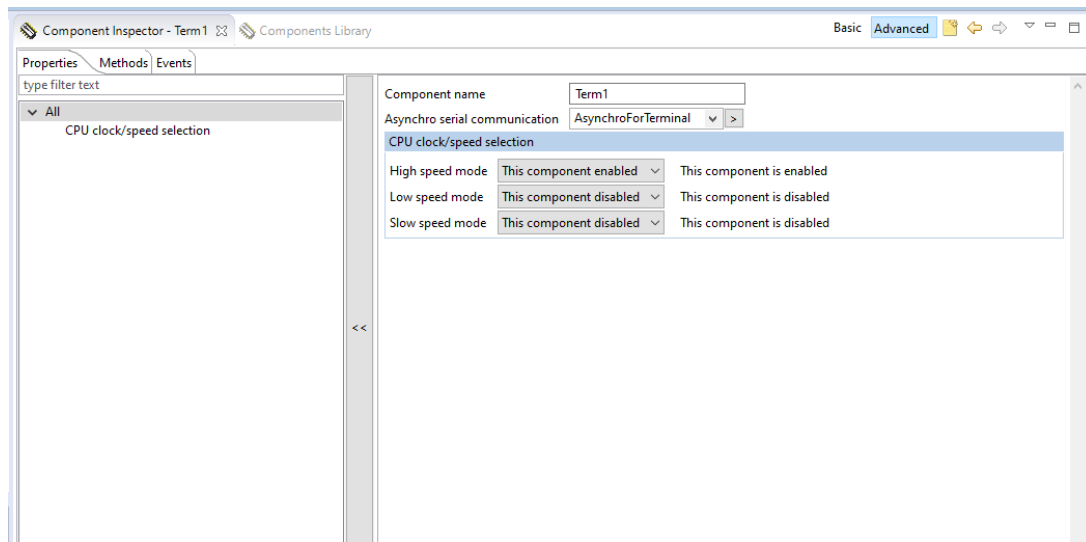


Ilustración 7: Configuración del componente Terminal en Kinetis Studio

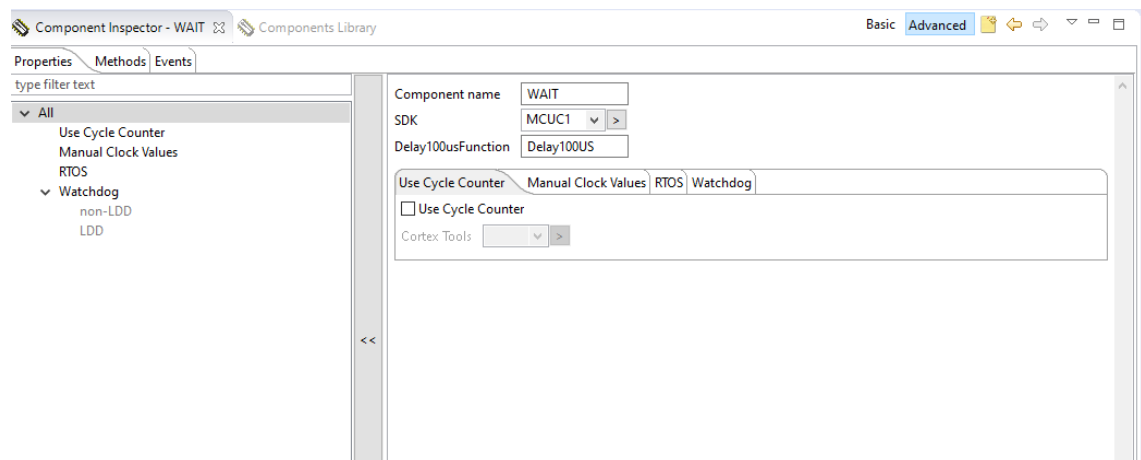


Ilustración 8: Configuración del componente Wait en Kinetis Studio

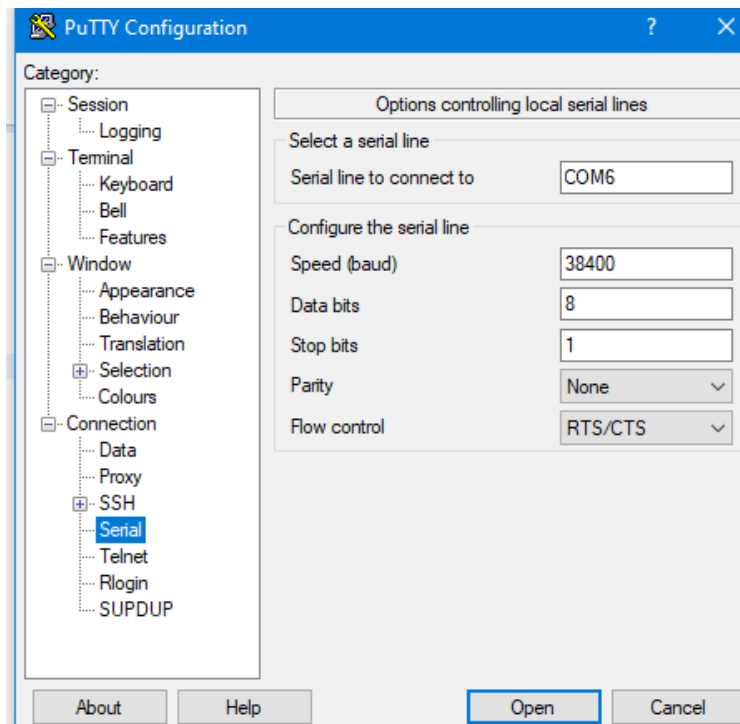


Ilustración 9: Configuración de conexión a la terminal desplegada en Putty

3.5.2. Aspectos del código

En cuanto a los aspectos del código fuente, una vez configurado el componente, se hará uso de las funciones del componente **WAIT** para establecer tiempos de espera para la generación de los bytes de representación. Por otro lado, las funciones **Term1_SendStr** y **Term1_SetColor** han permitido, por un lado, la escritura de caracteres por terminal (el laberinto y los mensajes) y la otra función el establecimiento del color de la terminal y de los jugadores. Para eso se mueve el cursor hasta la posición de la que se va a establecer el color. Un ejemplo es cuando el jugador ganador de la partida (o bien por tiempo o por llegada a la meta) se pone de color azul, cogiendo la posición en la que se encuentra.

4. Conclusiones

Como se puede ver en el informe, finalmente, se han cumplido los objetivos planteados para la práctica en el uso de los componentes para la implementación de la práctica haciendo uso del entorno de Kinetis Studio y de los componentes de la placa para el manejo de los distintos dispositivos y herramientas que han permitido la implementación final del juego.

Bibliografía

No hay ninguna fuente en el documento actual.