life.augmented

# STM32H5 Device Lifecycle Managment

# Introduction

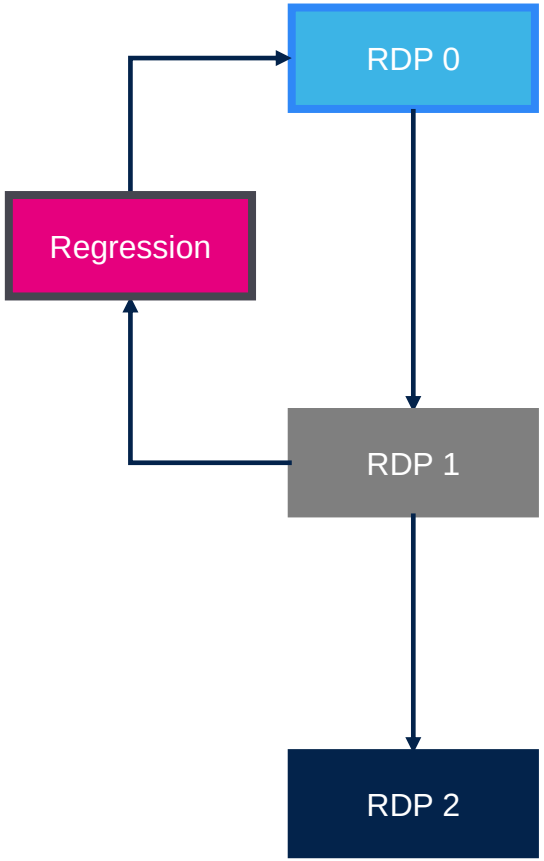
- STM32H5 implements a new mechanism for the protection of the code stored in internal flash
- Let’s see the impact of this change

# Reminder of RDP protection on legacy STM32

- RDP 0 : Open state dedicated to development

- RDP 1
  - Firmware in flash is protected from readout
  - Debugger can attach and read ram content
  - Possible regression to RDP0 with automatic flash erase
  - State mostly used because of this regression capability

- RDP2
  - No debugger access, no possible regression




life.augmented

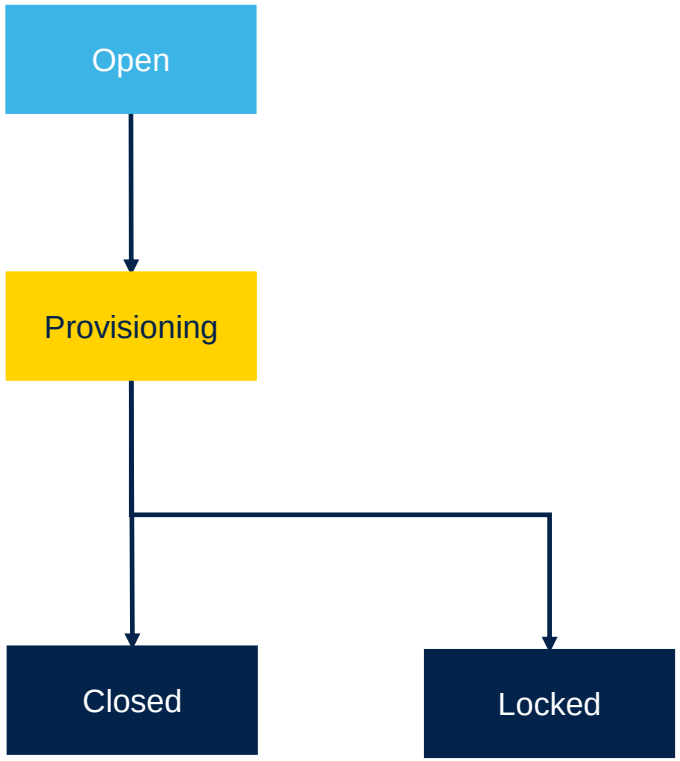
# STM32H5 evolution

- RDP levels replaced by PRODUCT_STATE
  - Option byte in both cases
  - RDP values fixed for RDP 0 (0xAA) and RDP 2 (0xCC). All other values mean RDP 1
  - PRODUCT_STATE have fixed value for each state. No default

- Regression control mechanism : Debug Authentication
  - JTAG dedicated access point
  - ADAC protocol defined by ARM
  - Password used for regression
  - Certificate used when enabling TrustZone of Cortex M33 (Case not detailed here)

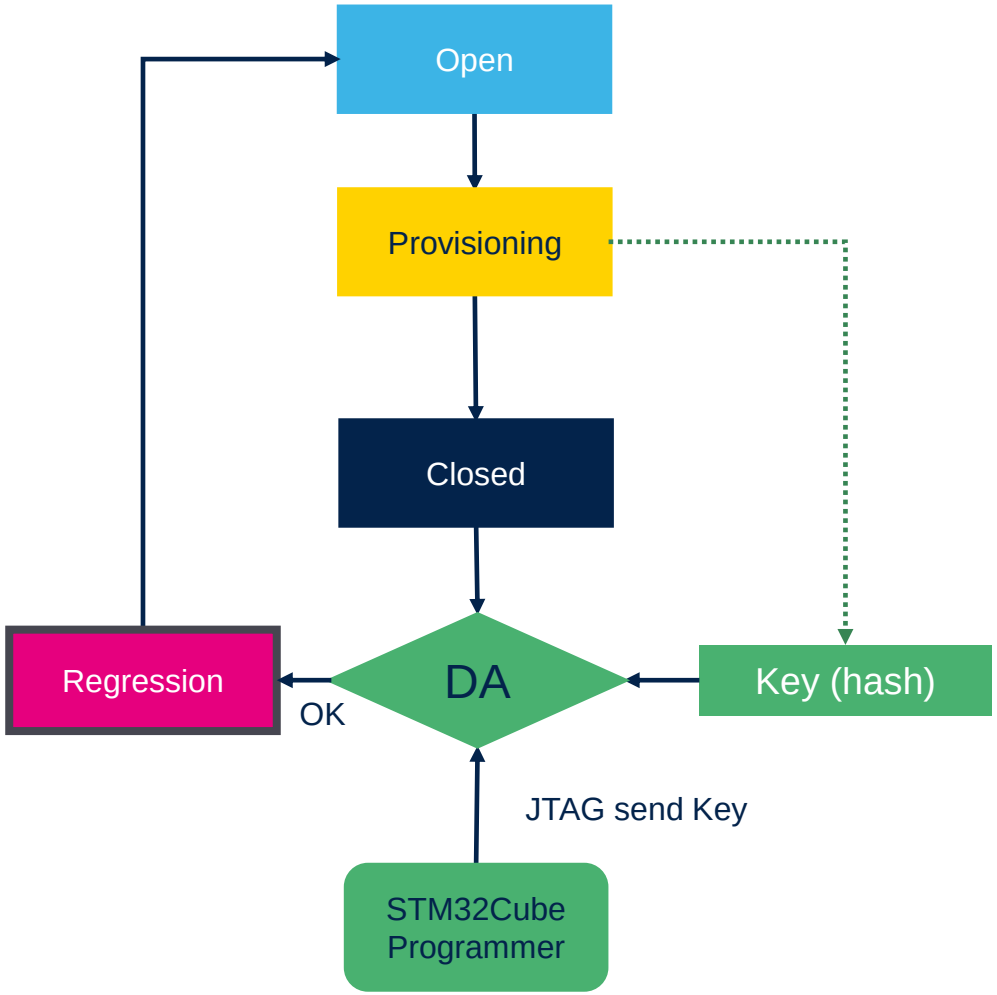
# Product State

- Open state dedicated to development
- Provisioning allows transmitting specific file containing keys and data to be provisioned
- Closed and Locked are used in the field to protect device.
  - Closed state allows possible regression
  - Locked state is definitive



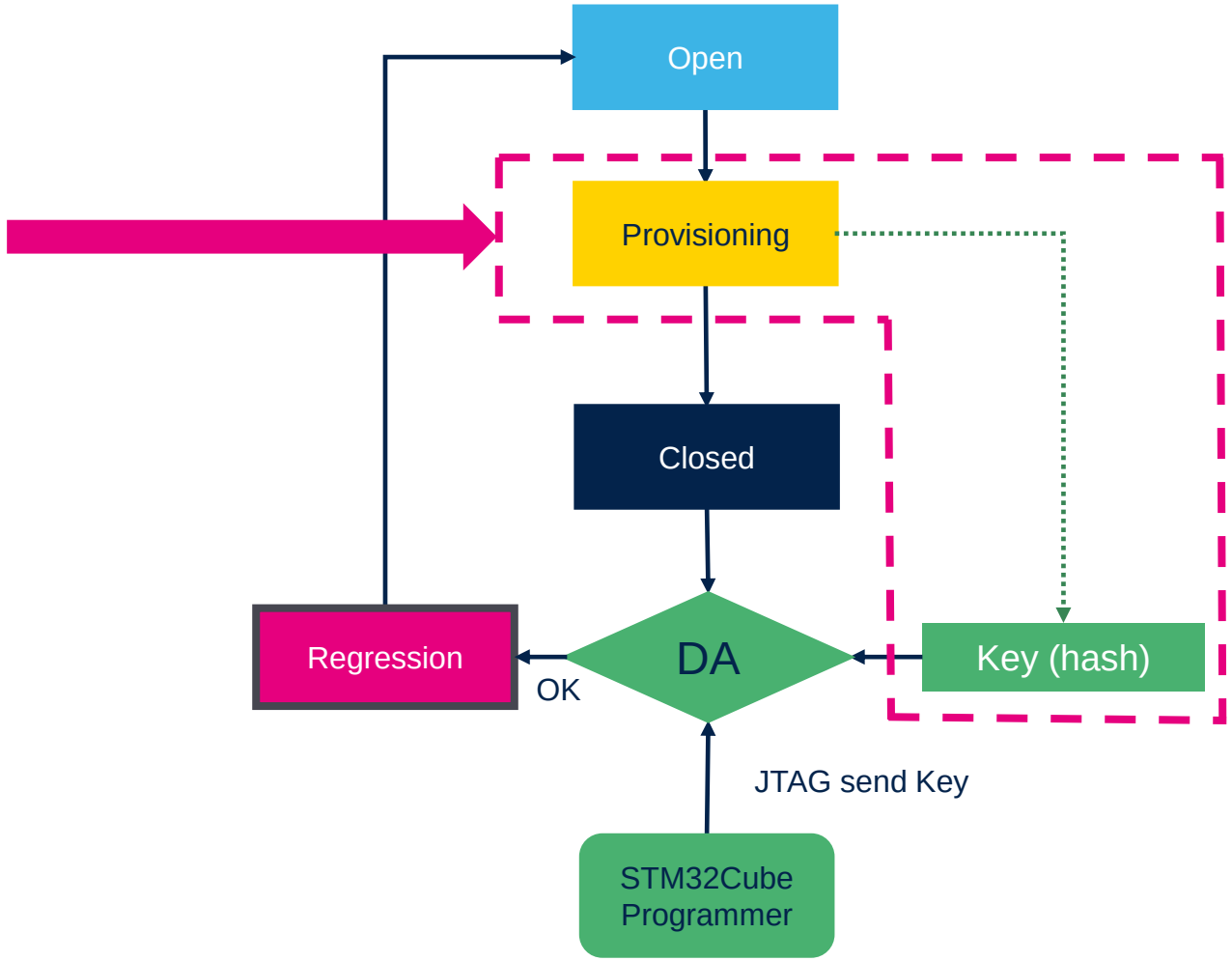
# Regression controlled by Debug Authentication

- Firmware can be flashed in open state
- Provisioning is used in production to transmit auth key to the secure storage
- Close device : no more debug access
- Field return : use key to open the device securely through JTAG/SWD interface using dedicated access point. Flash content will be erased.



# Impact on production process

New step required to have the capability to perform regression




life.augmented

# Let’s see how it works

- Next part will show you how to ensure the regression capability when flash code protection is activated on STM32H5
- We will use the tools provided in STM32 ecosystem
  - STM32TrustedPackageCreator
  - STM32CubeProgrammer

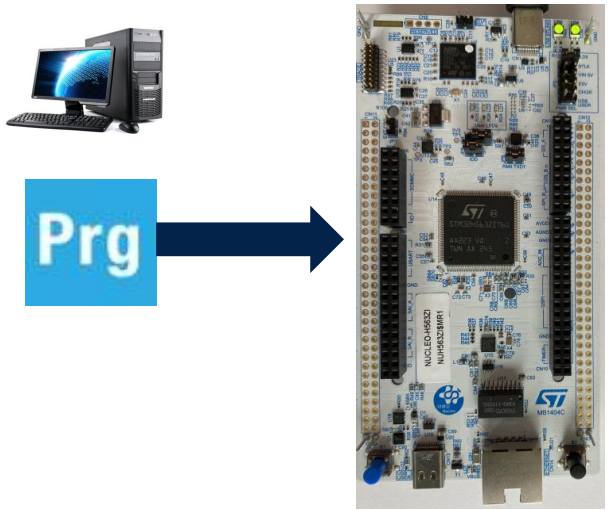
# Hands-on purpose

- Application firmware with TZ disabled
- Use case could be compared to RDP usage in STM32F4
- Simple LED blink application
- When TZ is disabled we use a password for debug authentication
- This demo shows the steps to provision this password in order to enable the regression

# Debug Authentication Hands-on Steps

- TZEN disabled
  - Step 1 : Create provisioning file (obk file)

  - Step 2 : Flash blinking LED binary
  - Step 3 : Move to **PROVISIONING** state
  - Step 4 : Provision the obk file to the board
  - Step 5 : Move to **CLOSED** state
  - Step 6 : Back to original state: regression



# Demo sum-up

In this short demo we have seen how to provision a password on STM32H5 to enable secure regression capability

# Take away

- STM32H5 replaces RDP by PRODUCT_STATE
- STM32H5 introduces Debug Authentication feature to control regression with a password
- The provisioning of this password adds a step in production
- By disabling debug interface, this new mechanism increases the robustness of the flash protection
- The Debug Authentication offers even more capabilities when TrustZone is enabled. This will be presented in STM32H5 security workshop

life.augmented

# Debug Authentication hands-on TZ disabled

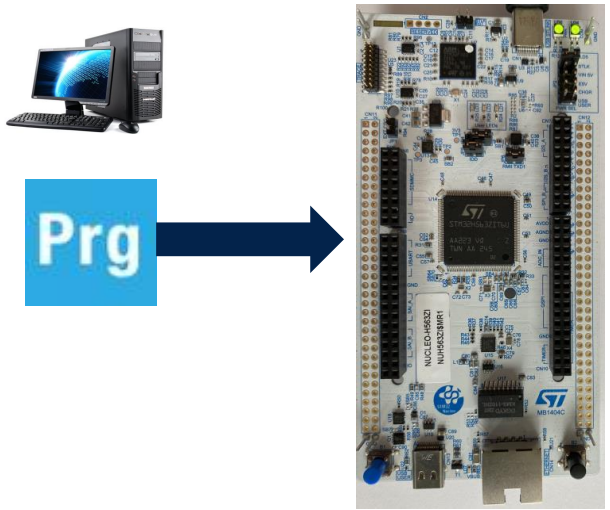
# Hands-on purpose

- Application firmware with TZ disabled
- Use case could be compared to RDP usage in STM32F4
- Simple LED blink application
- When TZ is disabled we use a password for debug authentication
- This demo shows the steps to provision this password in order to enable the regression

# Debug Authentication Hands-on Steps

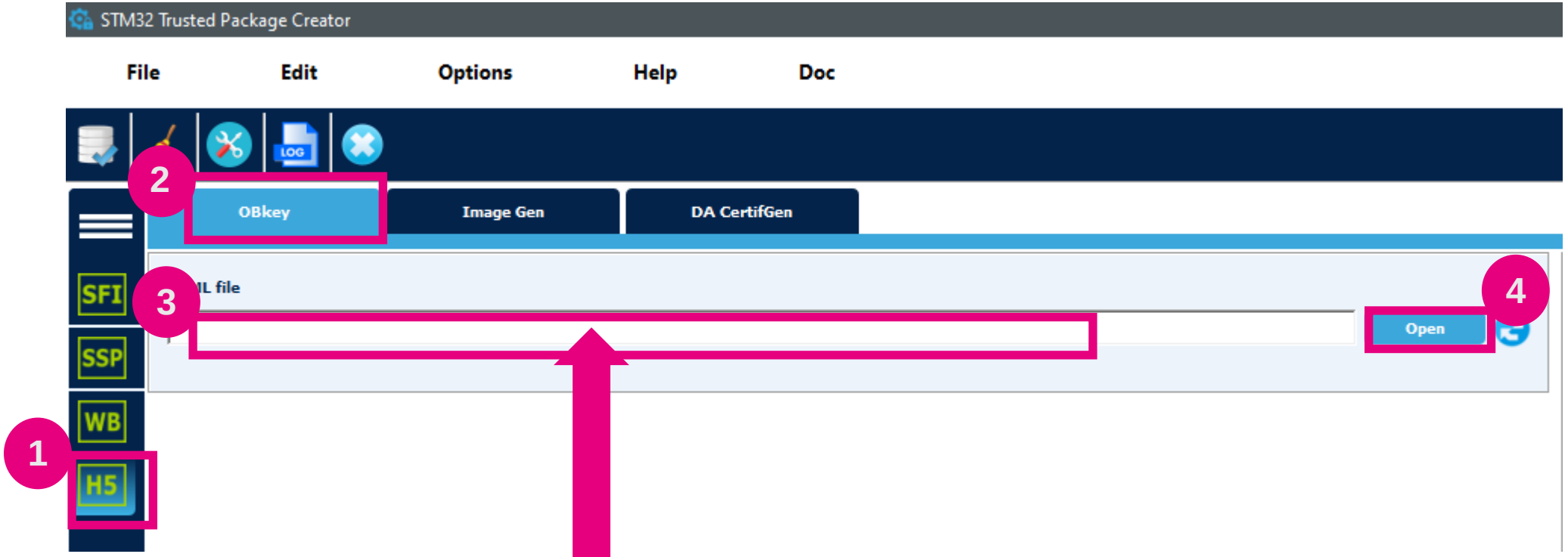
- TZEN disabled
  - Step 1 : Create provisioning file (obk file)

  - Step 2 : Flash blinking LED binary
  - Step 3 : Move to **PROVISIONING** state
  - Step 4 : Provision the obk file to the board
  - Step 5 : Move to **CLOSED** state
  - Step 6 : Back to original state: regression



# Step 1 : Create provisioning file (obk file)

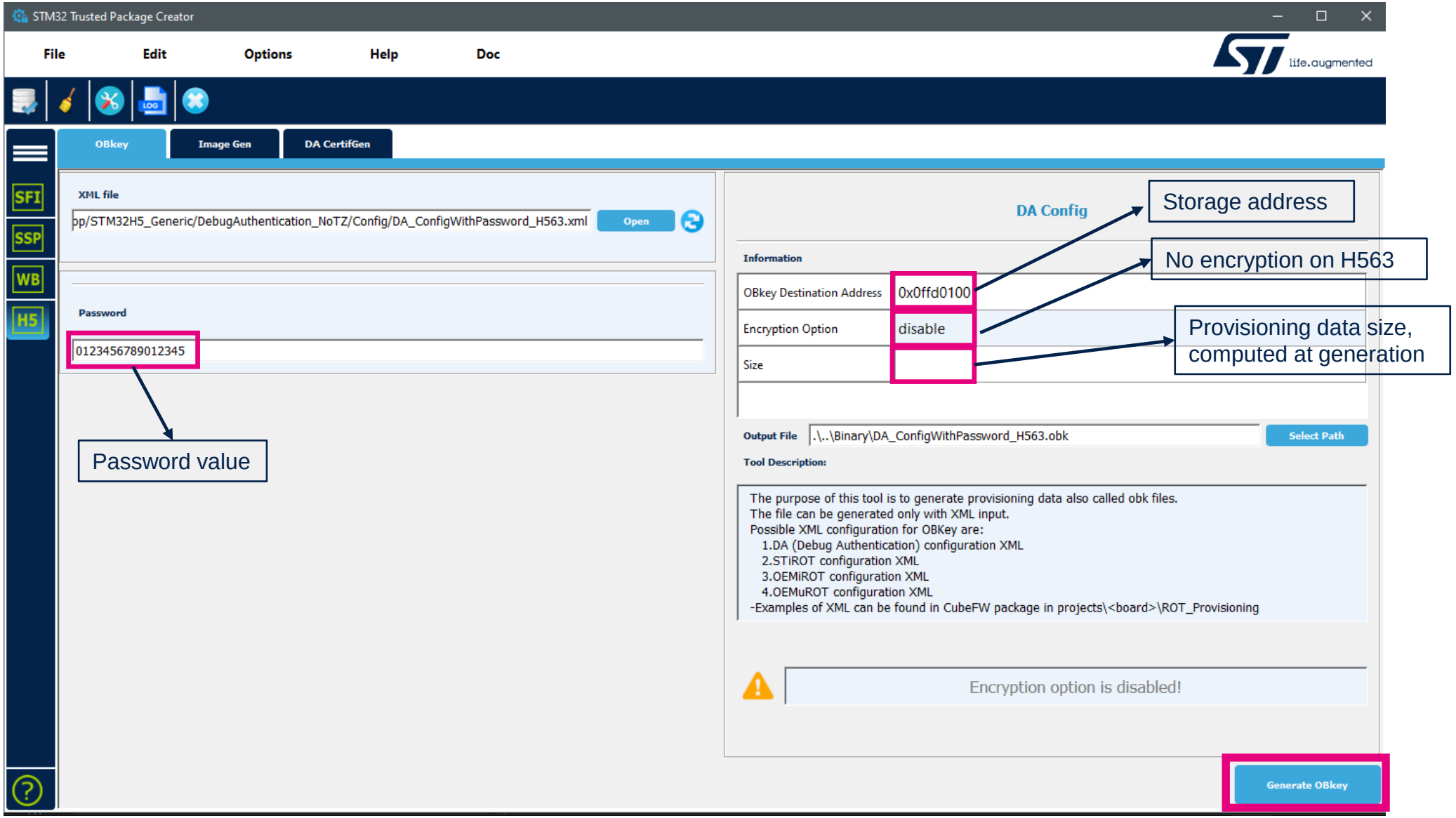
- Launch STM32Trusted Package creator
  Select DebugAuthentication_NoTZ\Config\DA_ConfigWithPassword_H563




…\DebugAuthentication_NoTZ\Config\DA_ConfigWithPassword_H563.xml

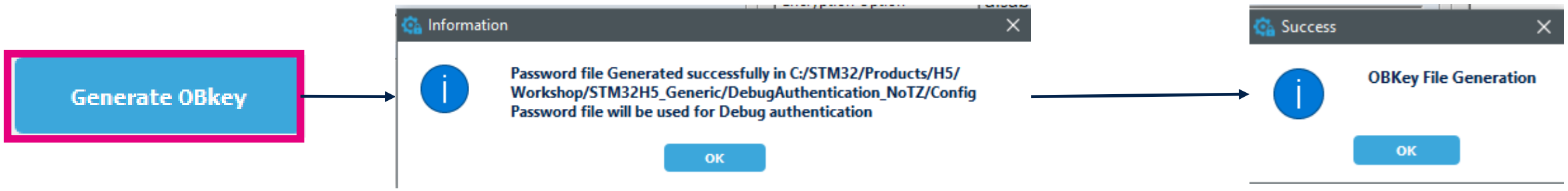
life.augmented

# Step 1 : Create provisioning file (obk file)




life.augmented

# Step 1 : Create provisioning file (obk file)



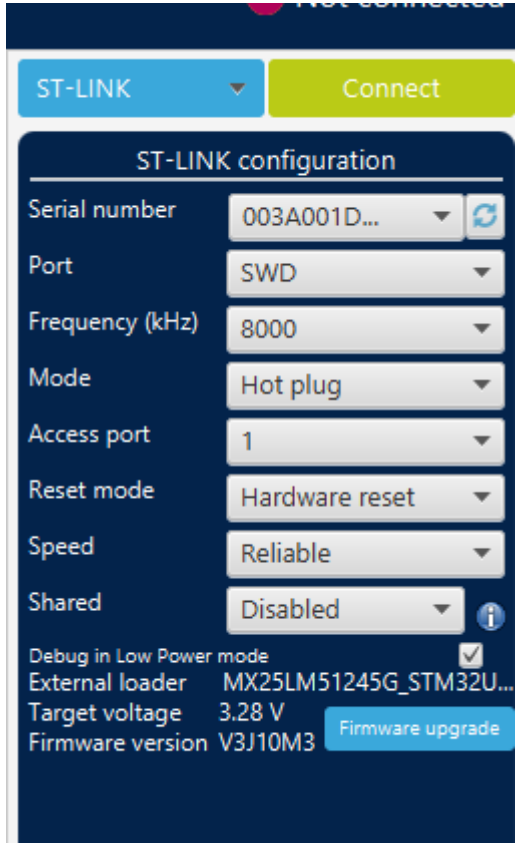

- Output files
  - OBK provisioning file: **Binary**\DA_ConfigWithPassword_H563.obk
  - AssociatedFormated password file : **Config**\password.bin

# Debug Authentication Hands-on Steps

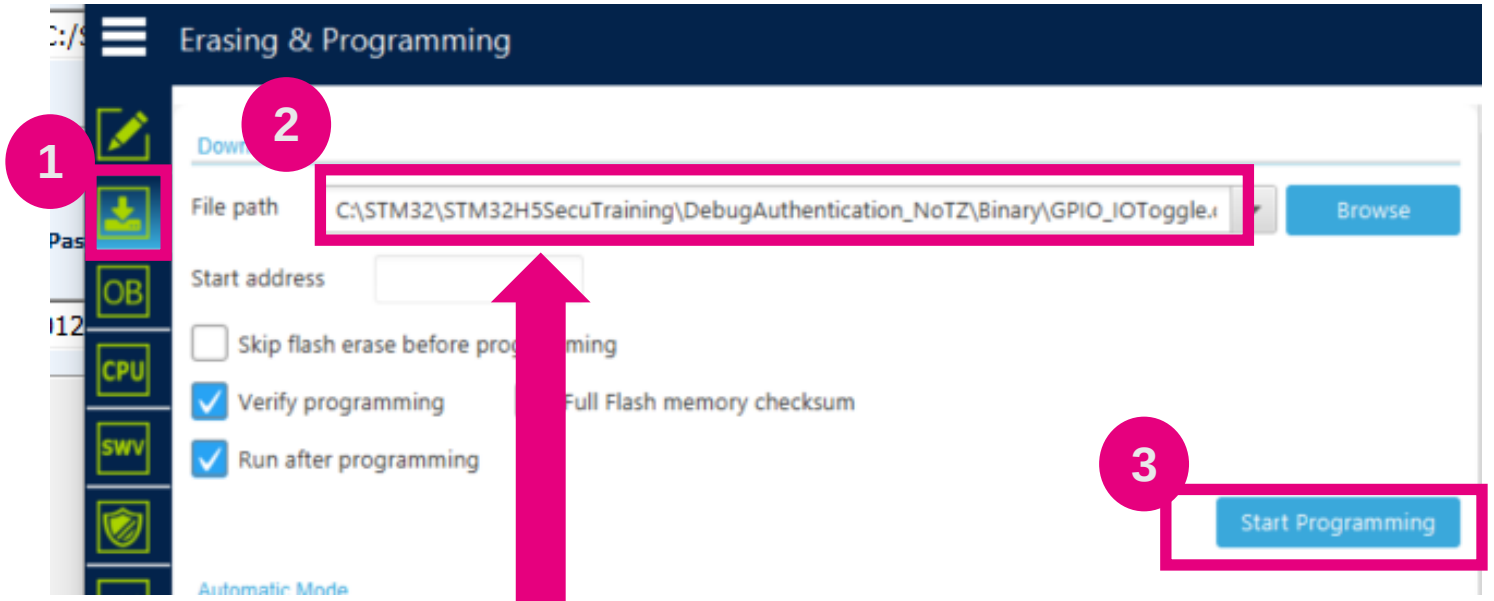
- TZEN disabled
  - ~~Step 1 : Create provisioning file (obk file)~~
  - Step 2 : Flash blinking LED binary
  - Step 3 : Move to **PROVISIONING** state
  - Step 4 : Provision the obk file to the board
  - Step 5 : Move to **CLOSED** state
  - Step 6 : Back to original state: regression

# Step 2 : Flash blinking LED binary

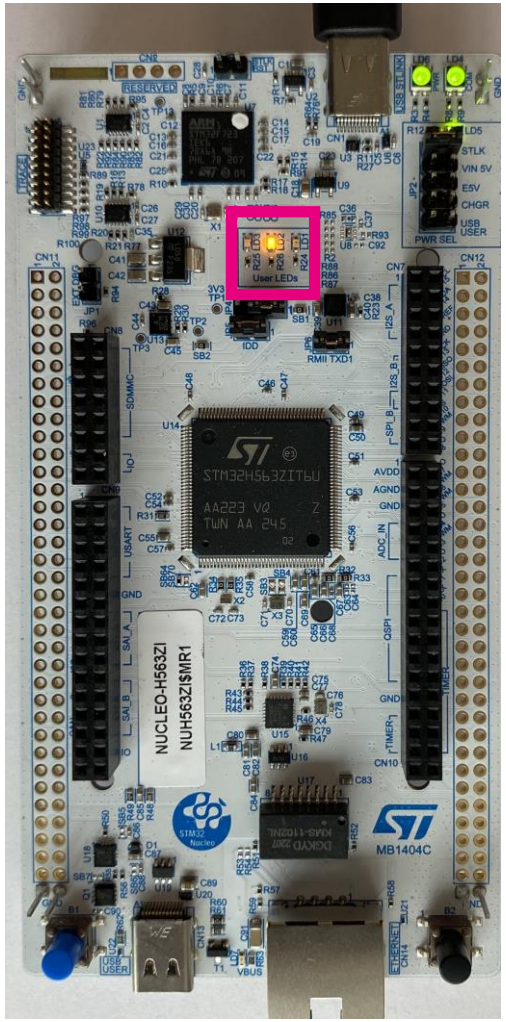
- Open STM32CubeProgrammer
- Connect to target in hotplug mode



# Step 2 : Flash blinking LED binary




…\DebugAuthentication_NoTZ\Binary\GPIO_IOToggle.out

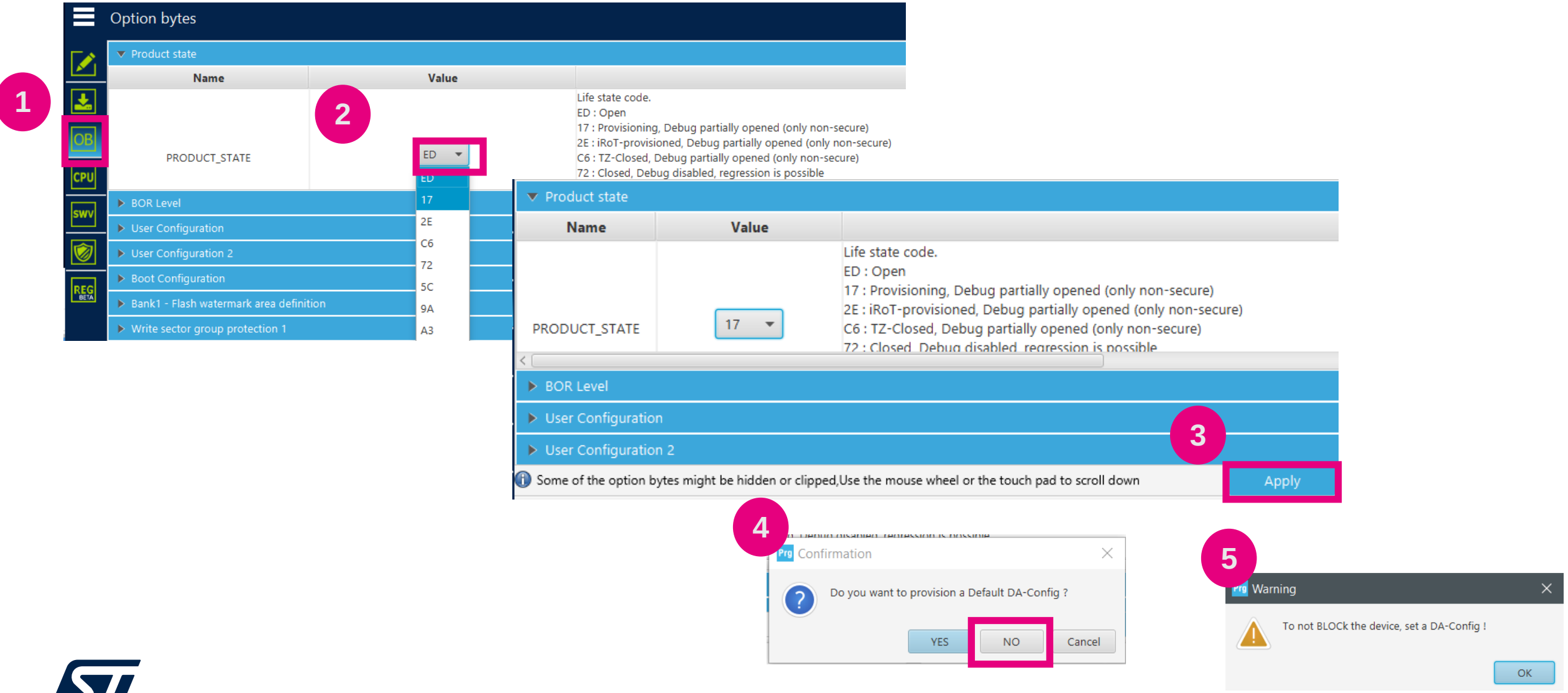



life.augmented

# Debug Authentication Hands-on Steps

- TZEN disabled
  - ~~Step 1 : Create provisioning file (obk file)~~
  - ~~Step 2 : Flash blinking LED binary~~
  - Step 3 : Move to **PROVISIONING** state
  - Step 4 : Provision the obk file to the board
  - Step 5 : Move to **CLOSED** state
  - Step 6 : Back to original state: regression

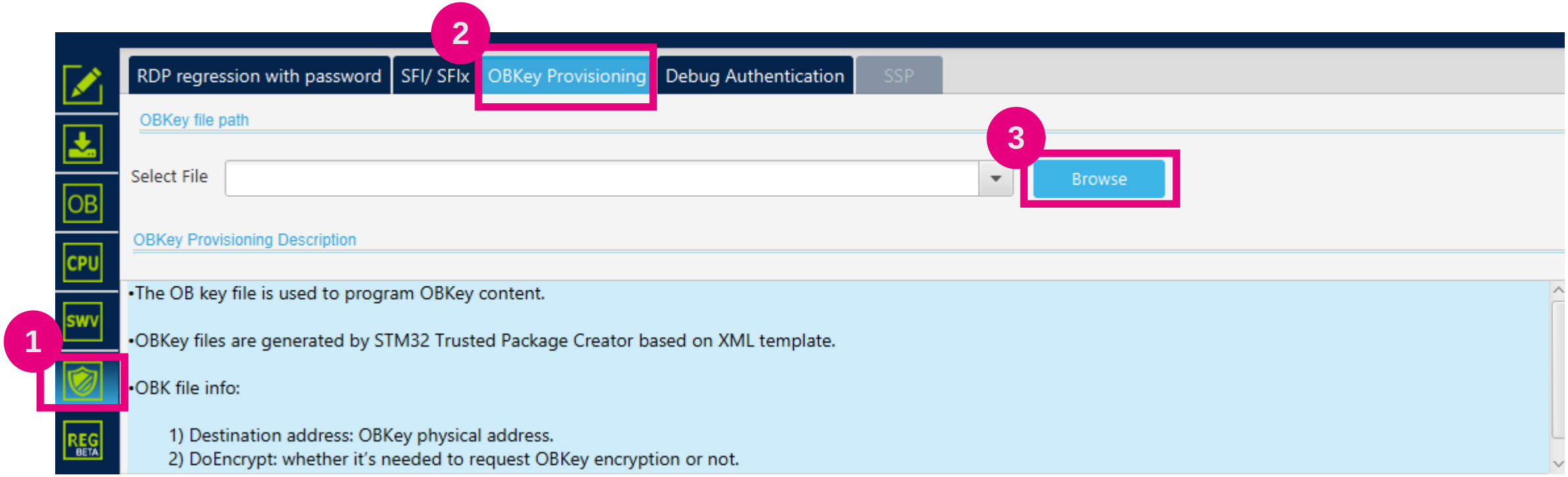
# Step 3 : Move to **PROVISIONING** state



# Debug Authentication Hands-on Steps

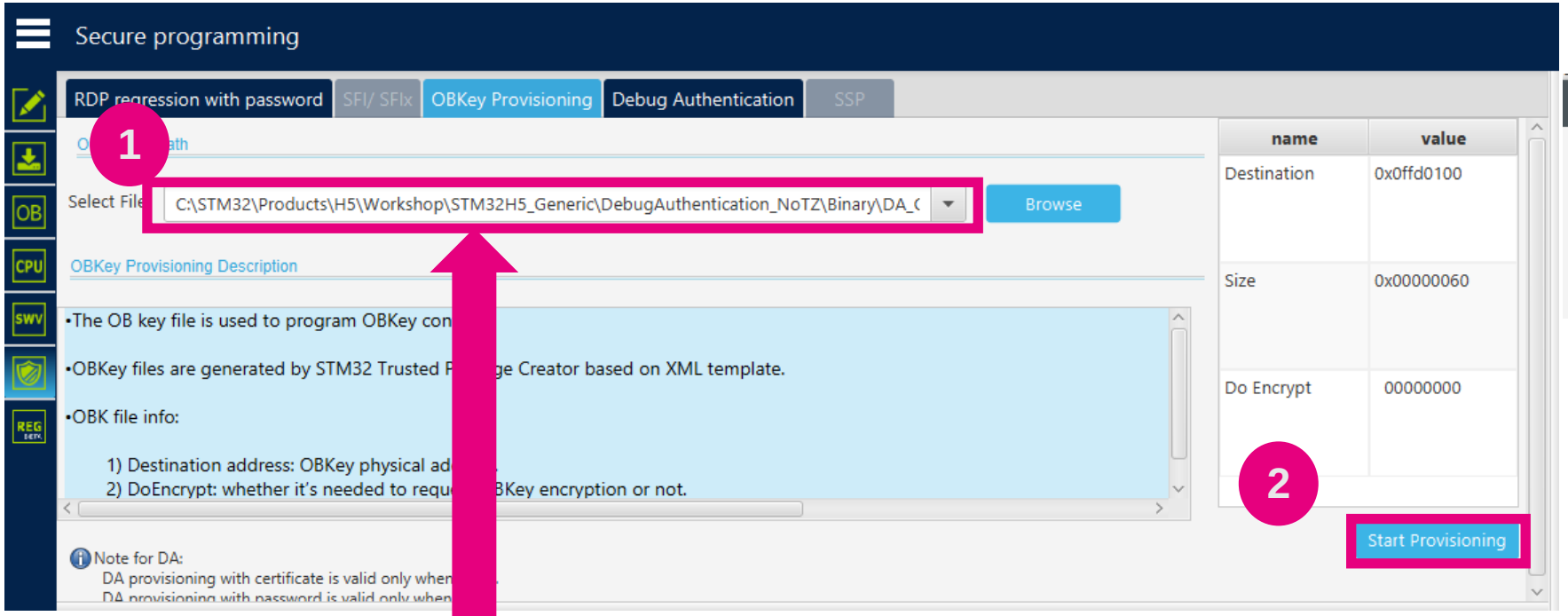
- TZEN disabled
  - ~~Step 1 : Create provisioning file (obk file)~~
  - ~~Step 2 : Flash blinking LED binary~~
  - ~~Step 3 : Move to **PROVISIONING** state~~
  - Step 4 : Provision the obk file to the board
  - Step 5 : Move to **CLOSED** state
  - Step 6 : Back to original state: regression

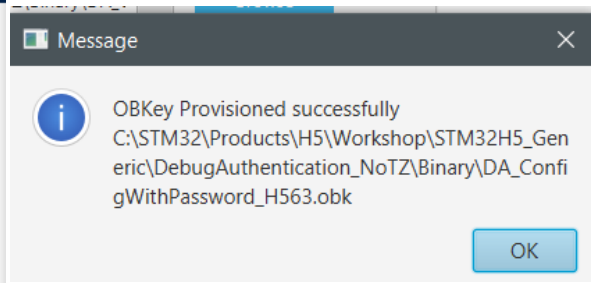
# Step 4 : Provision the obk file to the board

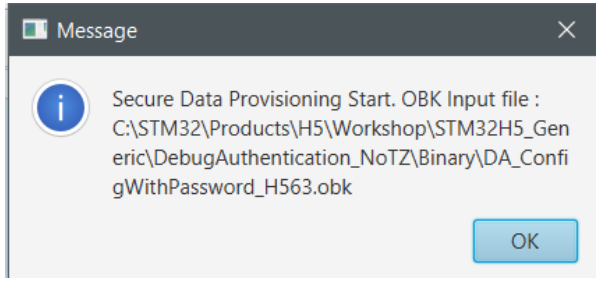



life.augmented

# Step 5 : DA provisioning



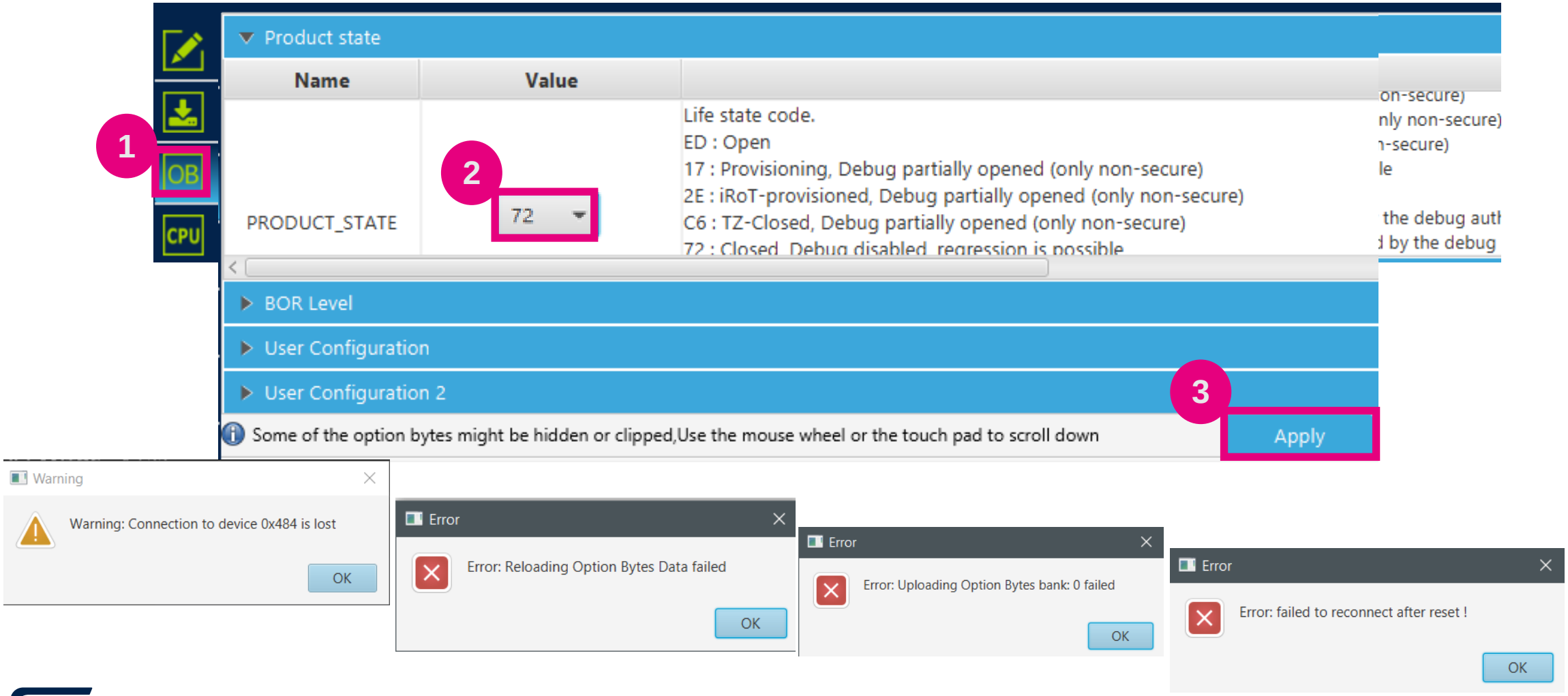

…\DebugAuthentication_NoTZ\Binary\DA_ConfigWithPassword_H563.obk

# Debug Authentication Hands-on Steps

- TZEN disabled
  - ~~Step 1 : Create provisioning file (obk file)~~
  - ~~Step 2 : Flash blinking LED binary~~
  - ~~Step 3 : Move to **PROVISIONING** state~~
  - ~~Step 4 : Provision the obk file to the board~~
  - Step 5 : Move to **CLOSED** state
  - Step 6 : Back to original state: regression

life.augmented

# Step 5 : Move to **CLOSED** state




life.augmented

# Move to **CLOSED** state

- Device is now closed.
- No JTAG DEBUG connection is possible
- Firmware runs normally
- **The device is ready to go to the field**

# STM32H5 Security
# Debug Authentication Provisioning

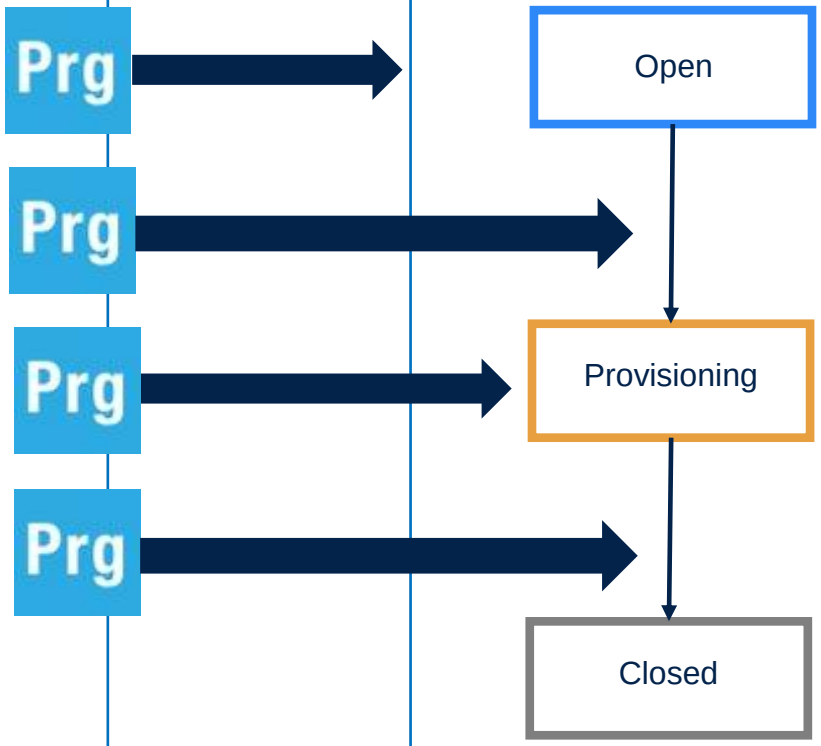
TZEN=0

Download LED blinking App

Change state to Provisioning

Provision password

Change state to Closed



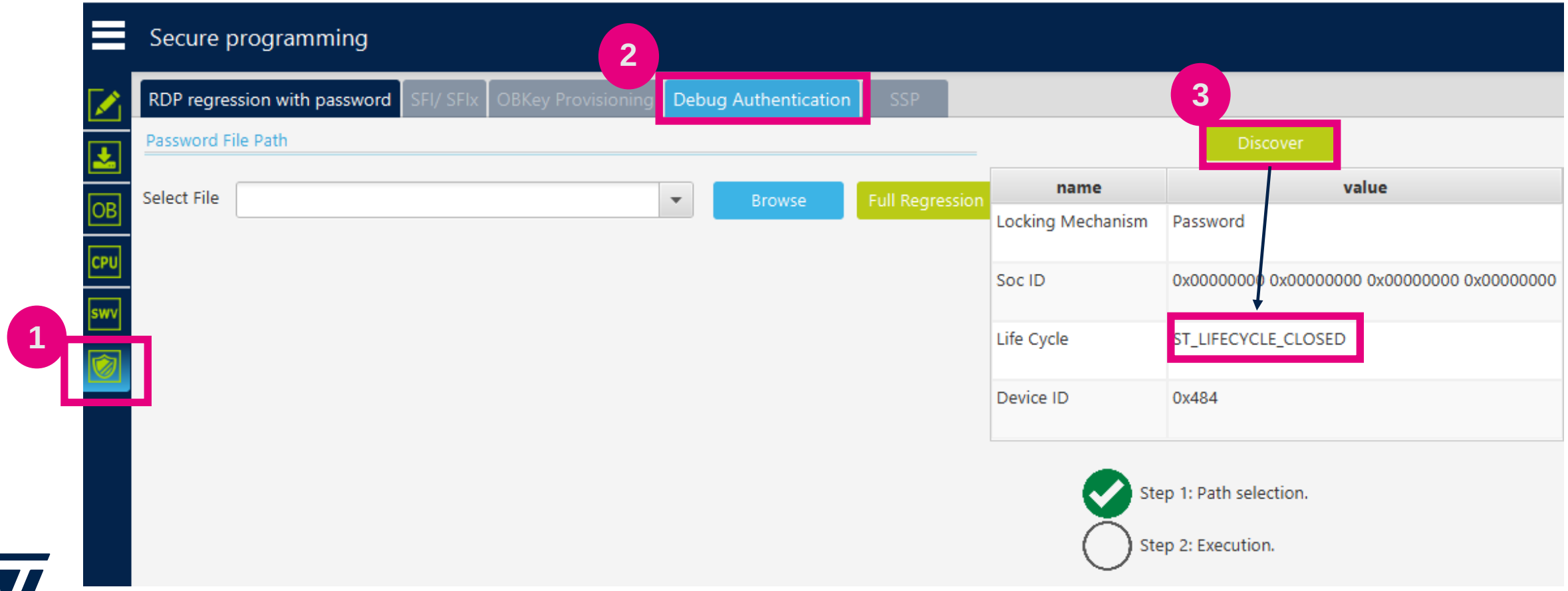

# Debug Authentication Hands-on Steps

- TZEN disabled
  - ~~Step 1 : Create provisioning file (obk file)~~
  - ~~Step 2 : Flash blinking LED binary~~
  - ~~Step 3 : Move to **PROVISIONING** state~~
  - ~~Step 4 : Provision the obk file to the board~~
  - Step 5 : ~~Move to **CLOSED** state~~
  - Step 6 : Back to original state: regression

life.augmented

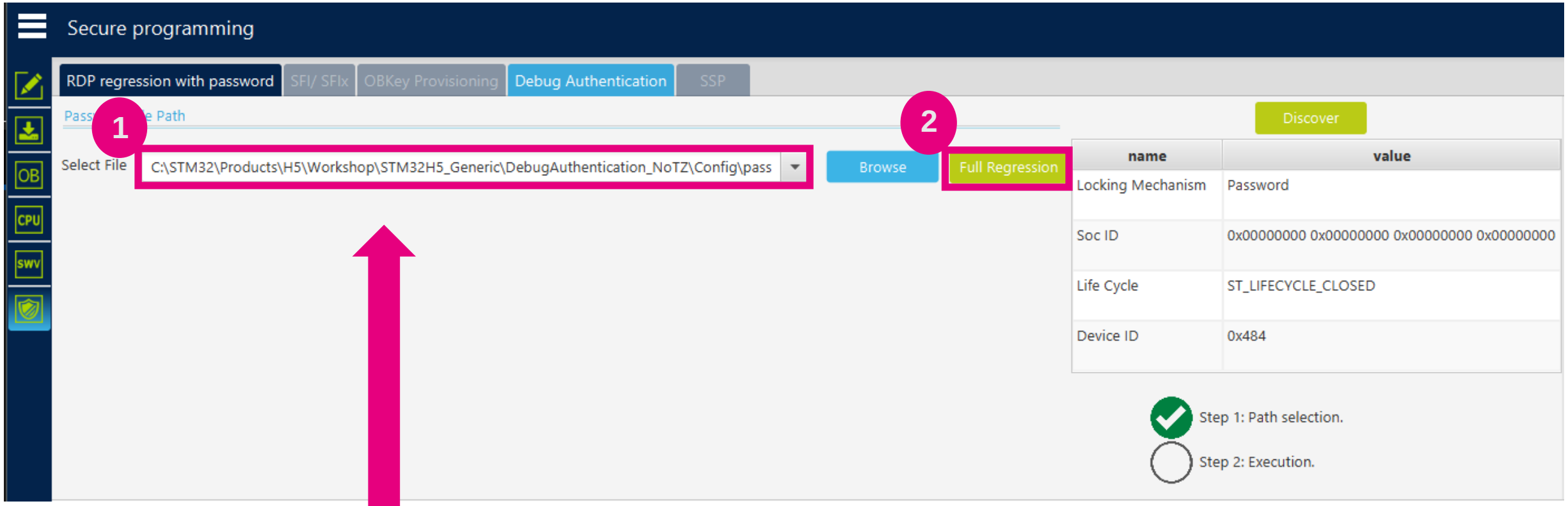
# Step 6 : Back to original state: regression

**First disconnect the programmer !**




life.augmented

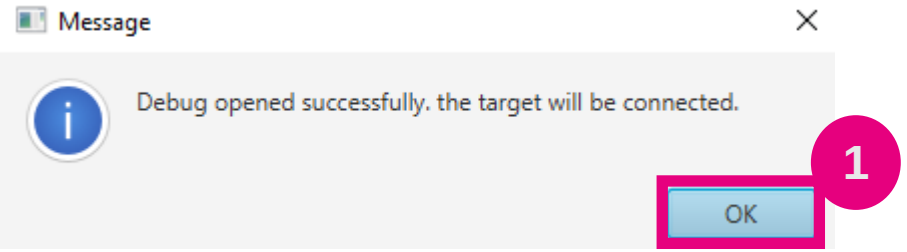
# Step 6 : Back to original state: regression



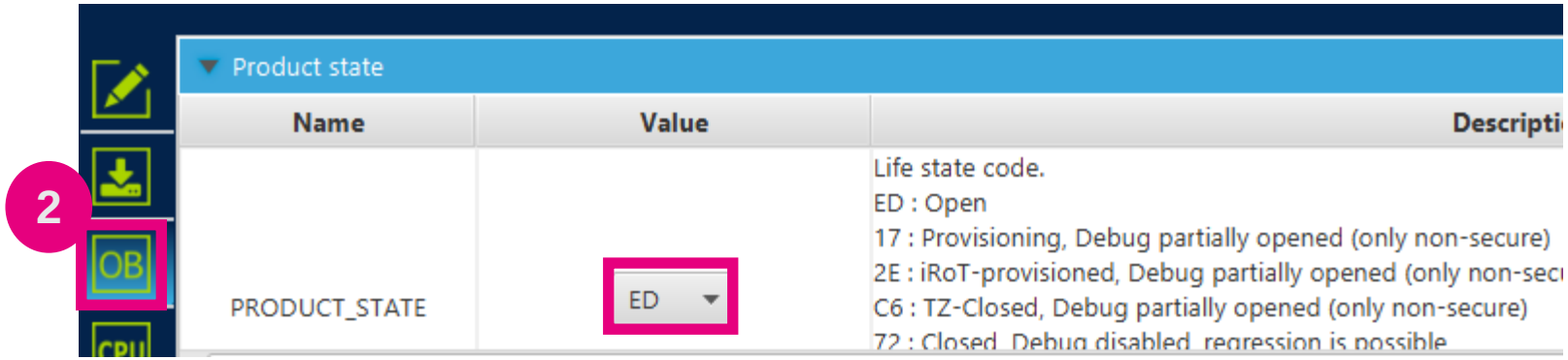

…\DebugAuthentication_NoTZ\Config\password.bin

life.augmented

# Step 6 : Back to original state: regression




**Check product state after regression : It is open and LED is not blinking as flash is erased during the regression**



# STM32H5 Security
# Debug Authentication for regression

TZEN=0

Regression

Prg

Closed

Regression

Open

life.augmented

# DA Hands-on no TZ take away

- We learned
  - How to create a password provisioning file (obk) with STM32TrustedPackageCreator
  - How to provision this obk file to the STM32H5 with STM32CubeProgrammer
  - How this provisioning is done in the production flow
  - How to use debug authentication interface to do a regression of  the device