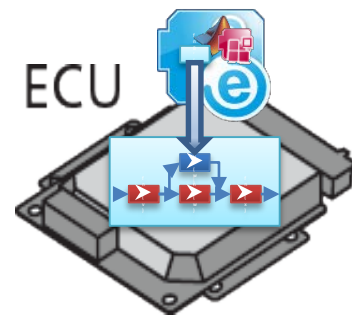


External signals in internal bypass

Using EHOOKS to read additional signals in on-target software

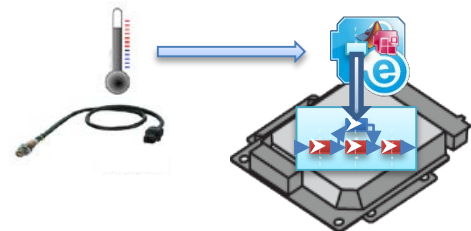
Internal bypass is a fast way to test or validate new or extended functions in the real ECU environment without struggling with the complex ECU's build process. With EHOOKS, using internal bypass has become a very simple and easy to use task. The user just needs to select the variables to be written and when the bypass functions shall be executed. With the EHOOKS integration for modelling tools like ETAS ASCET or Matlab Simulink®, this even can be done directly from within these tools.

Depending on the use case, both floating point or integer arithmetic can be used. After hitting the build button, EHOOKS will generate a new ECU hex file that contains the original ECU hex file merged with the newly developed and compiled functionality as well as an updated A2L file that, in addition to all ECU measurements and calibrations, includes the newly added measurements, calibrations and control variables. Advantages of internal bypass are the on-ECU-target capabilities - the bypass can be tested in exactly the same hardware environment as the final production code, using the same basic software and running under the same real time conditions



Limitations of internal bypass

The main limitation is caused by the ECU's resources. EHOOKS provides access to all internal signals and variables in the ECU (as defined in the preparation), but sometimes there is the need to read in additional sensor values or stimuli that do not exist on the ECU. These are typical use cases for an external bypass, which EHOOKS also supports.



The solution

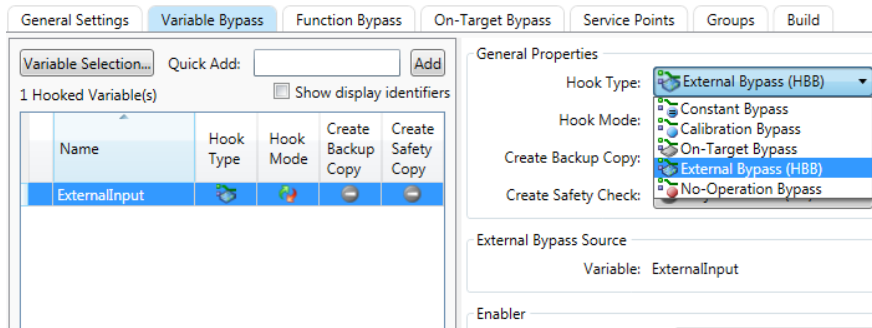
However, with EHOOKS there is a way to combine the best of both worlds: access to external signals and sensors while running the bypass on ECU target as a part of the production code environment. This is possible due to EHOOKS support of the Hook Based Bypass (HBB) mechanism. This bypass method does not only copy bypass values to an ECU address, but also takes care that this address is exclusively written by the bypass. When EHOOKS is hooking a variable, there are two things to be done at a hook:

1. prevent the ECU code from writing to the variable's address
2. retrieving an alternative value and writing it to the variable's address

The latter action can mean reading the value from a calibration, an internal buffer (that is, getting the value from an internal bypass), or from the buffer containing values written by external bypass. When "external bypass" is selected for the definition of the hook type, EHOOKS will call an ECU provided function which accesses the bypass buffer and returns a value of the specified type.

External signals in internal bypass

Using EHOOKS to read additional signals in on-target software



Add a variable for hook based external bypass in the EHOOKS GUI

The callback interface functions are defined in EHOOKS as

```
<EH_datatype> EH_tier1_getExt<datatype>(_handle, _default)
```

for scalar values. These callbacks are known within the EHOOKS build system and can also be used on target bypass code.

_handle is the EHOOKS-created structure used to define the bypass output buffer and the position within the buffer where the value can be found. The name EHOOKS generates will be *EH_chanvec_<variable_name>_handle*.

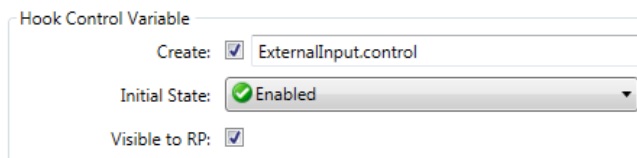
_default is the value which is returned if there is no bypass or it is not running properly. For example, for reading an unsigned byte value with the measurement name *ExternallInput* from an external ES900 RP system with the default value "0", the following code can be used:

```
EH_UBYTE ExternalInput = EH_Tier1_GetExtByte(EH_chanvec_ExternalInput, 0);
```

For more details, please see chapter 9.5 in the EHOOKS-DEV user guide, Add the On-Target Bypass Files to Configuration.

Creating new signals

The example above shows how to add an external bypass hook to a measurement value introduced by an internal bypass. GUI when defining an optional Control Variable: For EHOOKs generated *control variables* this can be achieved with the *Visible for RP* attribute, which can be set in the



Make variables introduced by EHOOKS visible to external bypass

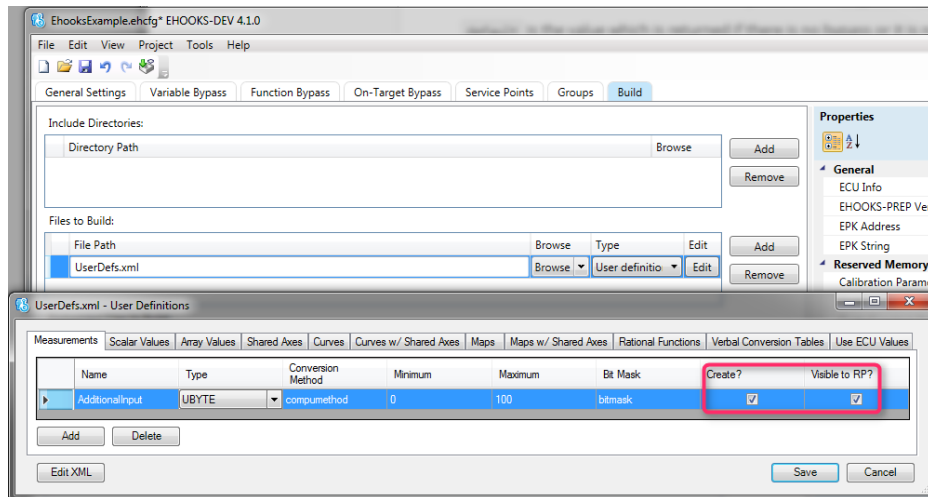
However, EHOOKS can also create new internal variables by defining them in a *User Definitions* file used by the EHOOKS build process.

External signals in internal bypass

Using EHOOKS to read additional signals in on-target software

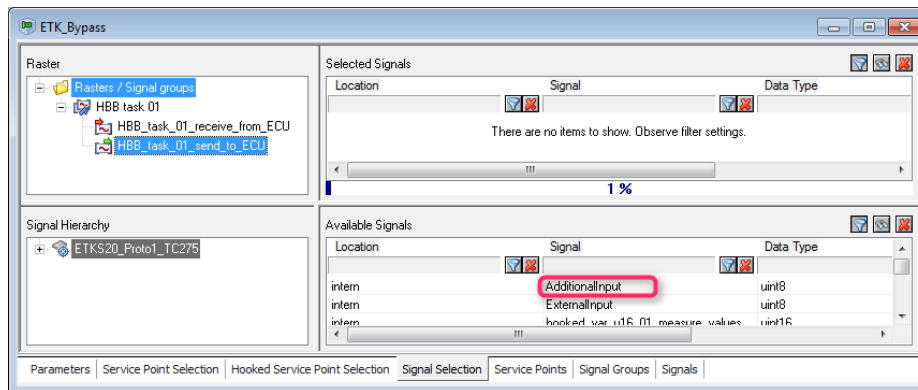
A new variable can be defined and added to the EHOOKS build process by adding a User Definitions XML description as a „file to build“ in the *Build* tab of the EHOOKS GUI.

By adding the *Create* attribute, the variable will not only be added to the A2L, but also will be defined in the generated code. Setting the *Visible for RP* attribute will allow to use this variable in INTECRIO.



Create a new measurement entry and add it to the A2L for external access

This will add the measurement into the a2l file, but as a measurement hooked for external bypass. It is created by EHOOKS and consumed by external bypass tools ASCET-RP or INTECRIO, where it can be selected as bypass output:



Select the new ECU variable for writing by an external signal

Using new signals

With this approach, the internal bypass can use data from the ES910 external prototyping system. These values may contain data read in from the CAN or LIN bus, or they are representing digital or analog input signals from any of the ETAS Daisy chain modules (ES4xx, ES63x or ES930). There are plenty use cases which can benefit from this feature.

One example use case will be described on the following page.

External signals in internal bypass

Using EHOOKS to read additional signals in on-target software

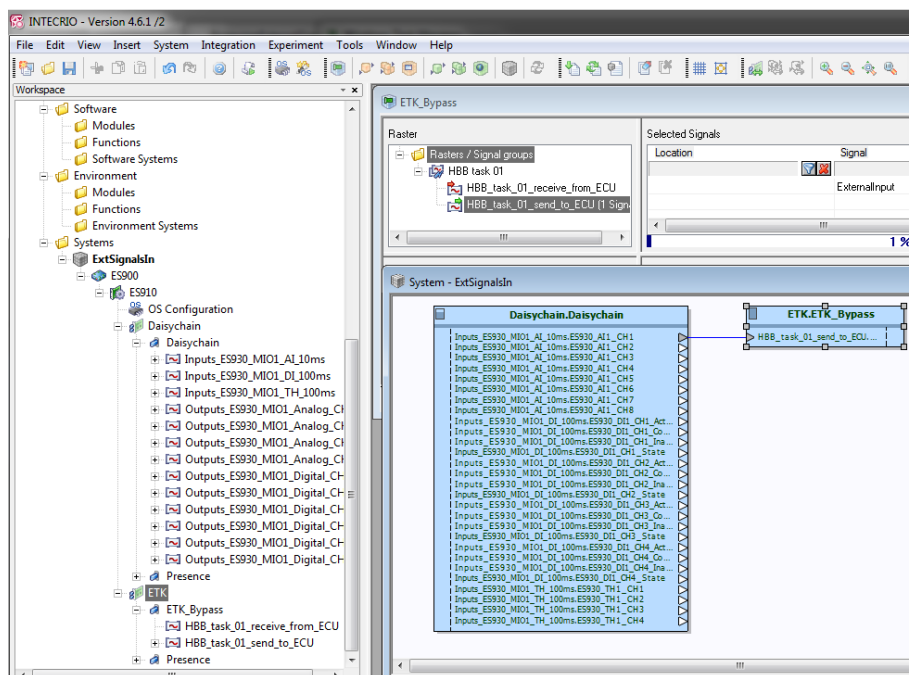
External signals

Use case:

Allow access to data from additional A/D-sensors which are available during development phase in internal bypass.

Solution:

1. DaisyChain Modules to these Configure the ETAS I/O modules, for example the ES930, to read in A/D
2. Set up an ASCET-RP or INTECRIO workspace with the ES930 configuration
3. Add an ETK bypass to the workspace
4. Create a system containing both hardware entries in INTECRIO and connect the signals, or (in ASCET-SD) add a module to map the signals:
5. Set up the access of EHOOKS added internal variables as described above and assign A/D data coming from the „new“ internal ECU variables.



Tools used

EHOOKS, ES910, ASCET-RP or INTECRIO

Your ETAS Contact

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