**The ugly:**

In dit artikel gaat Elliot Williams in op de minder mooie aspecten van het gebruiken van bijvoorbeeld interrupts.

The “obvious” pitfall when sharing variables with ISRs is the [race condition](https://en.wikipedia.org/wiki/Race_condition): your code sets the variable’s value here and then uses it again there. The “race” in question is whether your code can get fast enough from point A to point B without an interrupt occurring in the middle.

Als je variabelen toegankelijk maakt voor je main én interrupts kan je nooit weten wanneer een interrupt zal worden geactiveerd (murphy’s law) waardoor je variabele kan veranderen naar iets ongewenst na het checken op deze ongewenste waardes.

Een mogelijke oplossing is het gebruiken van een shadow copy. Dit brengt dan wel enige problemen met zich mee wanneer de code op een 8 of 16 bit ARM machine wordt gebruikt.

Een variabele zal namelijk hoogstwaarschijnlijk (op een 8/16bit machine) in meerdere regels machine code worden opgeslagen (en is dus niet “Atomic”). Een interrupt tijdens het uitvoeren van deze code kan foutieve waardes opleveren.

Als laatste komt men met een goed werkende oplossing. Men zet gewoon de interrupts tijdelijk uit terwijl de main methode met de meest huidige variable werkt.