Advanced No-Hooks Toolkit

ATI ACCURATE TECHNOLOGIES

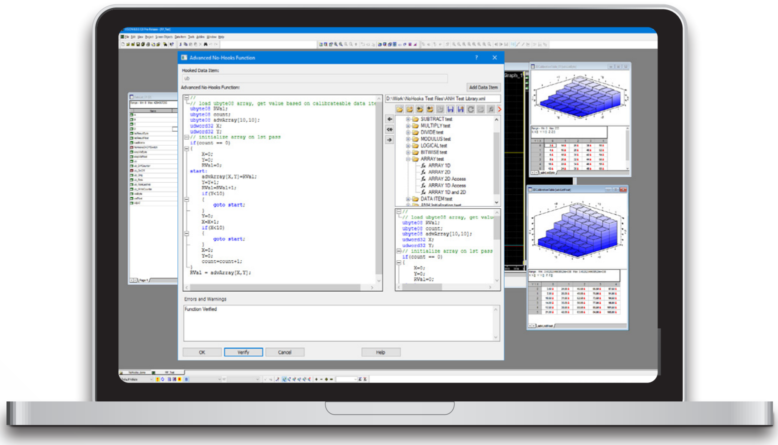
VISION • No-Hooks • Advanced No-Hooks

Advanced No-Hooks

Dynamic Hooks Without MATLAB/Simulink

Accurate Technologies' Advanced No-Hooks is based on ATI's patented No-Hooks technology, extending it to allow ECU functionality to be modified by user defined functions, without access to the original ECU source code or the need for other applications such as MATLAB/Simulink.

- User defines a function in C like syntax for the dynamic hook value using VISION data items, constants, and Advanced No-Hooks variables
- No-Hooks DLL creates relocation and trigger code
- Function Library and function editing with syntax highlighting.
- Requires only No-Hooks DLL for target CPU


Advanced No-Hooks Function

Hooked Data Item: ub

Advanced No-Hooks Function:

```
//
// load ubyte08 array, get value based on calibrateable data ite
ubyte08 RVal;
ubyte08 count;
ubyte08 advArray[10,10];
udword32 X;
udword32 Y;
// initialize array on 1st pass
if(count == 0)
{
    X=0;
    Y=0;
    RVal=0;
start:
    advArray[X,Y]=RVal;
    Y=Y+1;
    RVal=RVal+1;
    if(Y<10)
    {
        goto start;
    }
    Y=0;
    X=X+1;
    if(X<10)
    {
        goto start;
    }
    X=0;
    Y=0;
    count=count+1;
}
RVal = advArray[X,Y];
```

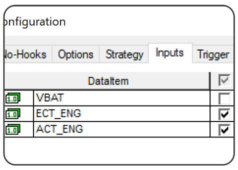
D:\Work\NoHooks Test Files\ANH Test Library.xml

Errors and Warnings

Function Verified

Advanced No-Hooks shortens the development cycle to bring products to market faster. Prototype runs on production intent ECU which corresponds in every way to improving the success of the final product. The Advanced No-Hooks bypass runs in-line with the rest of the original code on the target ECU. There is no need for data transfer protocols or the resulting delays for running the bypass on external devices, which translates into virtually no latency.

- Access to the original ECU source code is not needed, only the software memory image (hex, s-record, binary or other) and data-item description file (ASAP2 file or other).
- No need to implement "hooks" in advance of using Advanced No-Hooks.
- Implementation of Advanced No-Hooks can be performed at any time in the development process.
- No re-compiling of the original source code is needed.
- Simultaneously view the original variable value along with the bypassed value and values declared in the Advanced No-Hooks functions.
- No additional hardware or software tools other than ATI's VISION with No-Hooks and Advanced No-Hooks.
- All ATI's Rapid Prototyping products work seamlessly with the calibration, signal monitoring, data acquisition, and post analysis capability of the VISION Calibration and Data Acquisition Software.


onfiguration

DataItem	☑
VBAT	☐
ECT_ENG	☑
ACT_ENG	☑

Each hooked item can have its own function. Hooked item functions can be selectively enabled/disabled to control the bypass operation as an Advanced or base No-Hooks bypass.

Overview of Advanced No-Hooks Functionality

Create User Defined Functions

User defined functions can be created using C like syntax. Functions consist of one or more single line statements ending with a ‘;’ or code blocks enclosed in {}.

Function statement types:

- Declaration - Declares name and type of an ANH variable.
- Expression - Performs a calculation using data item and/or ANH variables, store the result to the hooked data item or an ANH variable. Math, Bitwise, and Logical (compare) operators are supported along with casting between integer data types.
- Flow Control Statements - ANH flow control keywords include ‘if’, ‘else if’, ‘else’, ‘goto’, and ‘label:’.
- Comments - Not evaluated, used for internal documentation of the ANH function.

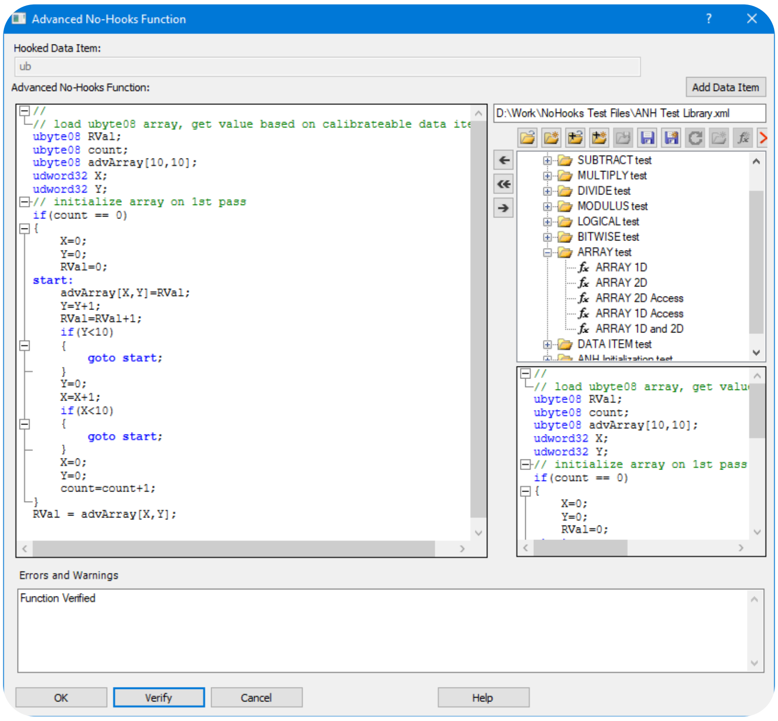
Example function:

```
// ACT_ENG GAIN and OFFSET
// data item gain function
//
float32 ACT_ENG_GAIN;
float32 ACT_ENG_OFFSET;
Measurements.ACT_ENG = Measurements.ACT_ENG * ACT_ENG_GAIN + ACT_ENG_OFFSET;
```

Organize Functions in Libraries and Edit in VISION

Users can edit or create new Advanced No-Hooks functions in VISION and organize the functions in libraries to save and share with other users.

- Edit Advanced No-Hooks functions
- Organize functions into libraries
- Preview functions
- Share libraries
- Replace generic variable names with VISION Data Item names
- Verify function before running No-Hooks
- Set initial values of variables



Advanced No-Hooks Function

Hooked Data Item: ub

Advanced No-Hooks Function:

```
//
// load ubyte08 array, get value based on calibrateable data ite
ubyte08 RVal;
ubyte08 count;
ubyte08 advArray[10,10];
udword32 X;
udword32 Y;
// initialize array on 1st pass
if(count == 0)
{
    X=0;
    Y=0;
    RVal=0;
start:
    advArray[X,Y]=RVal;
    Y=Y+1;
    RVal=RVal+1;
    if(Y<10)
    {
        goto start;
    }
    Y=0;
    X=X+1;
    if(X<10)
    {
        goto start;
    }
    X=0;
    Y=0;
    count=count+1;
}
RVal = advArray[X,Y];
```

D:\Work\NoHooks Test Files\ANH Test Library.xml

```
//
// load ubyte08 array, get valu
ubyte08 RVal;
ubyte08 count;
ubyte08 advArray[10,10];
udword32 X;
udword32 Y;
// initialize array on 1st pass
if(count == 0)
{
    X=0;
    Y=0;
    RVal=0;
```

Errors and Warnings

Function Verified

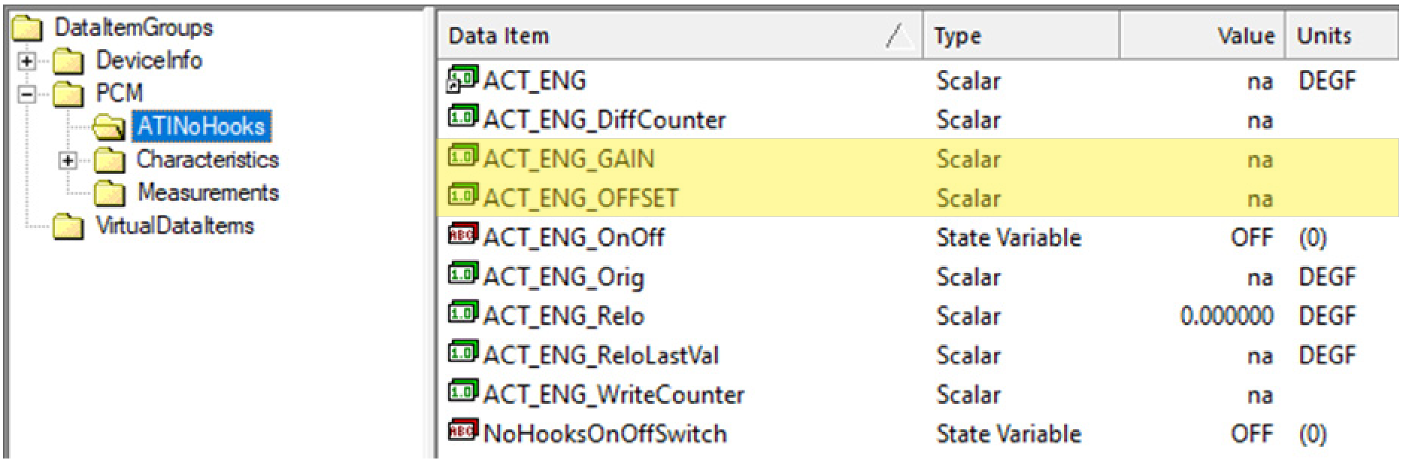
Use Existing Data Items as Variables

Create user functions such as gain with user calibratable gain factor. Advanced No-Hooks can assign a result to the hooked data item and may use any VISION data item as a factor in an equation.

```
//
// data item gain function
// Measurements.ub is the hooked data item
// Advanced No-Hooks functions can not assign to data items other than
// the hooked item that calls the function
//
ubyte08 gain;
Measurements.ub = Measurements.ub * gain;
```

Advanced No-Hooks Variables are VISION Data Items

Advanced No-Hooks variables are created as VISION data items, along with Base No-Hooks items. Currently all variables created in Advanced No-Hooks functions are located in No-Hooks RAM region.



Data Item	Type	Value	Units
ACT_ENG	Scalar	na	DEGF
ACT_ENG_DiffCounter	Scalar	na	
ACT_ENG_GAIN	Scalar	na	
ACT_ENG_OFFSET	Scalar	na	
ACT_ENG_OnOff	State Variable	OFF	(0)
ACT_ENG_Orig	Scalar	na	DEGF
ACT_ENG_Relo	Scalar	0.000000	DEGF
ACT_ENG_ReloLastVal	Scalar	na	DEGF
ACT_ENG_WriteCounter	Scalar	na	
NoHooksOnOffSwitch	State Variable	OFF	(0)

Advanced No-Hooks VISION Toolkits

Part Number	Name	Description
Toolkits		
152-xxxx	VISION No-Hooks Toolkits	✓ Adds the ability to bypass global ECU variables with a simple calibration parameter, all without access to the ECU source code. *Requires 152-0008, 152-0020, 152-0021*
152-0130	VISION Advanced No-Hooks Toolkit	✓ Adds the ability to bypass global ECU variables with user defined programmable functions, all without access to the ECU source code nor the need for a compiler. *Requires a specific VISION 152-xxxx No-Hooks Toolkit, where xxxx is the CPU type* *Currently only the TriCore (using TRICORE16 No-Hooks), PowerPC (using BOOKE No-Hooks), RH850, and VLE Processor types are supported*

ATI ACCURATE TECHNOLOGIES

sales@accuratetechnologies.com
www.accuratetechnologies.com

V1 - Advanced No-Hooks

DATA ACQUISITION • ECU CALIBRATION • ECU RAPID PROTOTYPING • ECU INTERFACES • CAN BUS PRODUCTS • NETWORK SOLUTIONS • TEST CELL MEASUREMENT

Information is provided on an “as is” basis and could include technical, typographical or other errors. Accurate Technologies Inc. makes no warranties, representations, or guarantees of any kind, express or implied, including but not limited to, accuracy, or completeness of the information, content, and products.