Understanding external bypass variants

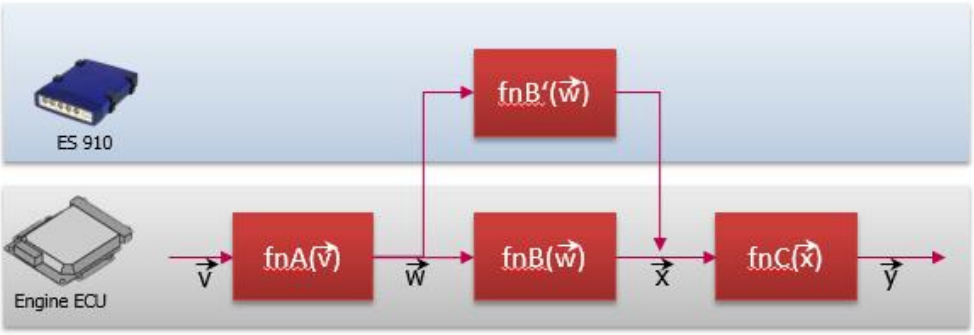
Use Case

Getting the best performance from a ES910 bypass set up

External bypassing is a common method to rapidly prototype or enhance an ECU algorithm. Data read from the ECU is used as inputs to the bypass algorithm which is created using modelling tools like Simulink or ASCET and executed externally on the ES910. The output of the bypass is fed back into the ECU and used instead of the output of the algorithm calculated on the ECU itself.
With this method, the new algorithm can be tested in a real time and closed-loop environment before undergoing the effort implicated by the process of developing ECU production software.

Do you know the advantages of the different ETK bypass types?

There are two different ETK bypass types available: Hook based bypass (HBB) and Service based bypass (SBB). In recent ECUs supporting ETK bypass, both variants are offered and can be used for external rapid prototyping - they even can be used within the same bypass experiment. EHOOKS also supports both methods by easily adding hooks (HBB) and service points (SBB).
However, both bypass types have their specific advantages, so understanding how they work can be of great benefit choosing the optimal variant for your use case to ensure getting the best performance out of your ETAS prototyping system.



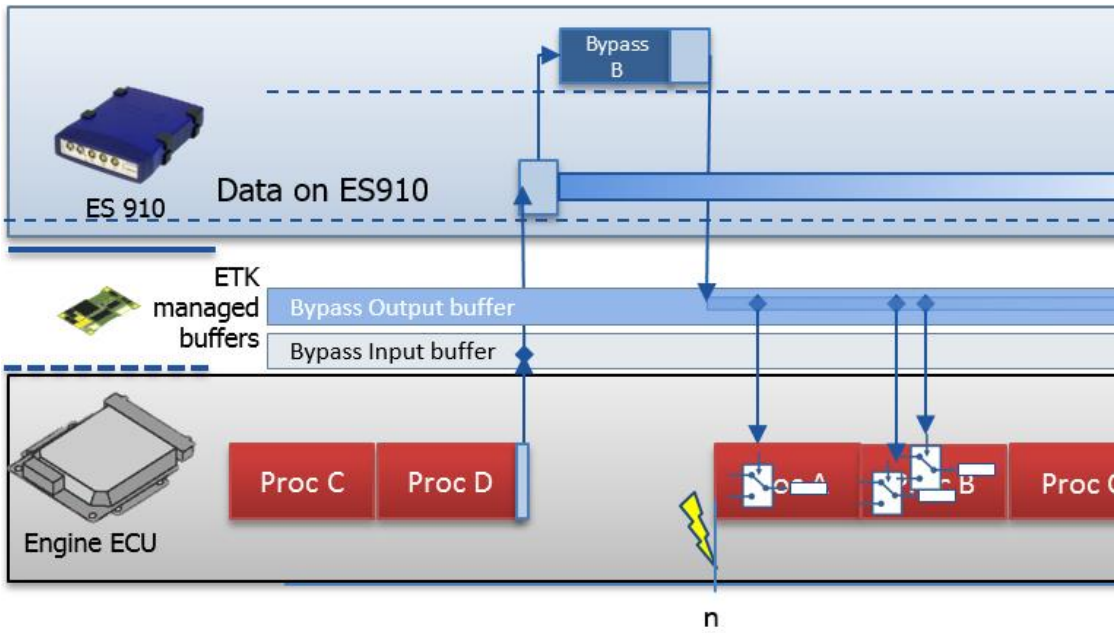
External function bypass

The bypass types – how they work

The communication mechanism between the ECU and the RP system is the same for both HBB and SBB. In both cases, a bypass configuration, generated by INTECRIO, RLINK or ASCET-RP, is written into memory managed by the ETK (for better reading, "ETK buffer"). Based on this, the bypass input values are copied from the ECU into memory accessible by the ES910 and the bypass is triggered. After calculating the bypass model, the results of the calculation are written back into another ETK buffer from where they can be accessed by the ECU.
The difference between HBB and SBB basically is **when** bypass input data is copied from the ECU into ETK and **when** data is written back.

Hook based bypass (HBB)

In HBB, all bypass input variables of bypasses from one ECU raster are copied at once, typically at the end of the ECU task. As a consequence, the ECU load for copying this data is minimized. Likewise, there is only one bypass trigger – i.e. one interrupt – for each ECU raster on the ES910, minimizing interrupt load on the ES910.



HBB data flow

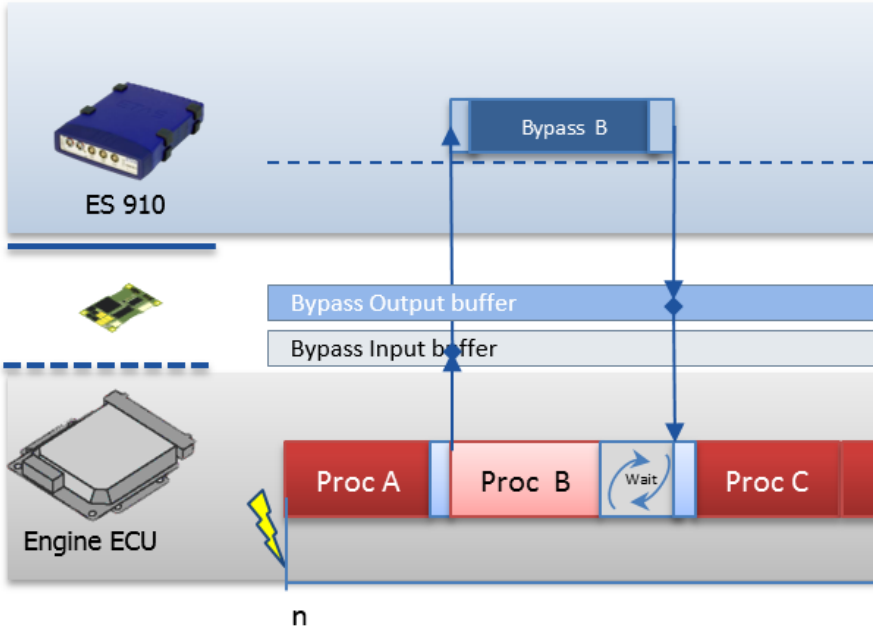
After bypass calculation, the outputs are written into the ETK bypass output buffer, from where each variable is read back at the corresponding hook in the ECU software. If this variable is written in different processes or branches in the ECU, it will be copied wherever the ECU software writes that variable.

From this, the following advantages of HBB can be derived:

- Minimal impact on ECU load and timing behavior
- data-oriented bypass – one or more variables can be bypassed, wherever they are written in the ECU, even process-internal values.
- No information on ECU scheduling needed
- Minimal interrupt load on ES910

Service based bypass (SBB)

In SBB, the individual bypass input variables of the bypass are copied at each service point. As a consequence, the bypass always gets the most recent ECU values, so the bypass can be executed with exactly the same inputs and exactly at the same time as the bypassed ECU function. Thus, SBB is the bypass version to chose if the bypass use case is replacing or improving an existing ECU function (that is, the bypass using the same inputs and outputs as the original ECU implementation). The user should have some idea of the ECU's scheduling, which he typically gets from the name of the service point, referring to the ECU function's name.

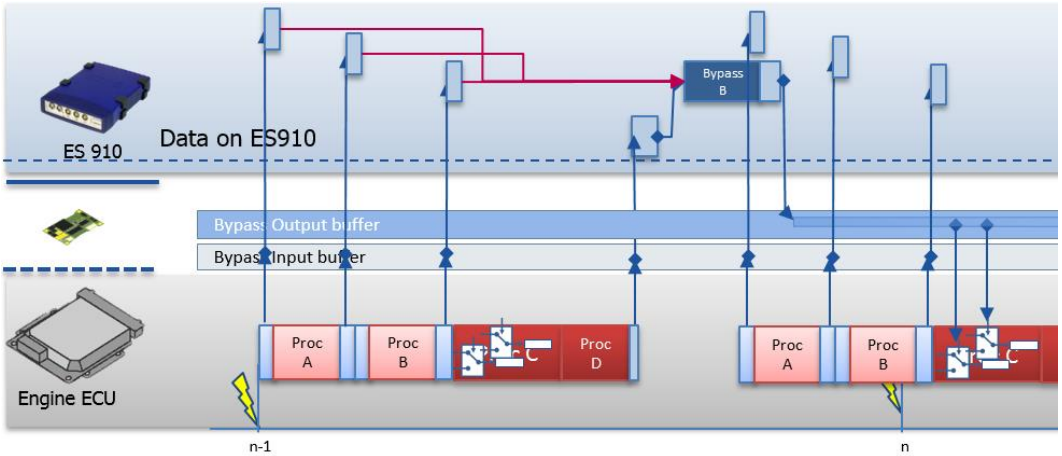


SBB data flow

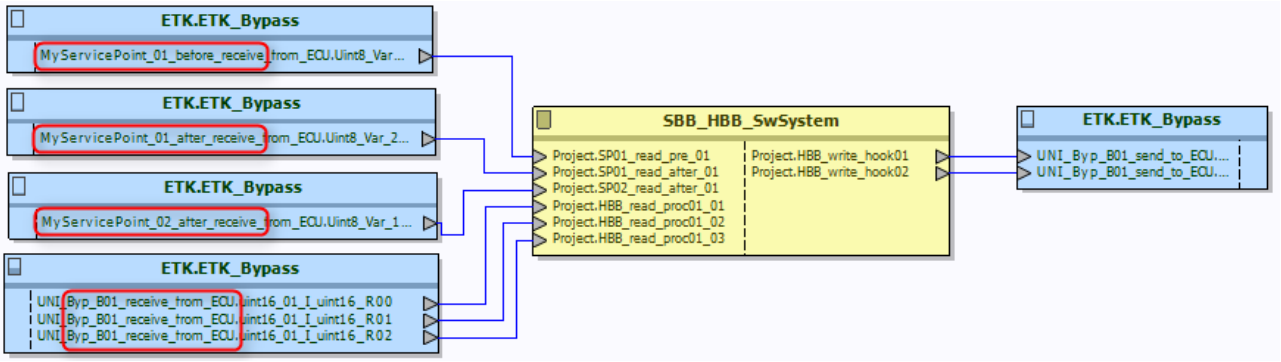
The second feature of the SBB is that bypass output data is written back to the ETK output buffer and from there to the ECU's memory at one point in time (typically at the end of the service point). Thus, this write access can by synchronized with the ECU's scheduling by assigning a wait time in the service point's configuration, in case the ECU's process execution is finished before bypass data is available. If the bypass output data is available before the ECU finished the execution of the bypassed process, the bypass output data remains in the ETK output buffer and will be copied from there afterwards. This setup allows a fully synchronized bypass execution with no raster delay.

Hook and Service based bypass combined

Since the ES910 provides a real operating system, all bypass input data can be made available to all processes running on the ES910. This allows decoupling of bypass calculation and write back from the trigger events in the ECU, leveraging load an minimizing impact to the ECU.



HBB triggered bypass using inputs sampled by SBB



HBB triggered bypass using inputs sampled by SBB – INTECRIO System set up

Tools used

- ASCET-RP
- INTECRIO
- INTECRIO-RLINK
- ES910

Your ETAS Contact

Markus Gebhardt; +49 711 3423 - 2278
markus.gebhardt@etas.com