Dezyne Challenge 2016/2017: A Scale, by Mark Weber.

Introduction.
For a hobby project I am experimenting with building a scale with the possibility to process the measured weight. These are commercial available, but way too expensive for a fun application and not suitable to build in your own enclosure. Around the time I was doing a little research the Dezyne Challenge was started. So, why not try to model this project in Dezyne?

The site of SparkFun explains exactly how to build a scale and which components are needed:
- a load cell, i.e. a piece of metal with a strain gauge glued to it.
- an ADC to readout the load cell
- an Arduino to readout the ADC and process the measured value.

The System
The modeled system consists of:
- an interface to enable / disable the scale.
- a display interface, which can be in fact any other module for processing the data.
- a tare interface.
- a serial protocol to readout the ADC

Modeled Functionality
Auto zero calibration
When enabling the scale, the first measured weight is considered the tare value. This value is stored and all future measurements are compensated with this value.

Tare functionality
This can be used to do a new zero calibration at anytime. It was a nice little exercise which showed how to process data: send the measured weight out of the Dezyne model and return a corrected value.

Periodic updates
The system periodically measures the weight and sends it to a display. I think it is valuable when Dezyne would offer timers in a kind of library.

The serial protocol to readout the ADC
It makes use of a register, a data bit and a clock bit. The datasheet of the ADC (HX711) describes the protocol. This was the most educational part to me, because I do not have experience with this kind of interfacing. Basically it is generating a clock pulse, reading the data bit and writing it into a register until it is full. Then wait until the ADC indicates a new measurement is available.

Lessons learned
1. The Dezyne is A LOT easier to use and more intuitive to programmers than ASD.
2. By modeling it this way, it becomes clear how much complexity there is in such a simple system.

Questions
After adding the Register to my model there was a life lock. It has been fixed by using a SubInt, but that’s not a flexible solution. Any idea how to solve this?