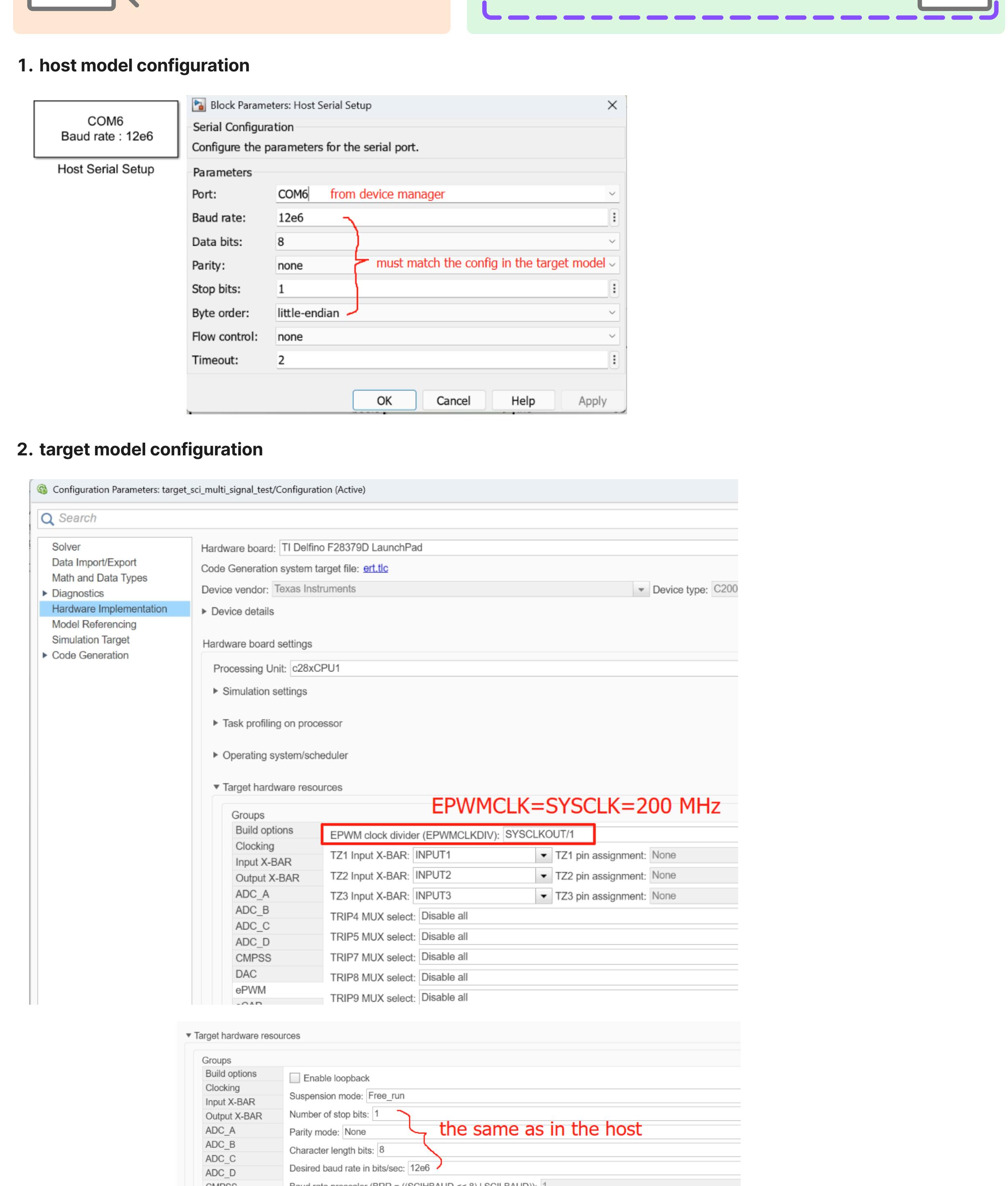
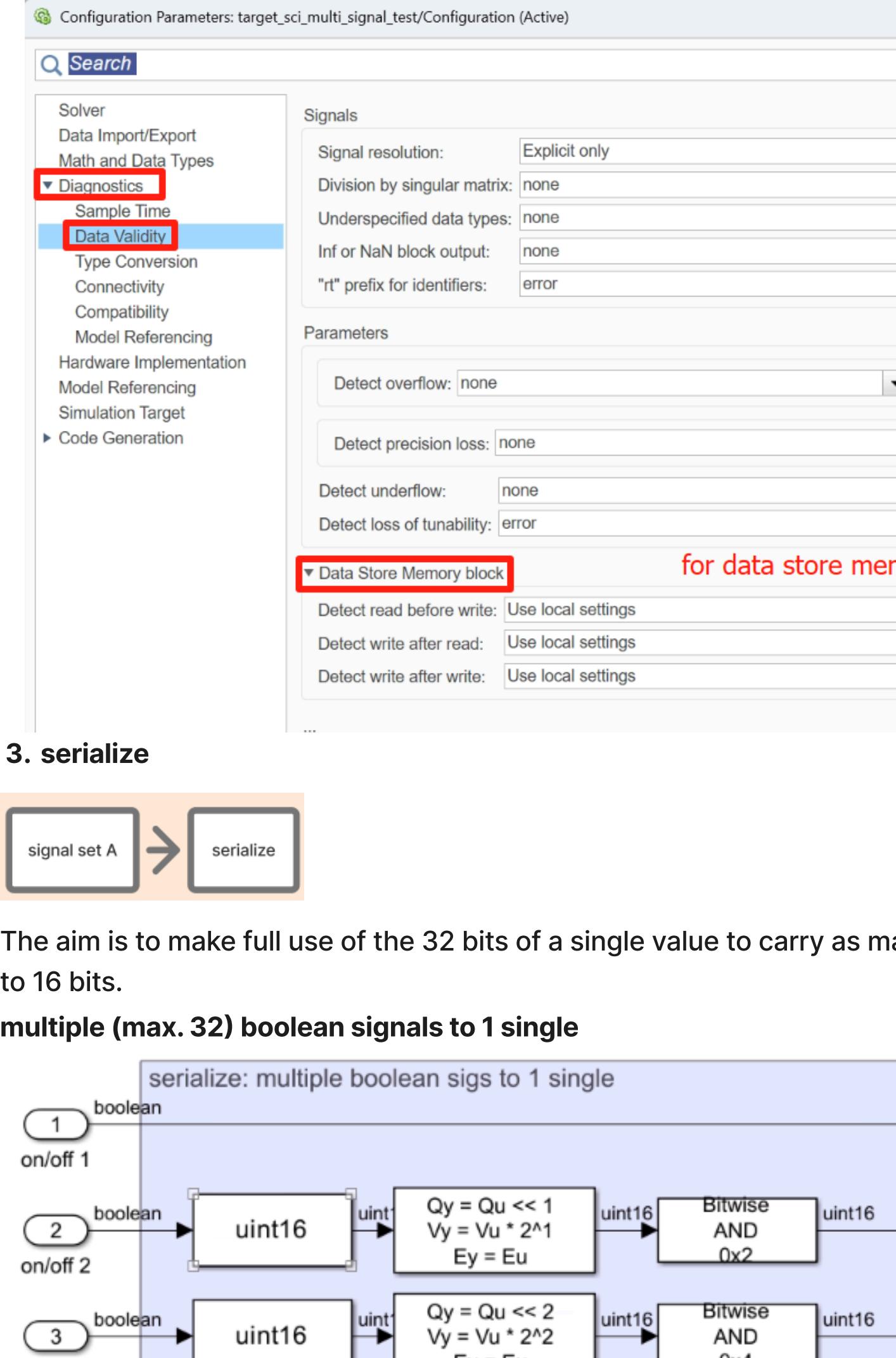


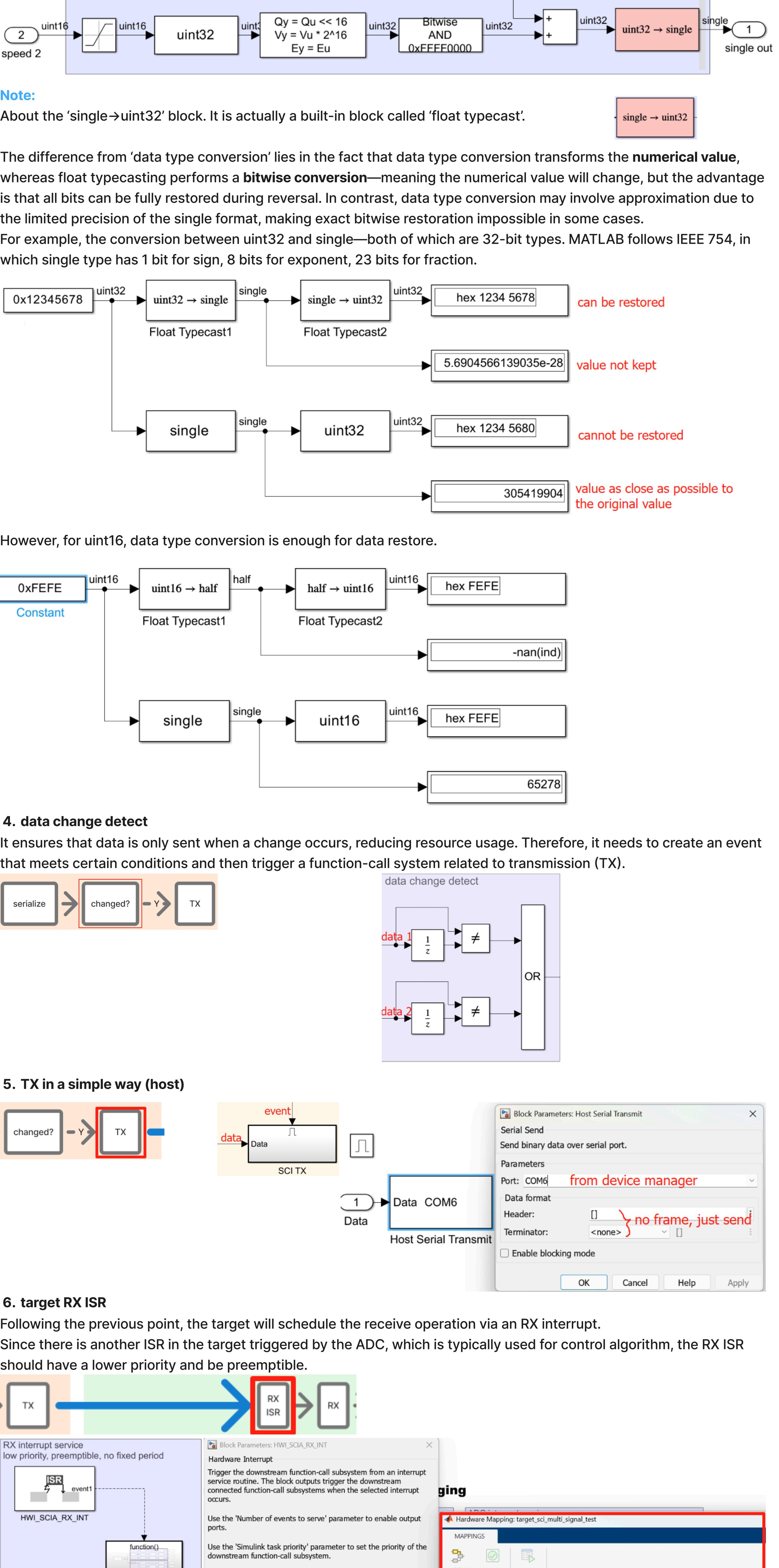
This example demonstrates how to efficiently and reliably exchange multiple signals of different types between a master and a slave device. It involves **signal merging** and **manual data frame packaging**.



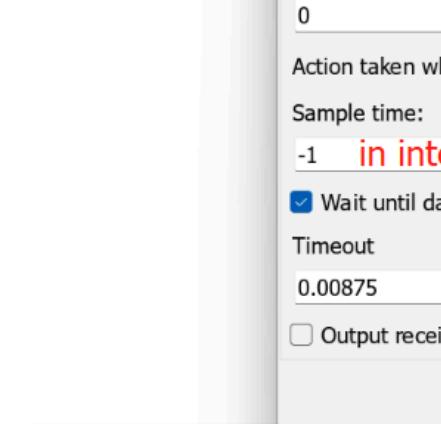
### 1. host model configuration



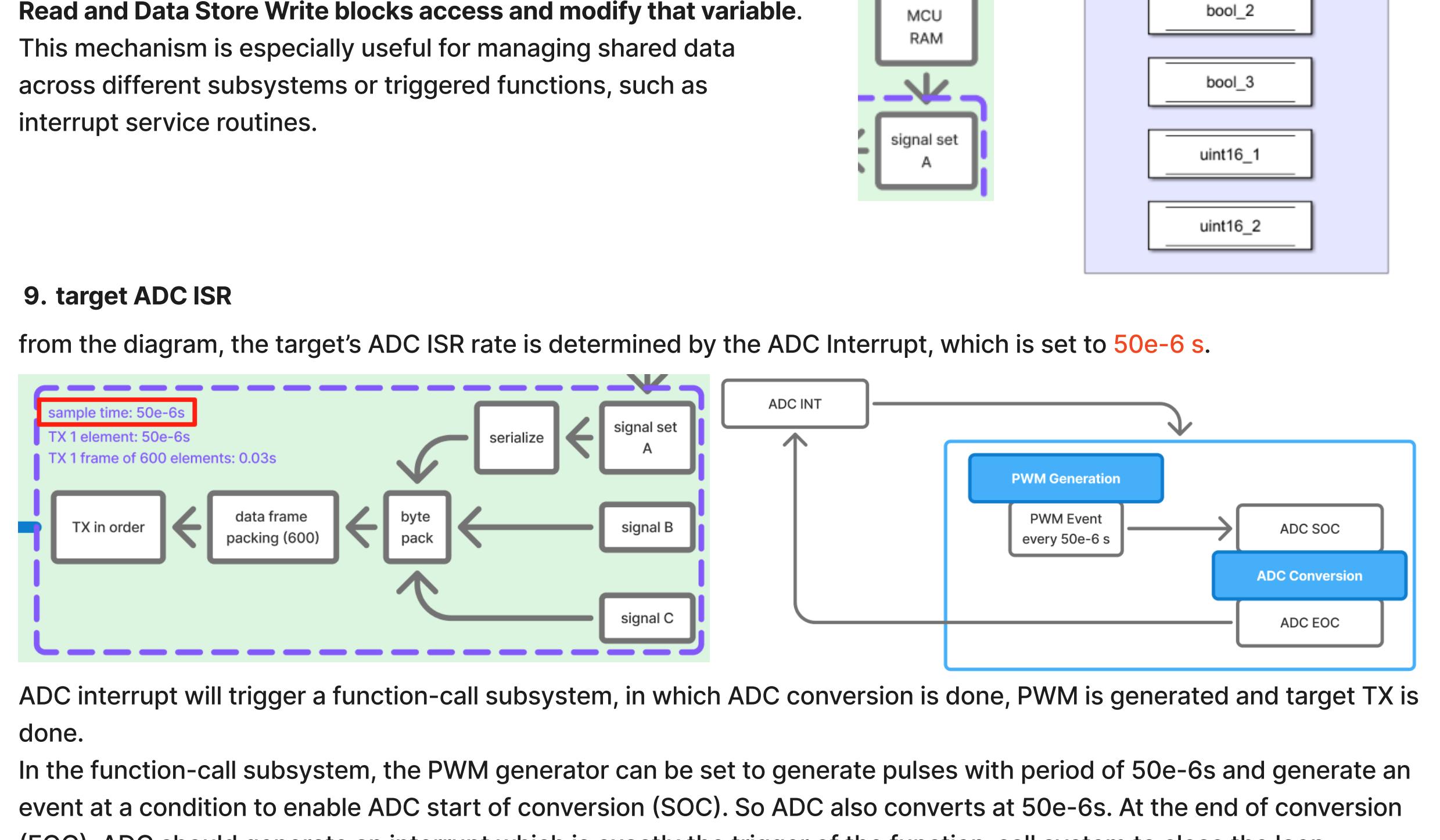
### 2. target model configuration



### 3. serialize



### 2 uint16 signals to 1 single

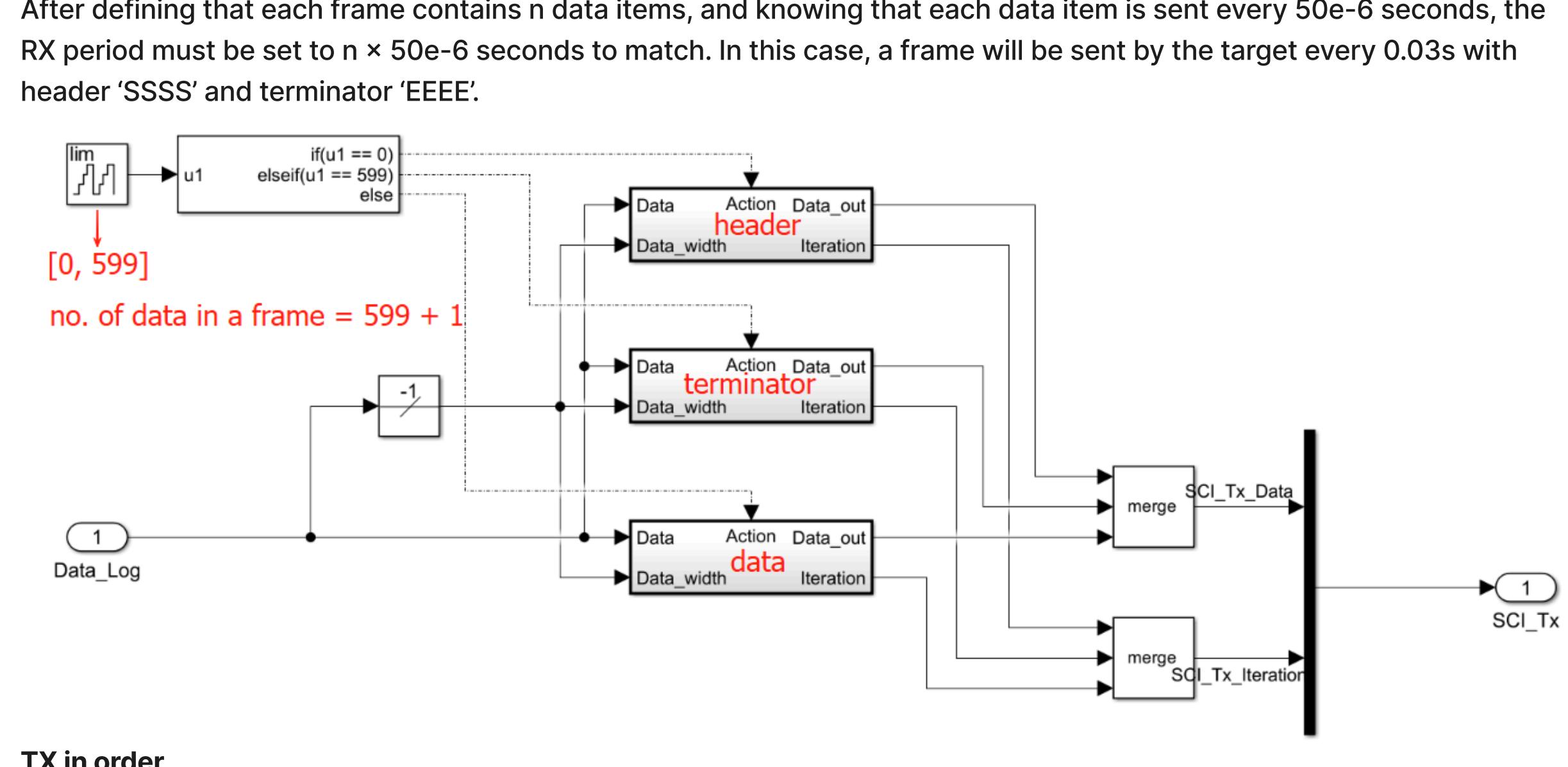


### Note:

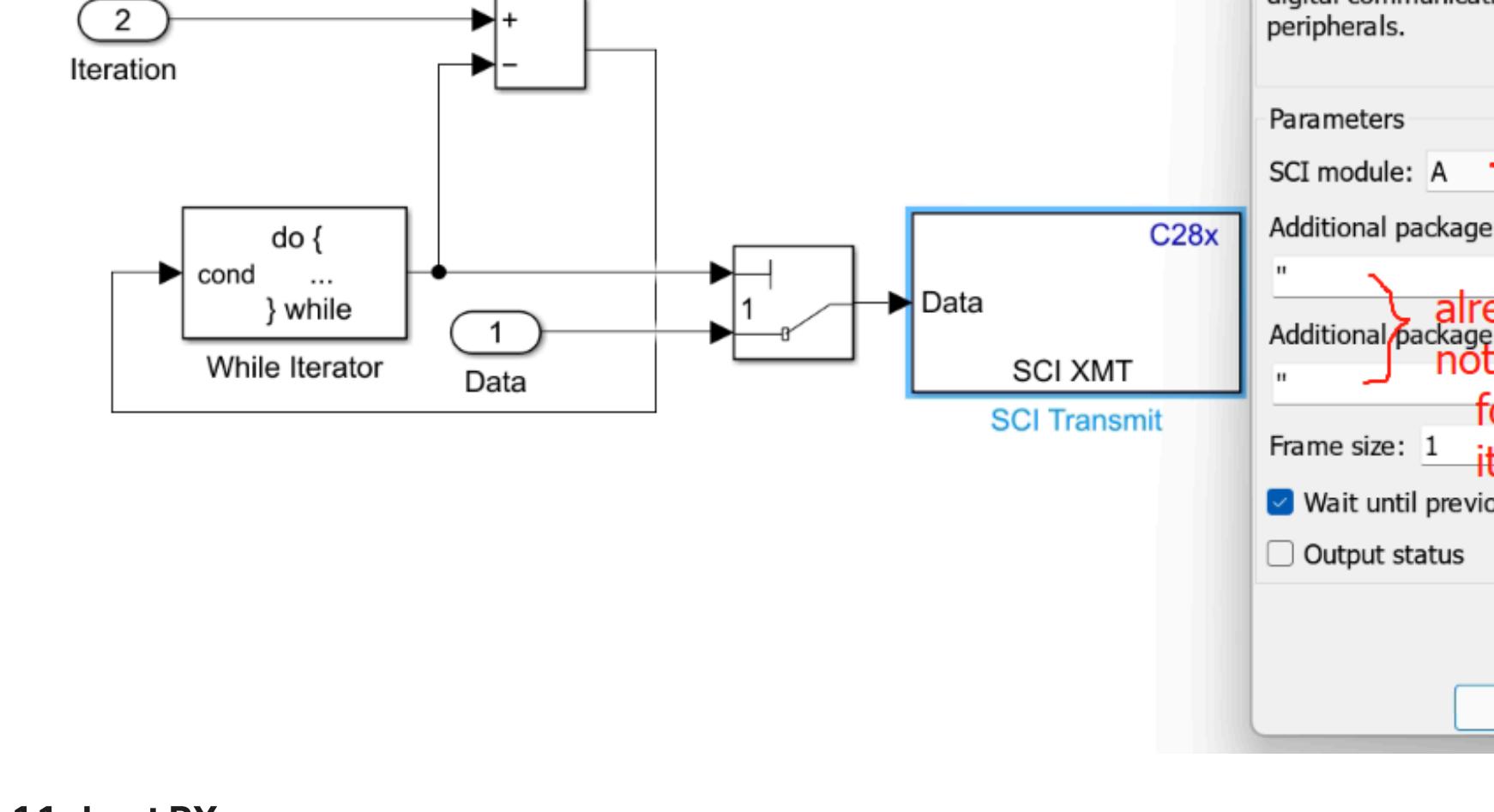
About the 'single → uint32' block. It is actually a built-in block called 'float typecast'.

The difference from 'data type conversion' lies in the fact that data type conversion transforms the **numerical value**, whereas float typecasting performs a **bitwise conversion**—meaning the numerical value will change, but the advantage is that all bits can be fully restored during reversal. In contrast, data type conversion may involve approximation due to the limited precision of the single format, making exact bitwise restoration impossible in some cases.

For example, the conversion between uint32 and single—both of which are 32-bit types. MATLAB follows IEEE 754, in which single type has 1 bit for sign, 8 bits for exponent, 23 bits for fraction.



However, for uint16, data type conversion is enough for data restore.

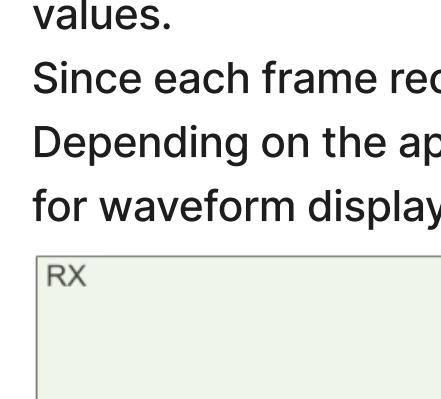


### 4. data change detect

It ensures that data is only sent when a change occurs, reducing resource usage. Therefore, it needs to create an event that meets certain conditions and then trigger a function-call system related to transmission (TX).



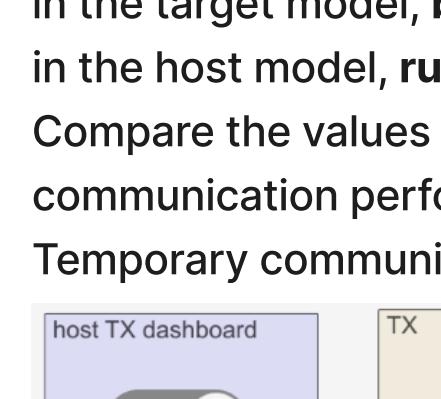
### 5. TX in a simple way (host)



### 6. target RX ISR

Following the previous point, the target will schedule the receive operation via an RX interrupt.

Since there is another ISR in the target triggered by the ADC, which is typically used for control algorithm, the RX ISR should have a lower priority and be preemptive.



### 7. target RX and parse

Following the previous point, the target will schedule the receive operation via an RX interrupt.

Since there is another ISR in the target triggered by the ADC, which is typically used for control algorithm, the RX ISR should have a lower priority and be preemptive.



### 8. target Data store memory, read and write

In Simulink for C2000, Data Store Memory, Data Store Read, and Data Store Write blocks are used to share data between different parts of a model without using direct signal lines. The Data Store Memory block defines a named global variable, while Data Store Read and Data Store Write blocks access and modify that variable.

This mechanism is especially useful for managing shared data across different subsystems or triggered functions, such as interrupt service routines.



### 9. target ADC ISR

From the diagram, the target's ADC ISR rate is determined by the ADC interrupt, which is set to 50e-6 s.



### 10. target TX through custom data frame packing

The purpose and method of serialization have already been detailed in Point 3 and will not be repeated here.



### 11. host RX



### 12. host RX data parse

The reverse process of serialization done in target TX, used to restore the transmitted signals to their original types and values.

Since each frame receives 600 data points, after being split into groups of 4 in the host RX, they are stored as arrays. Depending on the application requirements, these arrays can either be processed into scalars or retained in array form for waveform display on a scope.



### 13. debug

In the target model, build, deploy and start.

In the host model, run.

Compare the values sent from the host to the target with the values returned from the target to the host to evaluate the communication performance and accuracy.

Temporary communication interruptions may occur — they can be improved, but not completely avoided.

