Scheduling mechanism in Waijung 2

View table of content.View table of content.

Objective

The objective of this documentation is to explain the code generation mechanism used by Waijung2.

Pre-requisites

This documentation assumes the reader has a basic understanding about FreeRTOS functionality. Click here to learn more about FreeRTOS types, functions, and macros.

Introduction

Waijung2 block-set is designed to program an ESP32 board using FreeRTOS. The rest of this documentation will describe how Waijung2 implement the features of FreeRTOS when generating code for your model file.

In all model files where Waijung2 blocks are used the Target Setup Block should be present in order to upload the code to an ESP32. The Target Setup Block would adjust the necessary Model Configuration Parameters, acquire hardware specifications of your ESP32 development board in order to generate the SDK file and acquire necessary data to create the base task. The base task is the default FreeRTOS Task created to place the generated code. If all the blocks in the model has the sample time specified as inherited (-1), it will be executed within this task.

When a user specify a sample time to a block (other than -1), Waijung2 generate a separate task for each different sample time in a way that the block gets executed according to the specified frequency. If two blocks share the same sample time the code generated by the two blocks will be executed within the same task created for that specific sample time.

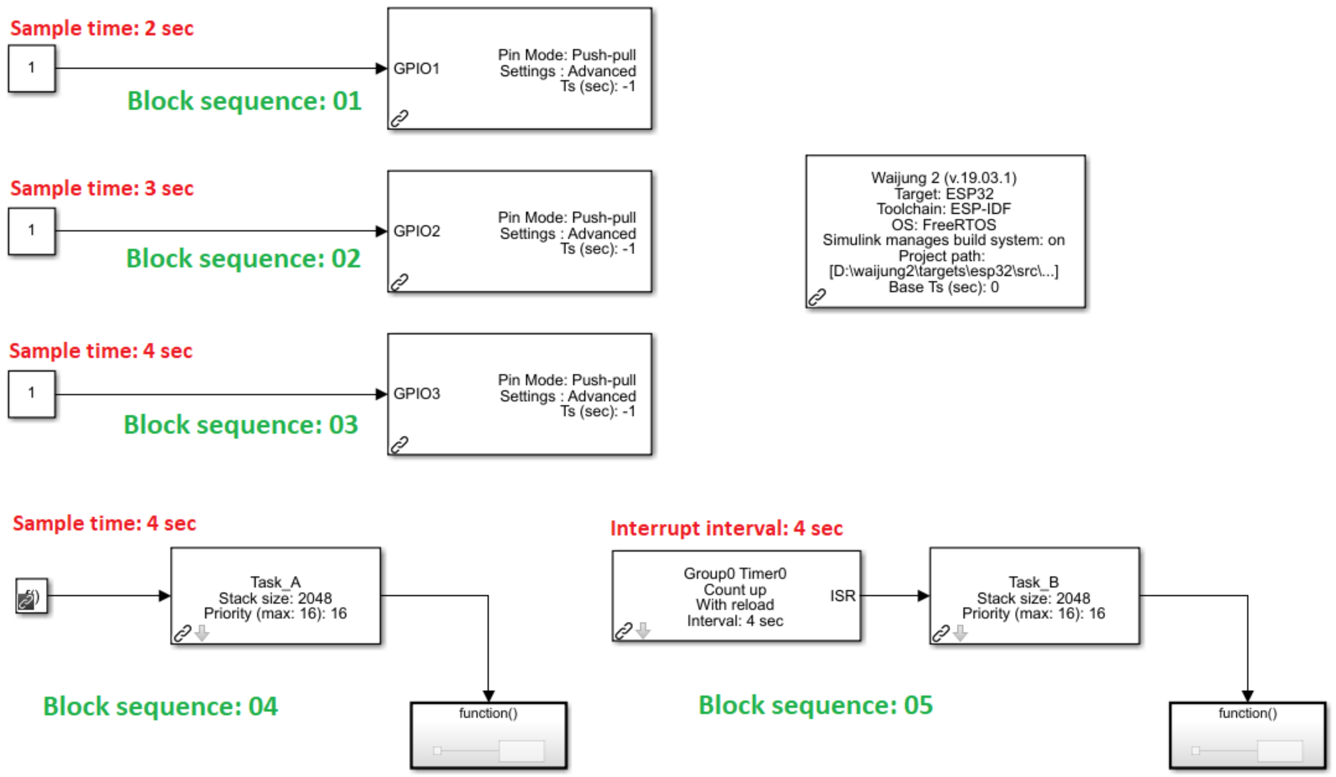
Another way to create a task is to use the xTaskCreate Block. The input to the xTaskCreate block can be of two types.

1. Function-Call Generator (From Matlab)
2. Timer Interrupt Block

When using the Function-Call Generator block the user has to specify a sample time for the block. If this sample time is used for another block in the same model, the generated code from that block and the block sequence connected to the output of xTaskCreate block will be executed in the same task. Therefore if there is a block with a blocking process (a process that is waiting for some event, such as a resource becoming available or the completion of an I/O operation), the block sequence connected to the output of the xTaskCreate block will not be executed in the specified frequency. In such case input of the xTaskCreate block should be a Timer Interrupt. This way the generated code for the two processes will be in separate tasks even though they have the same sample time. This is the guaranteed way to ensure a block sequence get executed despite blocking process of another block sequence with the same sample time.

Example

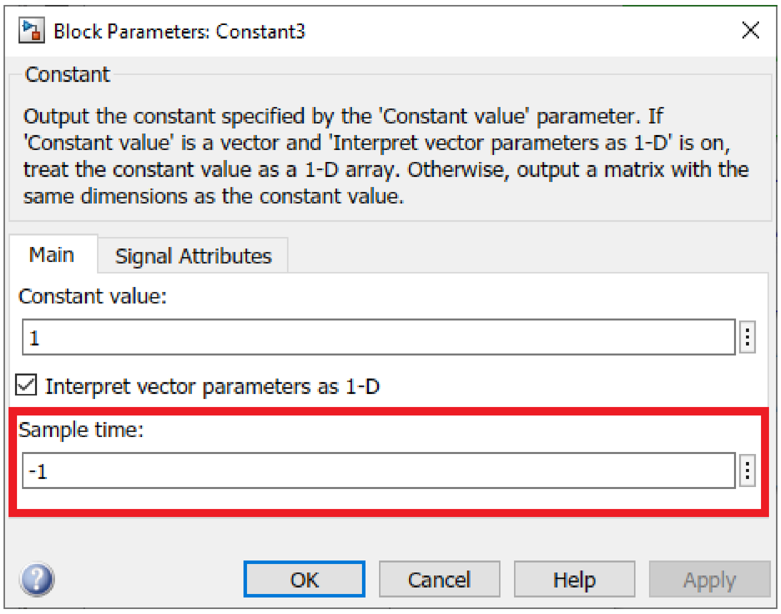
Let's analyze the model shown below with the information above. Block sequence 1,2 and 3 sample times are determined by the sample time specified in the Constant block. In block sequence 4, xTaskCreate block takes the input drive from a Function-Call Generator block while block sequence 5 xTaskCreate block takes the input drive from a Timer Interrupt Block.



Here Waijung2 will create 4 tasks. Block sequences 1,2, and 5 will each have there own task at their designated sample times while block sequences 3,4 will be run on the same task with a sample time of 4 sec. Since block sequences 3,4 are in the same task, if the block sequence 3 has a blocking process, it could affect block sequence 4 sample time as well.

Note

The sample times of the compatible blocks can be changed from the Sample time parameter in the block mask.


Block Parameters: Constant3

Constant

Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.

Main | Signal Attributes

Constant value:

1

☑ Interpret vector parameters as 1-D

Sample time:

-1

Previous : How to connect a hardware to a Wi-Fi Access Point
Next : Integrate Legacy C code to Waijung 2 using s-function builder

Files

example.png	61.7 KB	29 Oct 2020	Dhanika Mahipala (ดานิก้า)
Capture.PNG	29.2 KB	03 Nov 2020	Dhanika Mahipala (ดานิก้า)