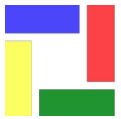
MTS3 Training



OpenATE Inc.

The open solution for IC tester

Environment Setup

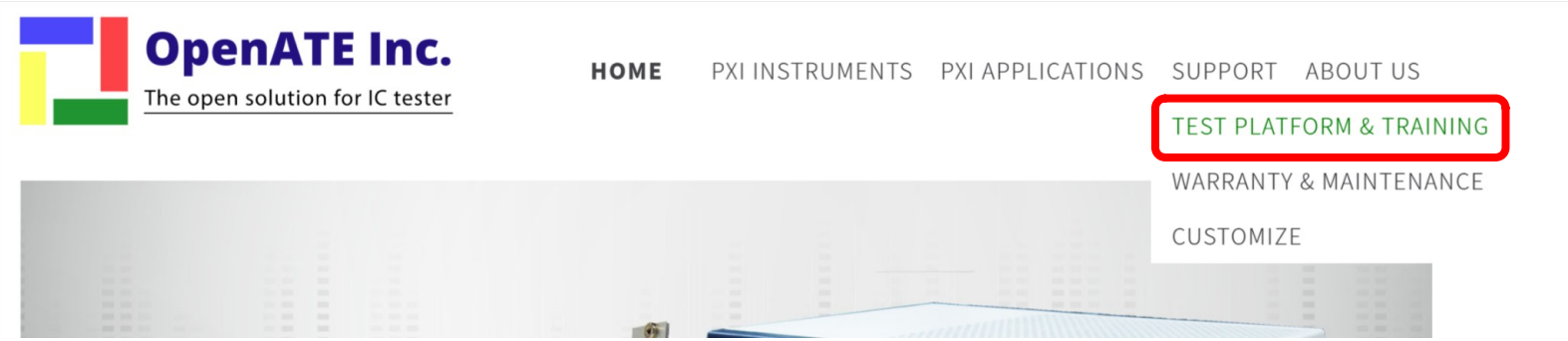
Preparing for using MTS3

- User should prepare the following equipment:
  - PE card
  - PXI chassis
  - C/C++ program (Visual C++, Version:2010~2013)
  - Adobe Reader V11.0
  - PC or NB
  - 68 pin vhdci cable
  - DUT board

Visual C++
Version:2013 (Standard)

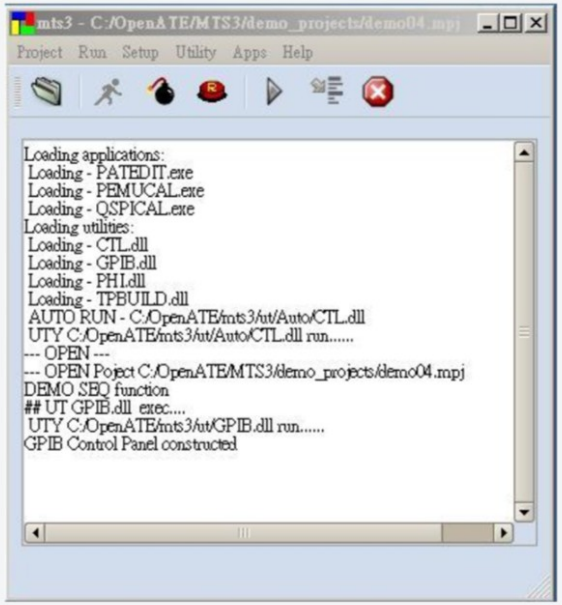
Download and Install MTS3

- Download MTS3 install file from OpenATE official web
  - http://www.openate.com/



MTS3 TEST PLATFORM

MTS3 is a compact software platform that includes test program development, test operation and debug functionalities. Based on open architectural design, this platform can run on any hardware on Windows and can easily add new instruments API and utilities.



DOWNLOAD

1 Data sheet

2 Install

CONTACT US

1. Data sheet:

User can download MTS3 data sheet, including simple using instructions.

2. Install:

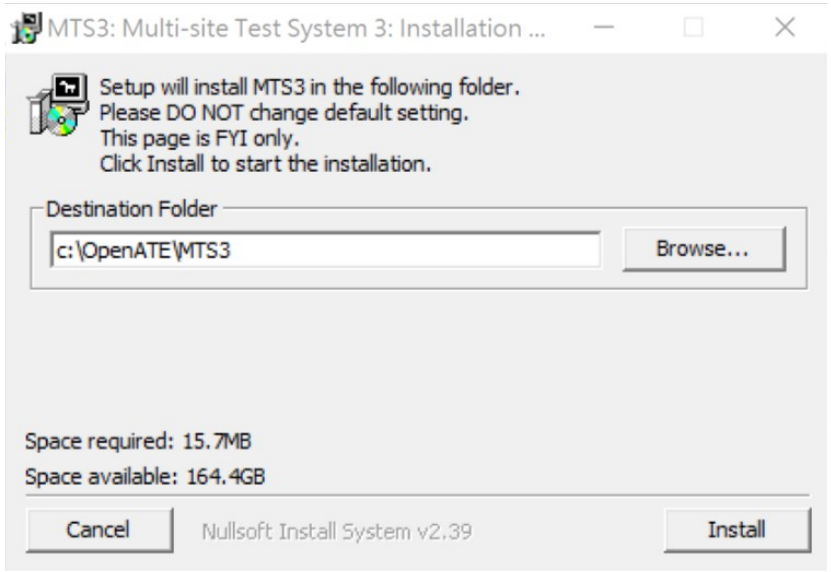
Download MTS3 install file from this link.

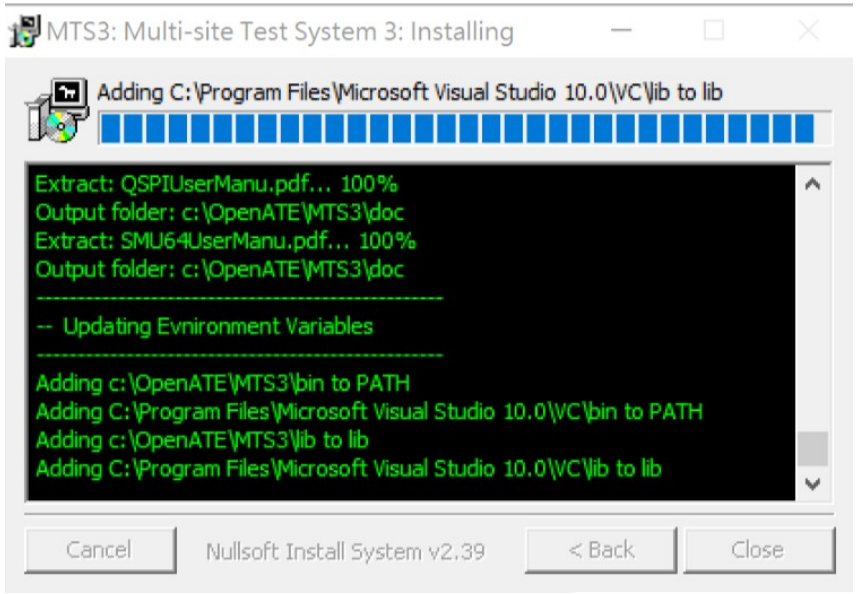
OpenATE Inc.

The open solution for IC tester

- Step 1:

Click and install mts3_VISAinstall201xxxxx.exe





- Step 2

Install Compiler "**Visual C++ 2013 Express**"

Complete the following instructions after installing the VC++:

(a) Add "C:\Program Files (x86)**Adobe\Reader11.0**\Reader" to PATH.

- Step 3: Download "**NI-VISA**" and install

Introduction to MTS3

Documents

- MTS3 user manual
  - C:\OpenATE\MTS3\doc\mts3_user.manual.pdf
  - C:\OpenATE\MTS3\doc\MTS3_install_readme.pdf

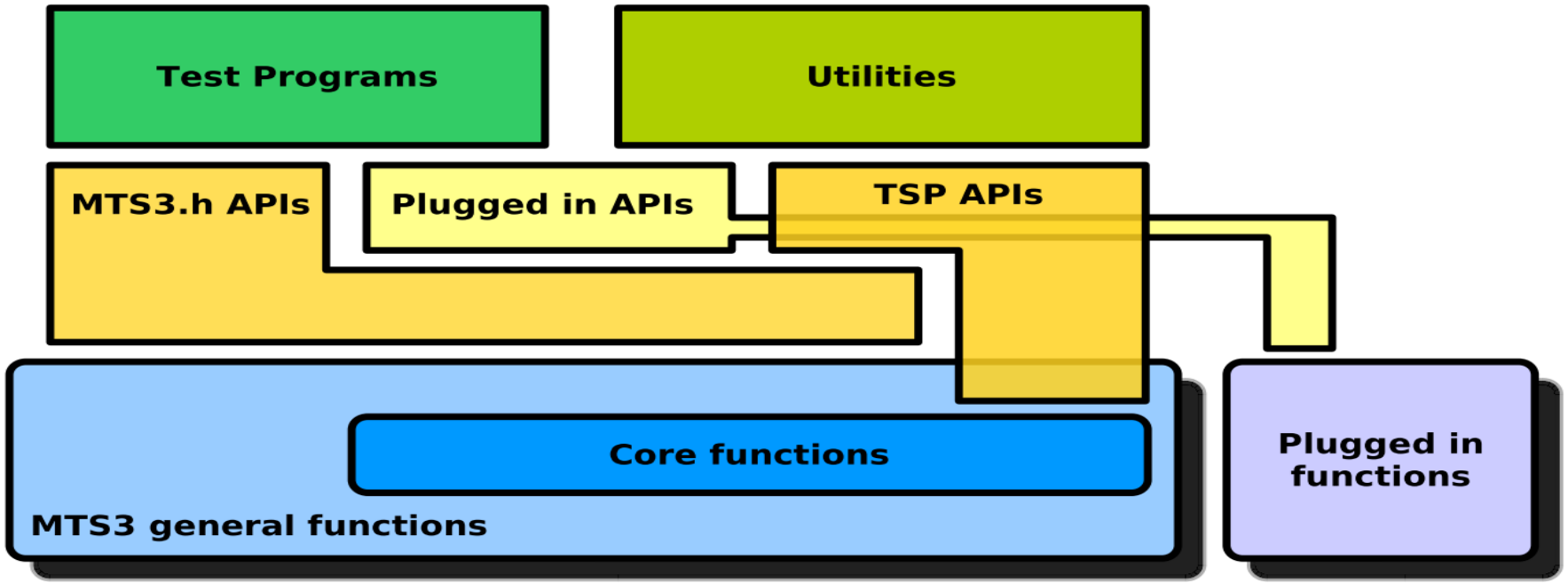
Features

- ATE software platform which can run with any hardware on Windows.
- Reconfigurable for new instruments API
- C/C++ development environments
- Integrated debugging capabilities:
  - DATALOG , DEBUG, OVERRIDE, STOP ON_FAIL
- Loop test capability
- Summary report ready

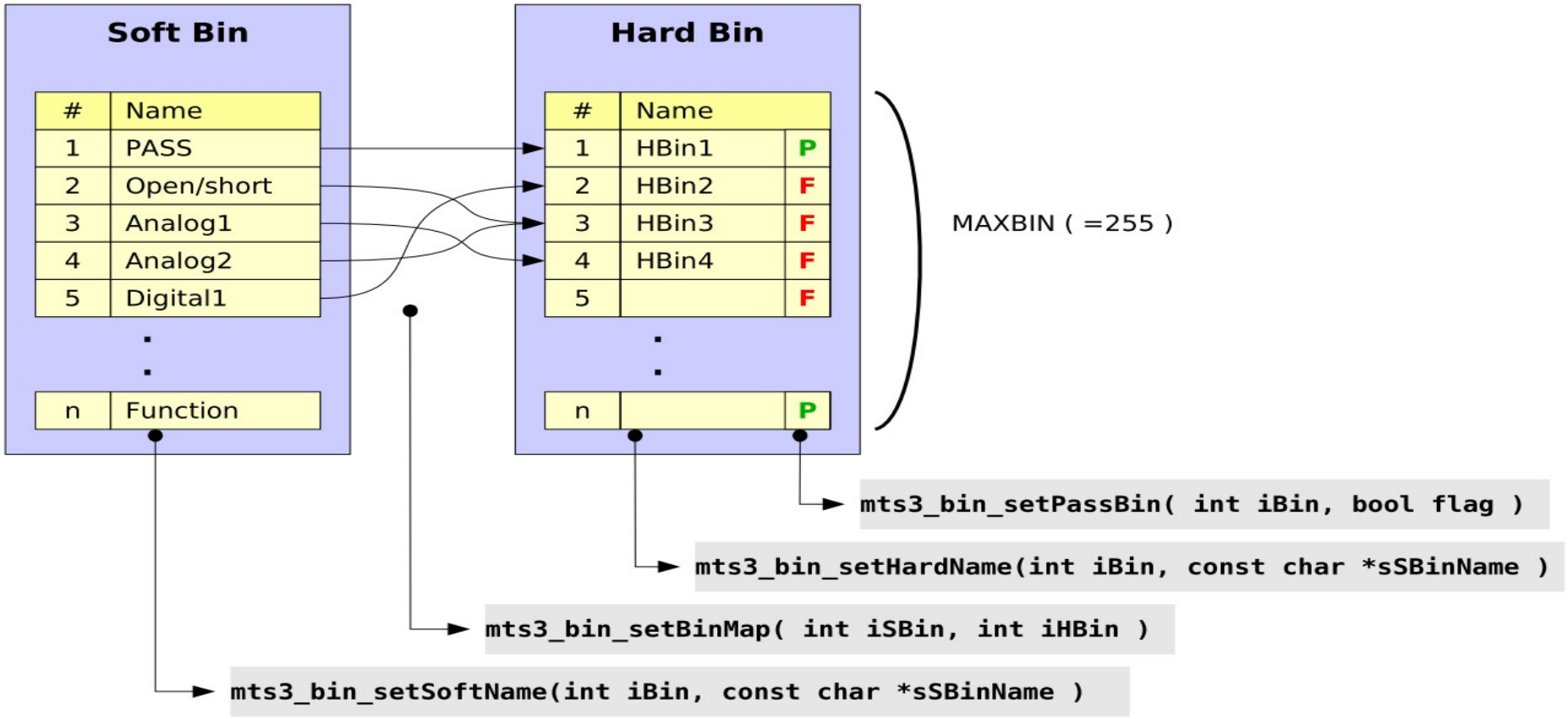
Features (cnt.)

- Datalog to file or STDF data format
- Multi-sites testing capability
  - No need to modify test program
  - One single test program for multi-sites testing
- Reconfigurable prober / handler interface.
- Easy to connect any type of prober / handler form any peripheral supplier

Software Architecture



Bin related functions



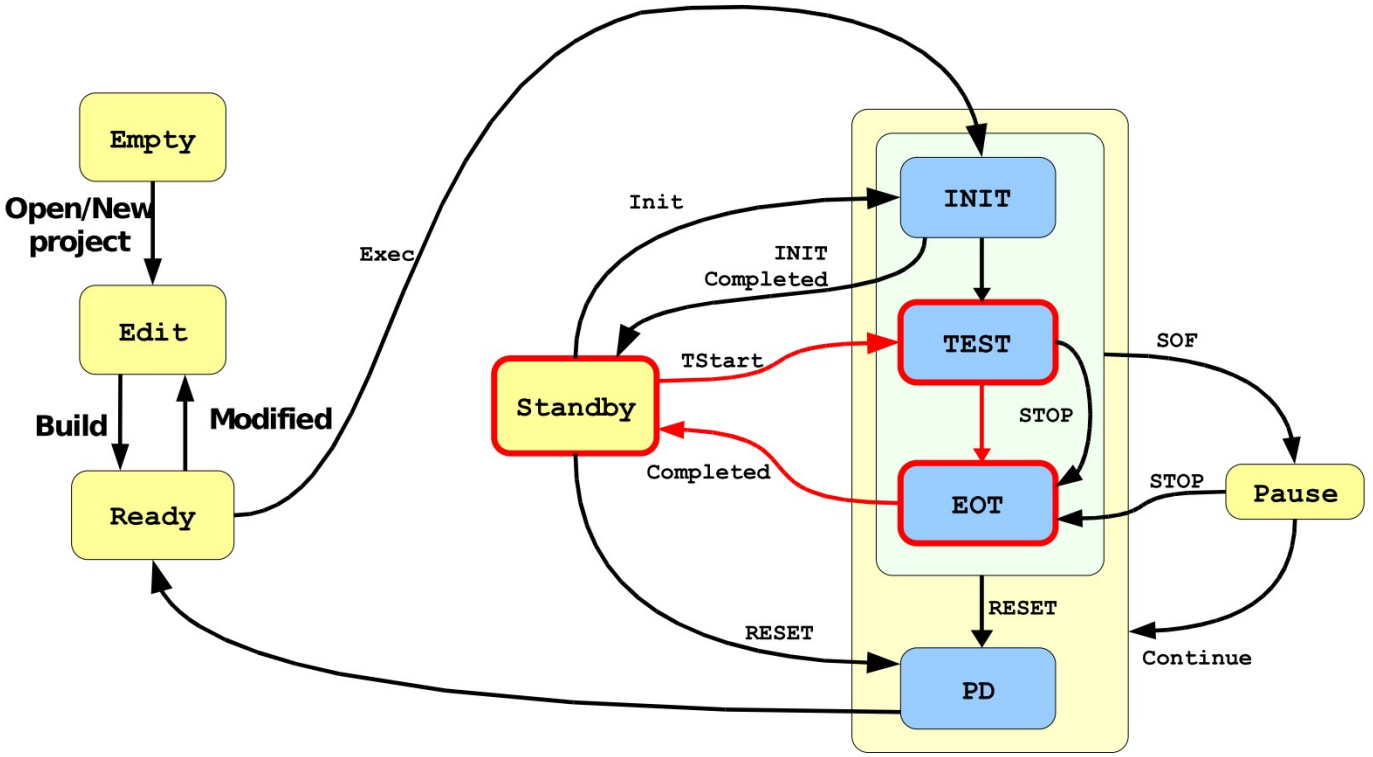
State Diagram

The following is the state diagram for MTS3:

-When user opens or creates a project, Exec it, the INIT module will be invoked and then the system stays at Standby state and waiting for TStart from PHI.

-When MTS3 receives TStart, the TEST module will be invoked, then followed by EOT module, and then after TEST & EOT have completed execution, the system goes back to Standby again and waits for next Tstart.

- When RESET button is pressed by user during above operations, the PD will be invoked, and then the system goes to Ready state.



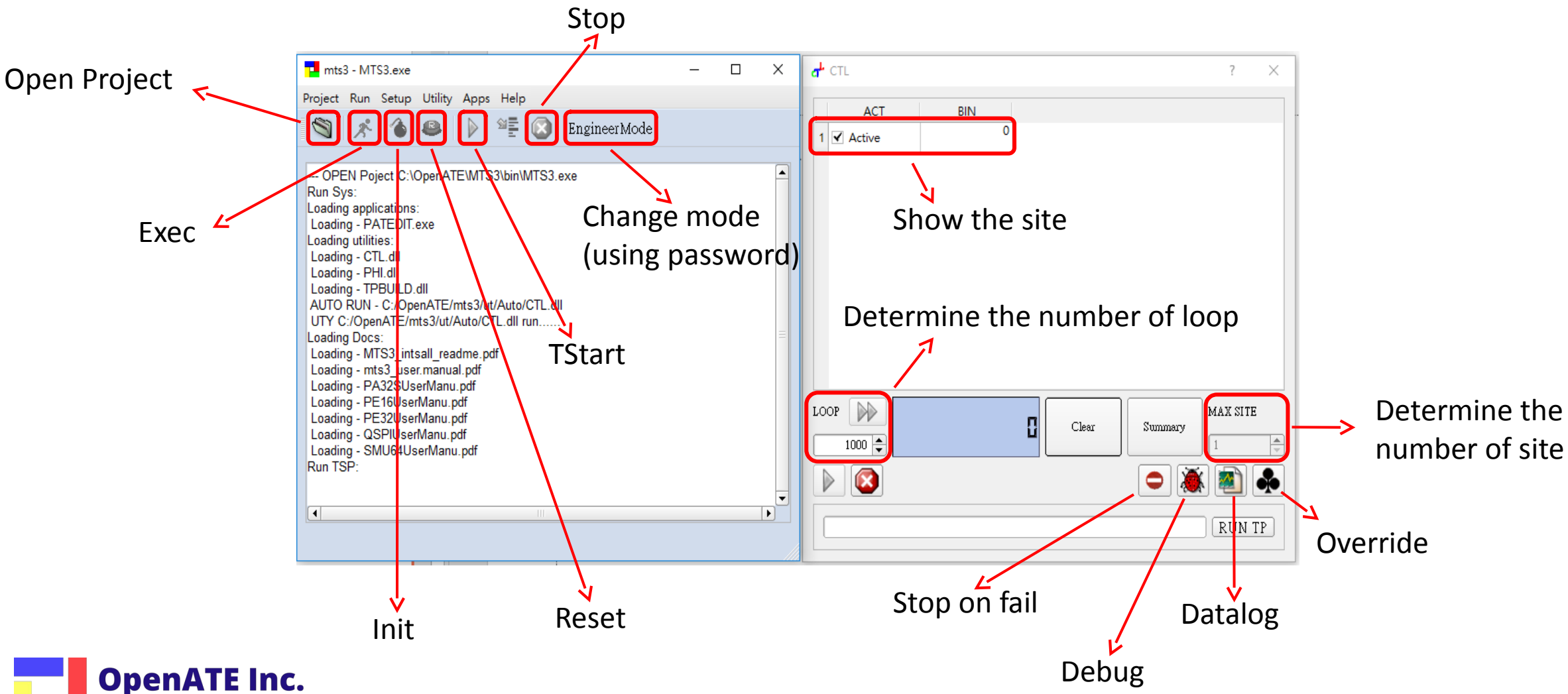
Test program structure

- INIT
  - The INIT module is responsible for the initialization of the test program. User should place all the initial routines or start-up procedures in here.
- TEST
  - The TEST module is the main test sequence. User should put all the test statements that applied to DUT in this module. Device pass/fail decision and binning strategy should be placed in this module.

Test program structure (cnt.)

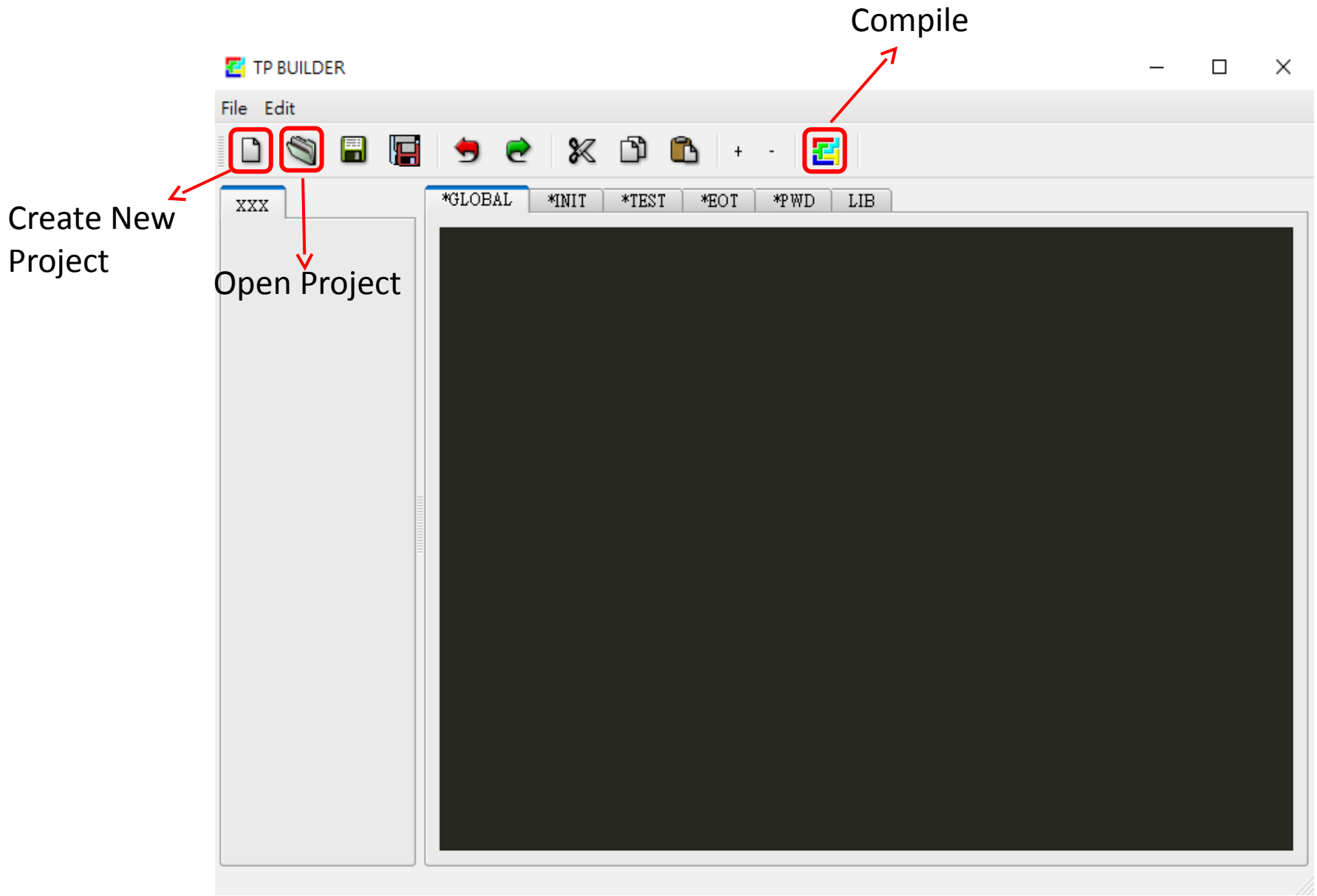
- EOT
  - The EOT module will be called after TEST module is executed completed; so the EOT module also applied to each device under test.
- POWERDOWN
  - The POWERDOWN module (PD) is invoked when user want to leave the MTS3 system and hoping to turn off all instrument powers insure safety of the test system.

User Interface for MTS3



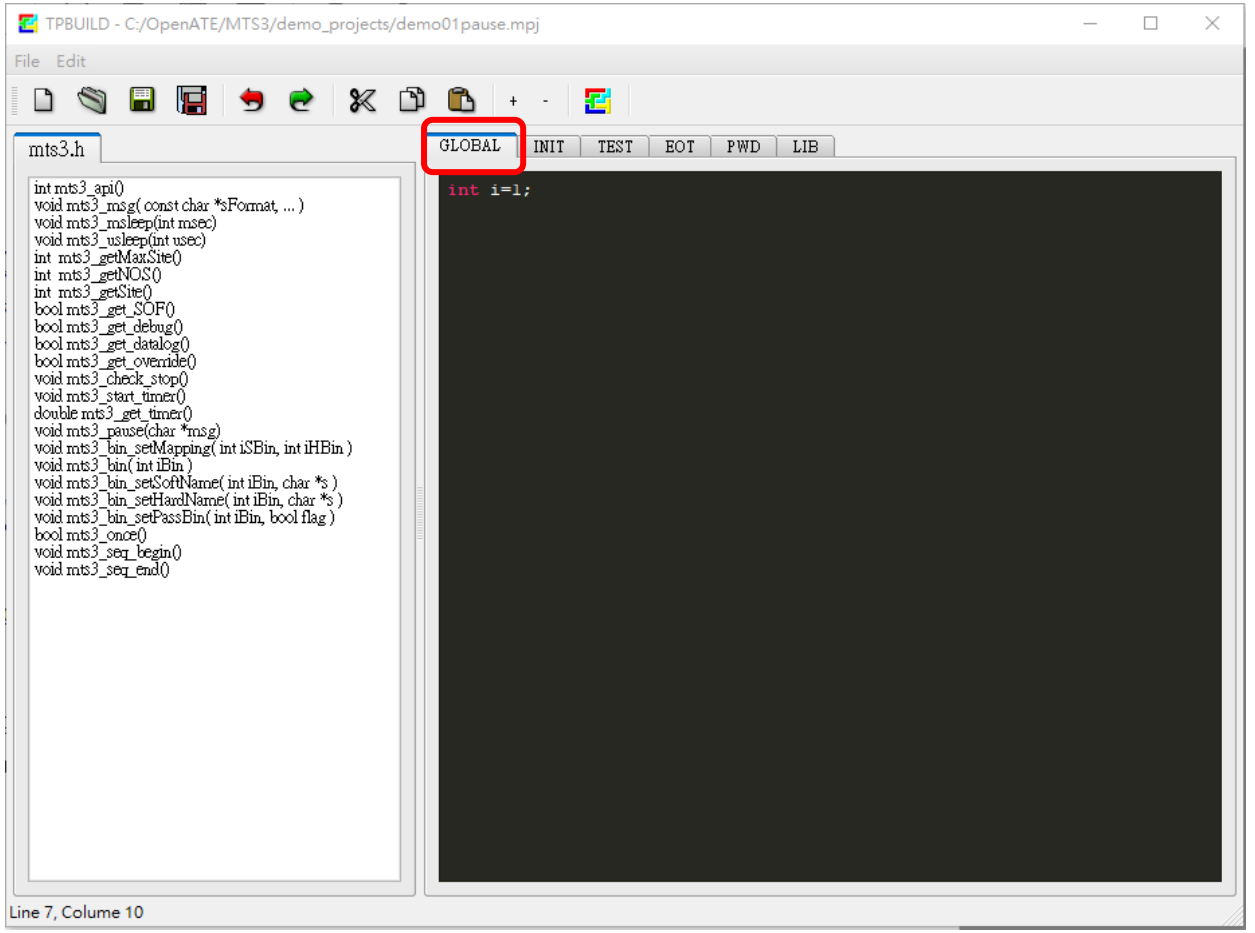
OpenATE Inc.
The open solution for IC tester

- Utility → TPBUILD



GLOBAL

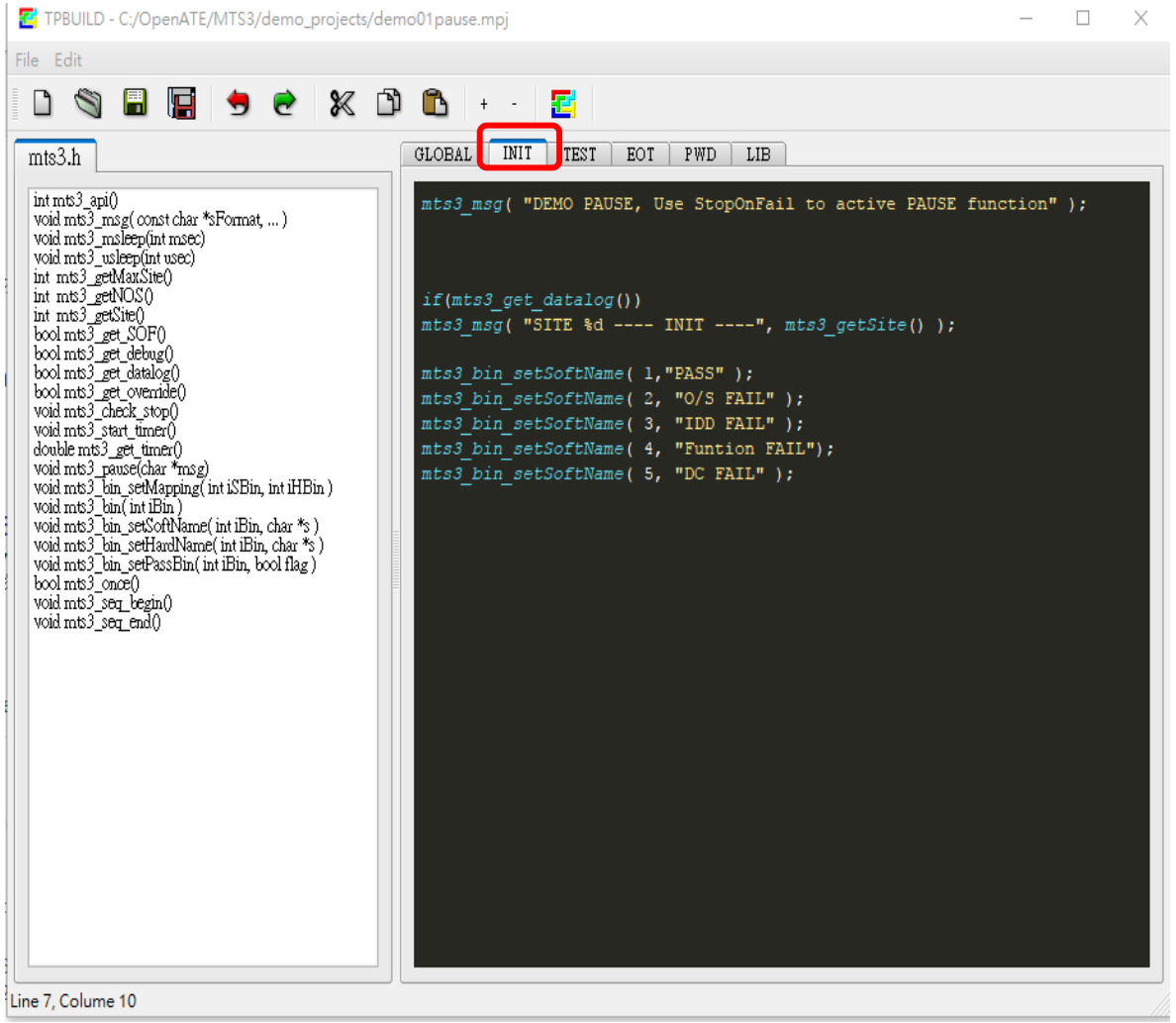
- Declare global variables and functions.



OpenATE Inc.
The open solution for IC tester

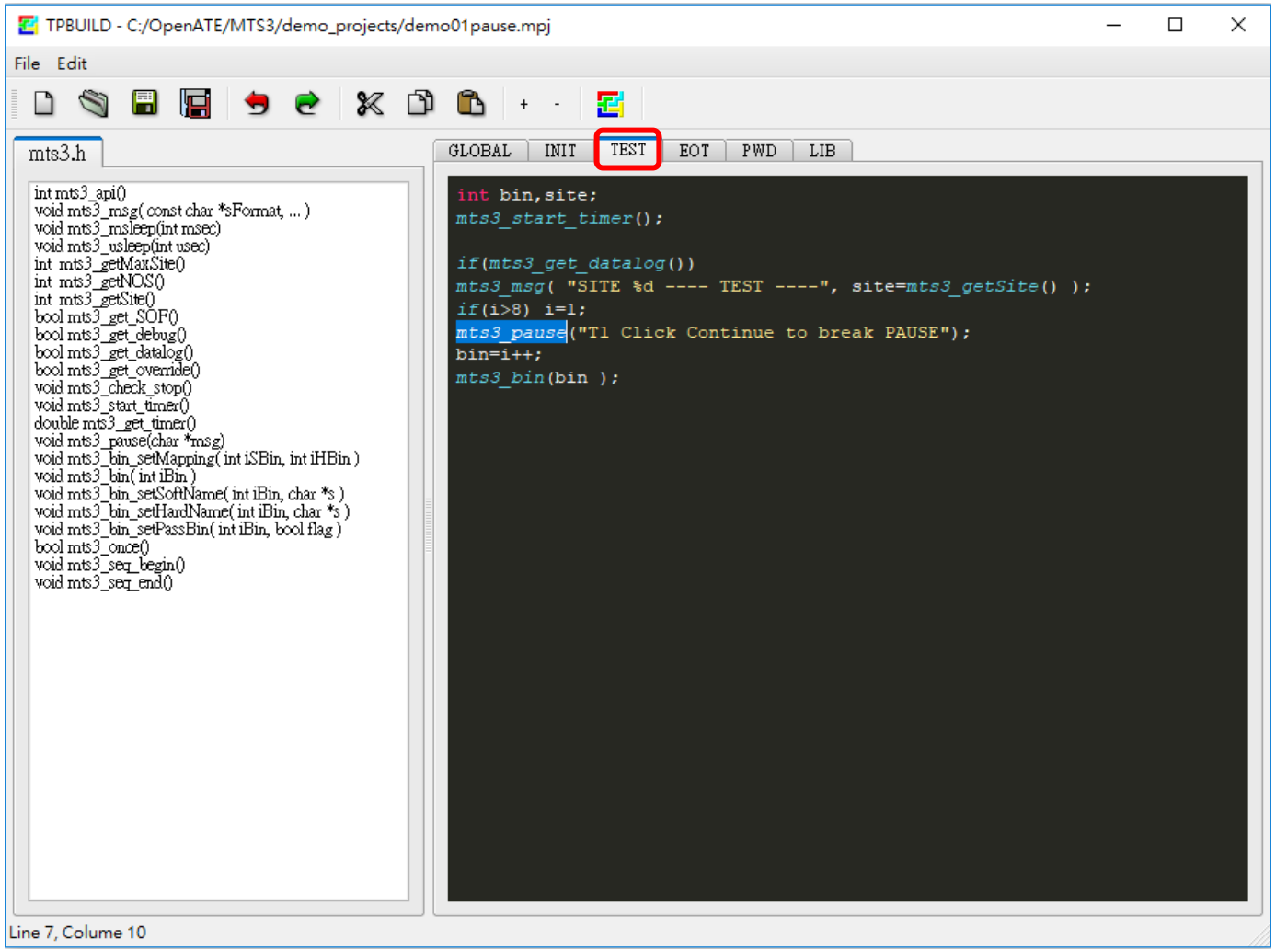
INIT

- We write initialization action in INIT window.
- When the “Exec” button or “Init” button is pressed, it will be initialized.



TEST

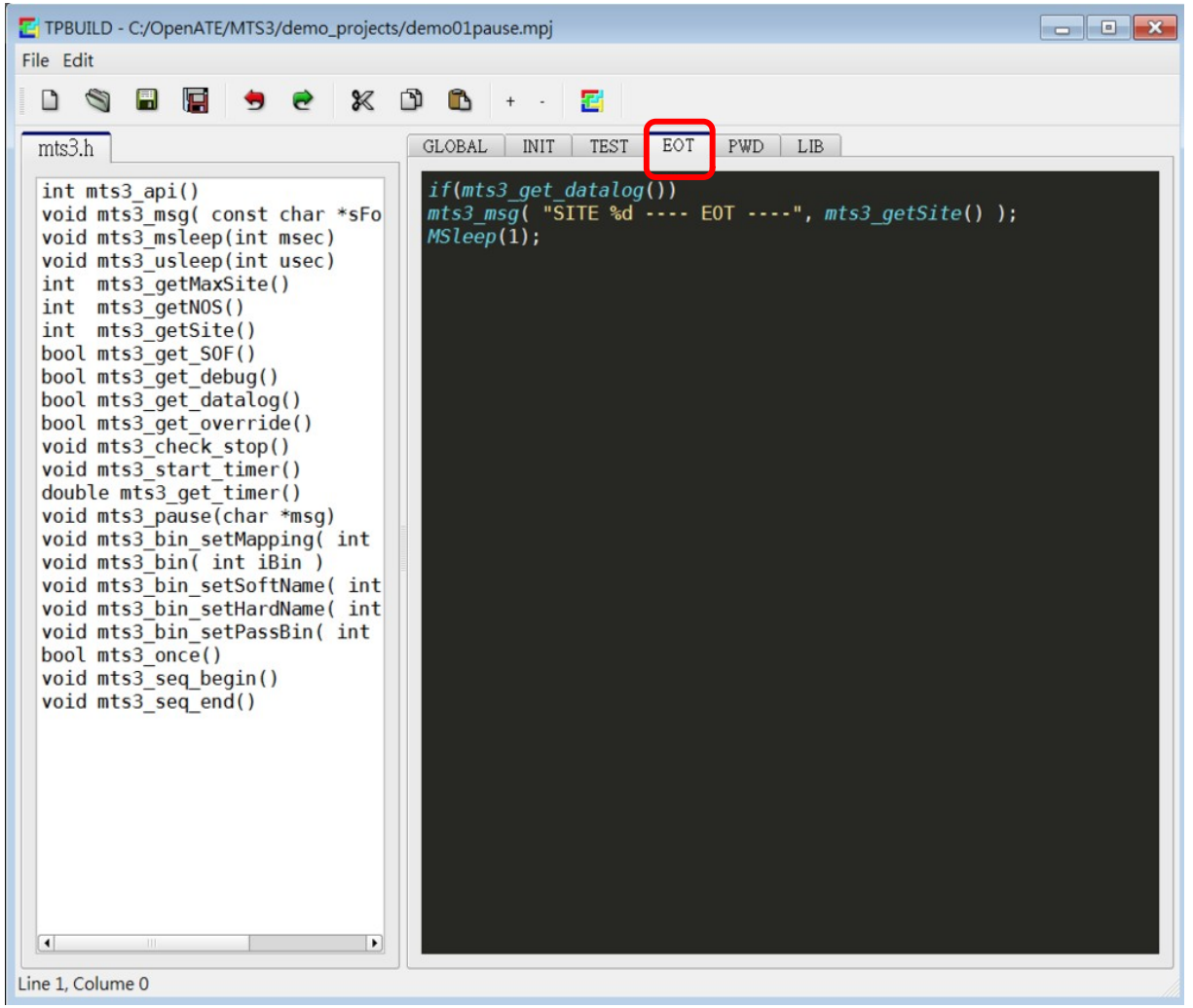
- We write test programs in TEST window.
- When the “Tstart” button is pressed, it will execute the program in TEST window.



OpenATE Inc.
The open solution for IC tester

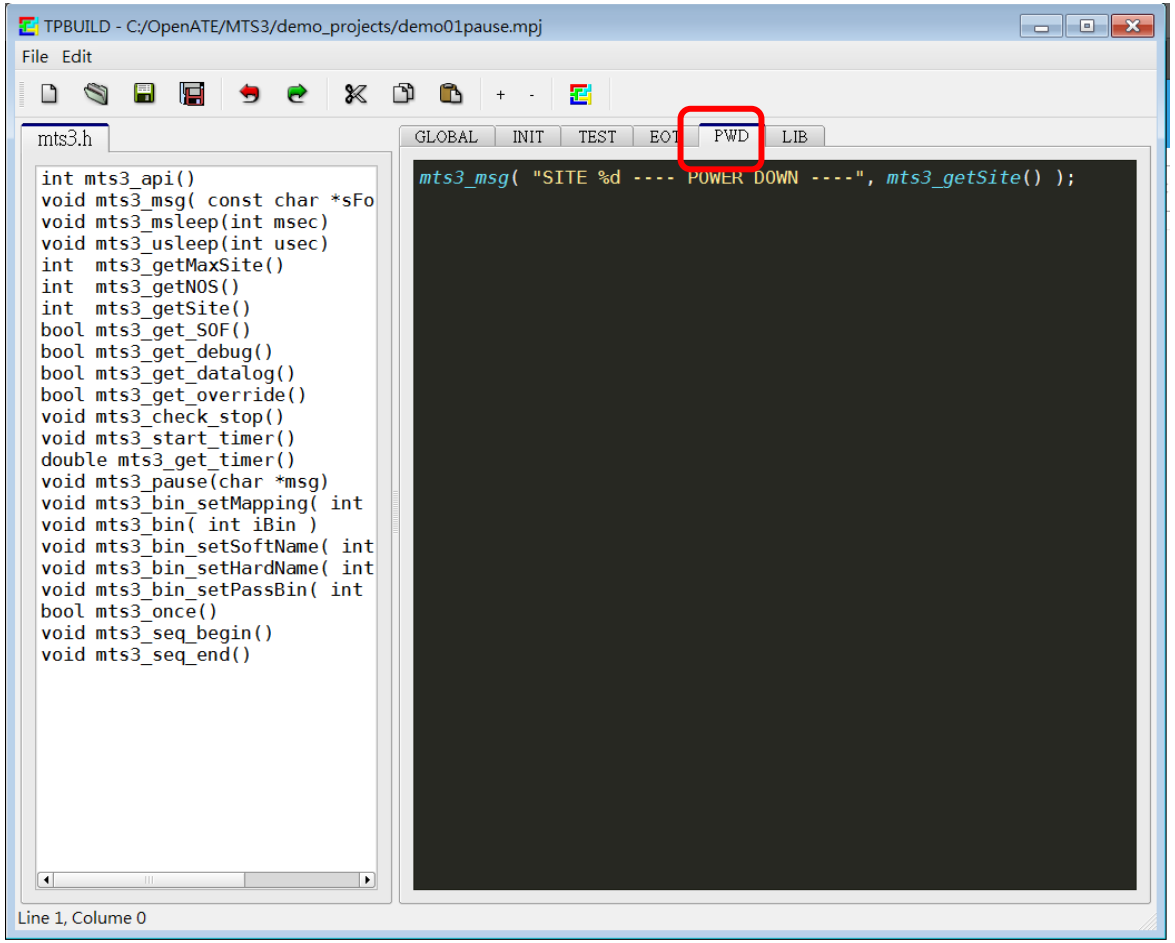
E0T

- The EOT module will be called after TEST module is executed completed.
- So the EOT module also applied to each device under test.



PWD

- The POWERDOWN module (PD) is invoked when user wants to quit the MTS3 test program(RESET pressed) and turn off all instrument powers to insure safety of the test system.



Demo Project1

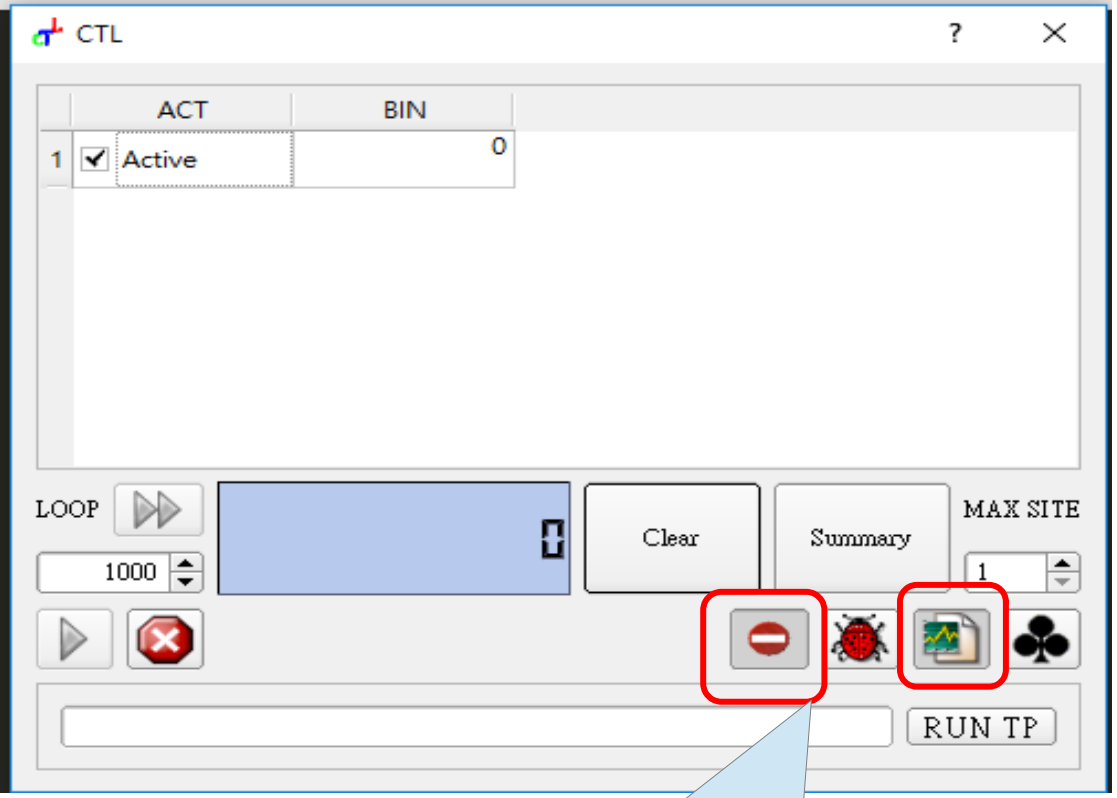
Step-by-step

Demo Project_1

Demo **PAUSE**

GLOBAL | INIT | TEST | EOT | PWD | LIB

```
int bin,site;
mts3_start_timer();

if(mts3_get_datalog())
mts3_msg( "SITE %d ---- TEST ----", site=mts3_getSite() );
if(i>8) i=1;
mts3_pause("T1 Click Continue to break PAUSE");
bin=i++;
mts3_bin(bin );
```



PAUSE and click continue

Introduction to API

- void mts3_msg(const char *sFormat, ...);
  - Print message.
  - Usage: arguments are same as c standard function “printf”.
- bool mts3_get_datalog();
  - Return button 'Datalog' state of main dialog.
- void mts3_bin_setSoftName(int iBin, char *s);
  - Set name of specified software bin.

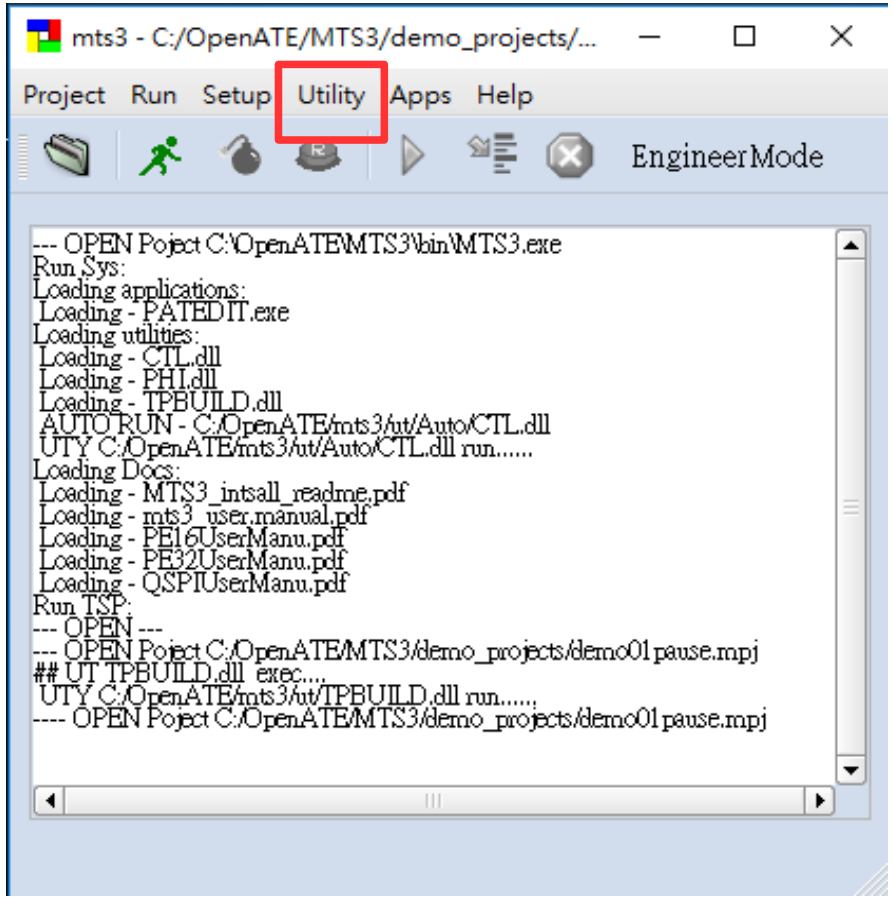
Introduction to API

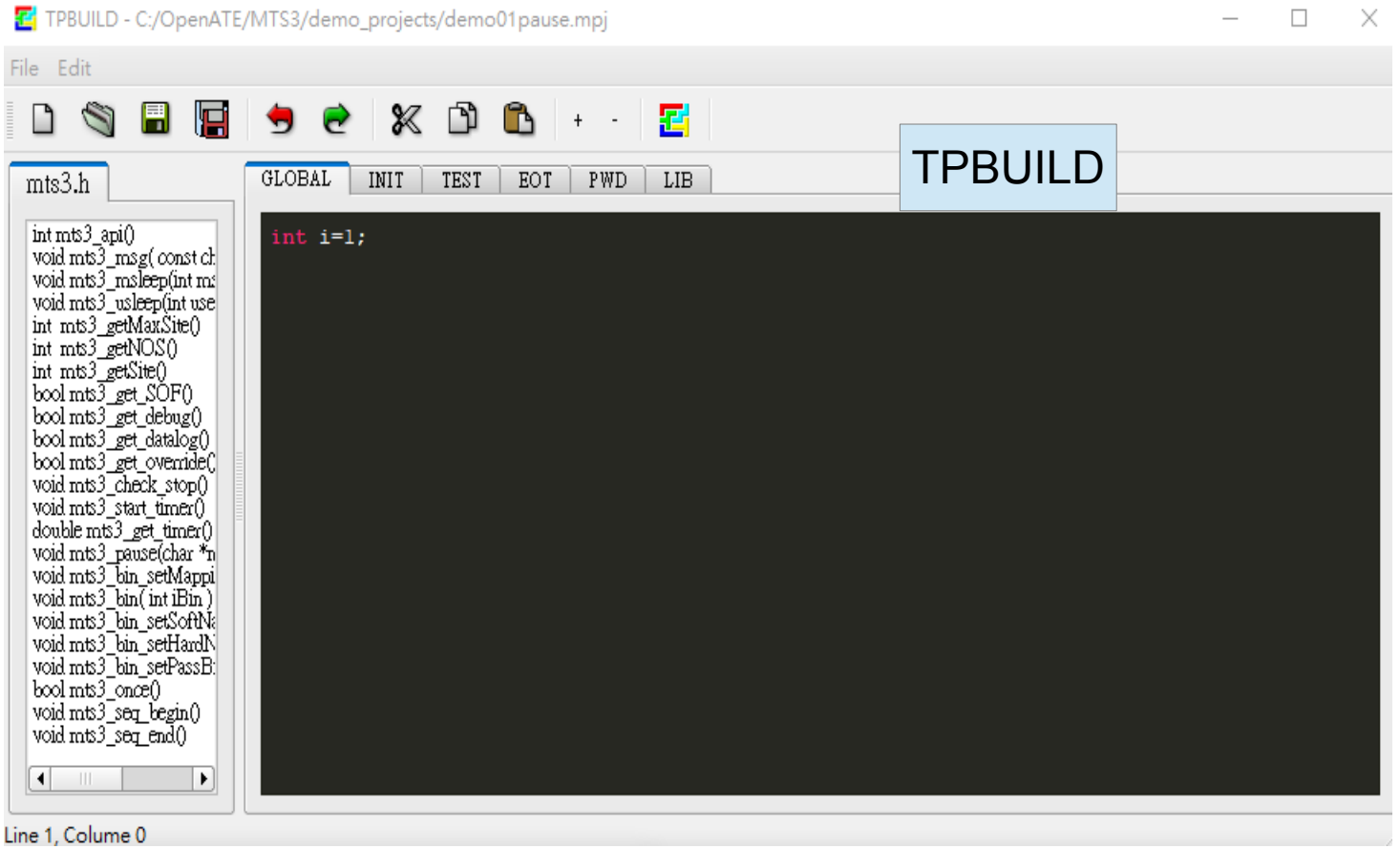
- void mts3_pause(char *msg);
  - Pause execution of test program until user hits continue button.

- void mts3_bin(int iBin);
  - Issue bin for current test.

Step1

- Open MTS3
- Open project “C:\OpenATE\MTS3\demo_projects\demo01pause.mpj”
- Open TPBUILD <Utility → TPBUILD >





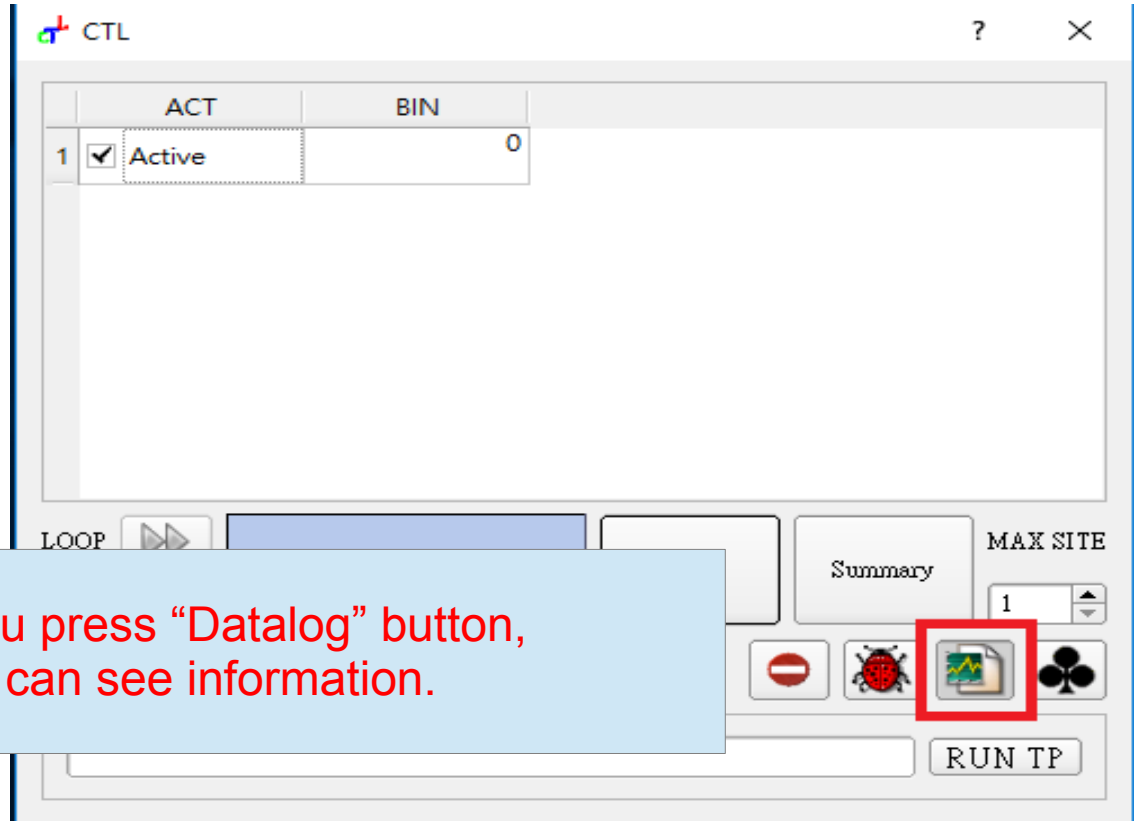
Step2

- You can press “Datalog” button to see some information.

GLOBAL | INIT | TEST | EOT | PWD | LIB

```
mts3_msg( "DEMO PAUSE, Use StopOnFail to active PAUSE function" );


if(mts3_get_datalog())
mts3_msg( "SITE %d ---- INIT ----", mts3_getSite() );

mts3_bin_setSoftName( 1,"PASS" );
mts3_bin_setSoftName( 2, "O/S FAIL" );
mts3_bin_setSoftName( 3, "IDD FAIL" );
mts3_bin_setSoftName( 4, "Funtion FAIL");
mts3_bin_setSoftName( 5, "DC FAIL" );
```

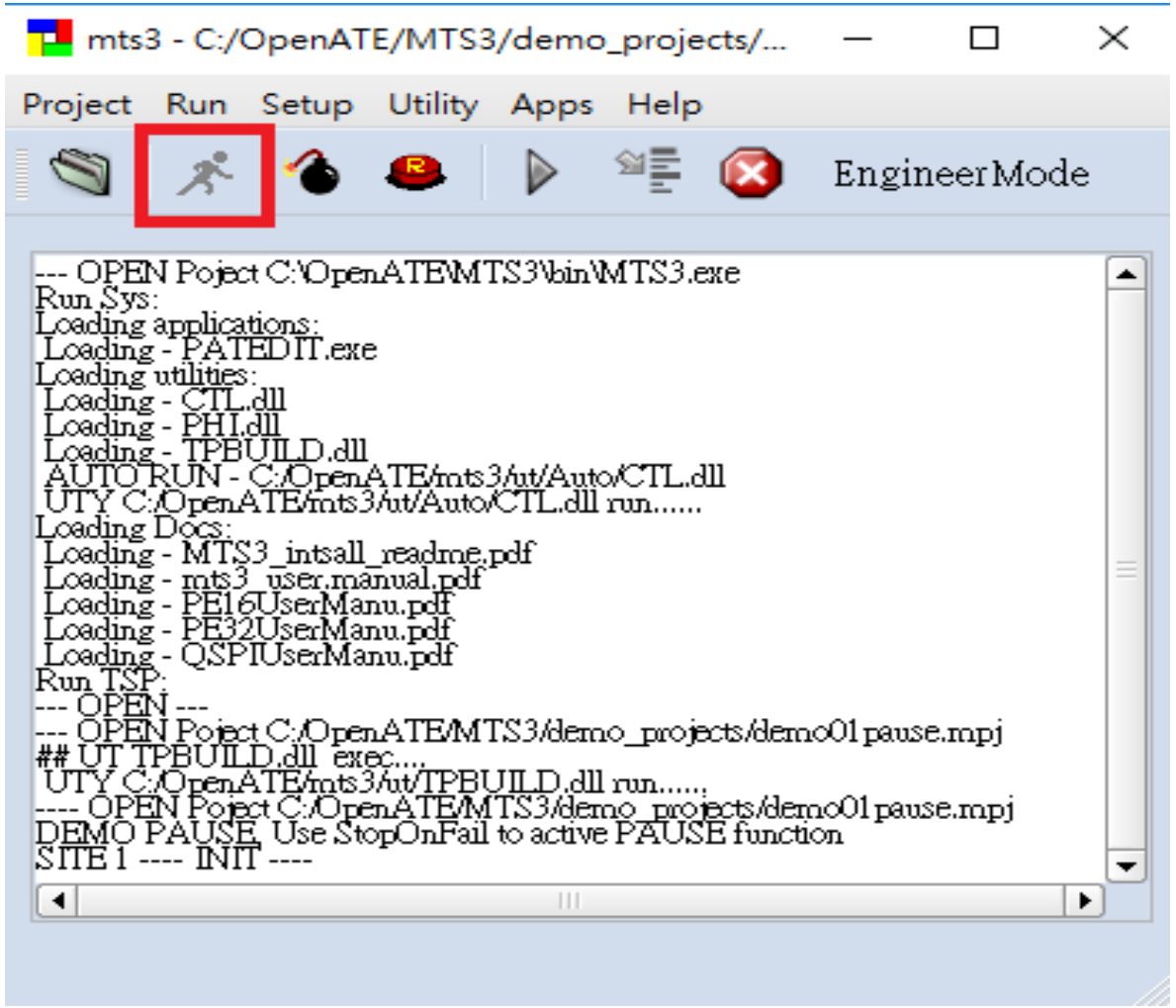
When you press “Datalog” button, you can see information.



OpenATE Inc.
The open solution for IC tester

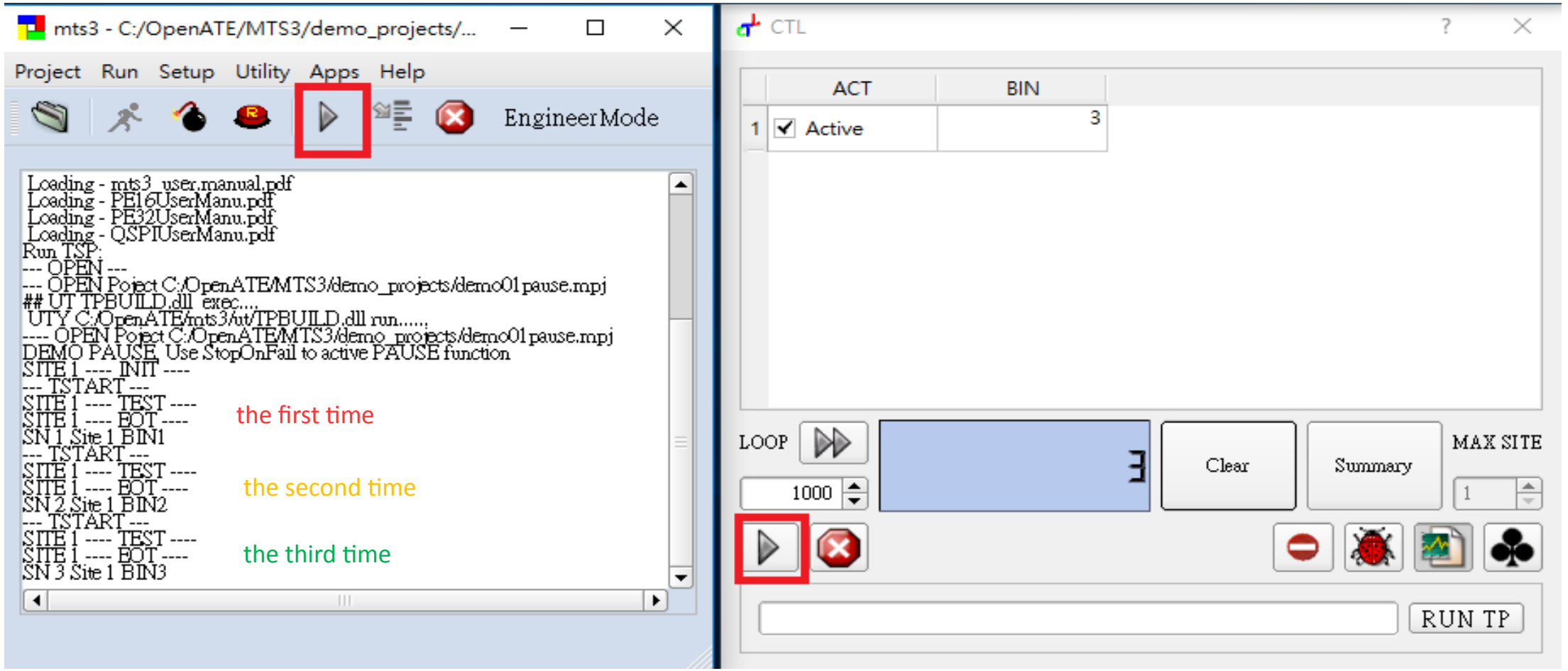
Step3

- Press “Exec” button to initialize.



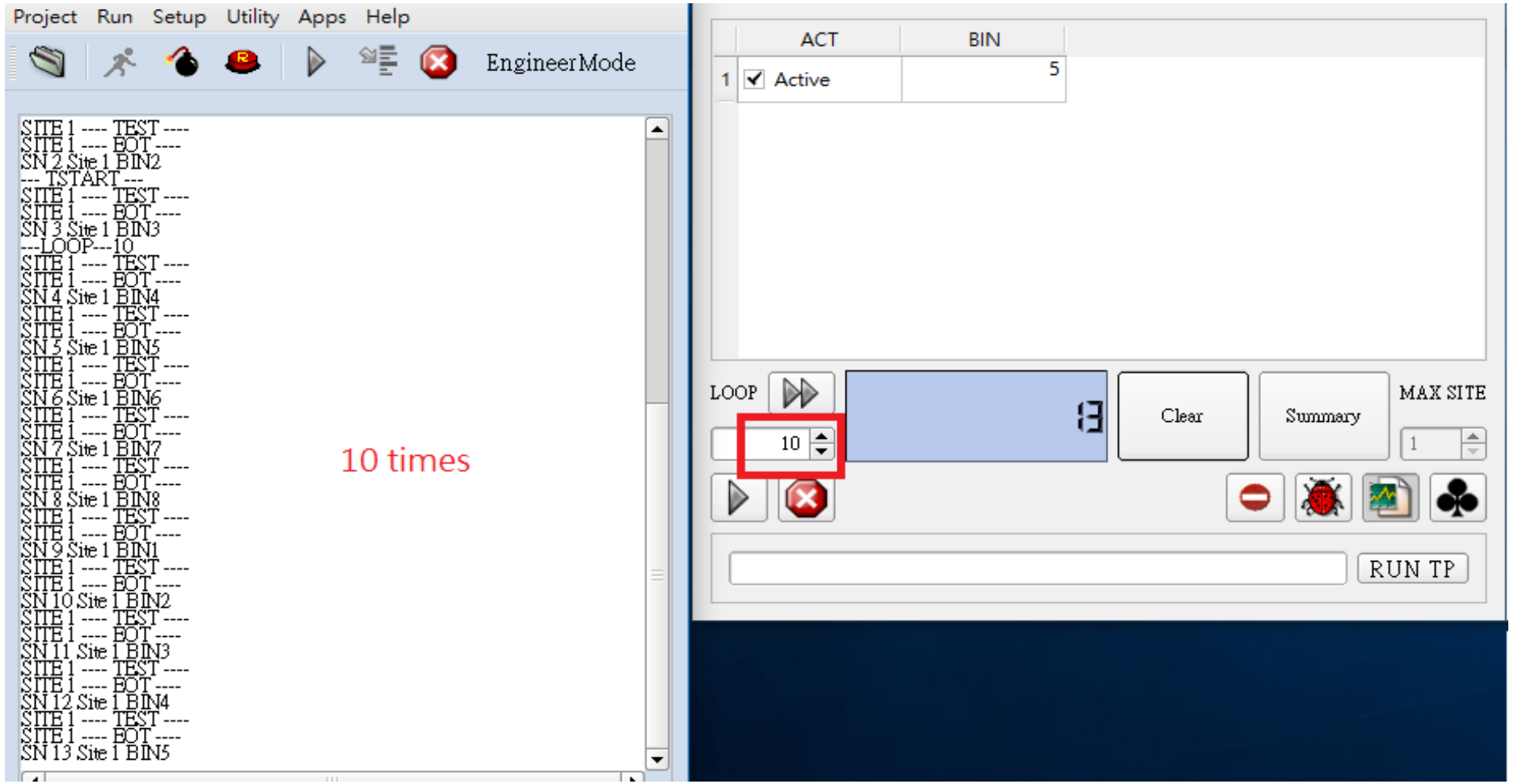
Step4

- Press “Tstart” button to run the test program once.



Step5

- You can press “Loop” button to run several times.



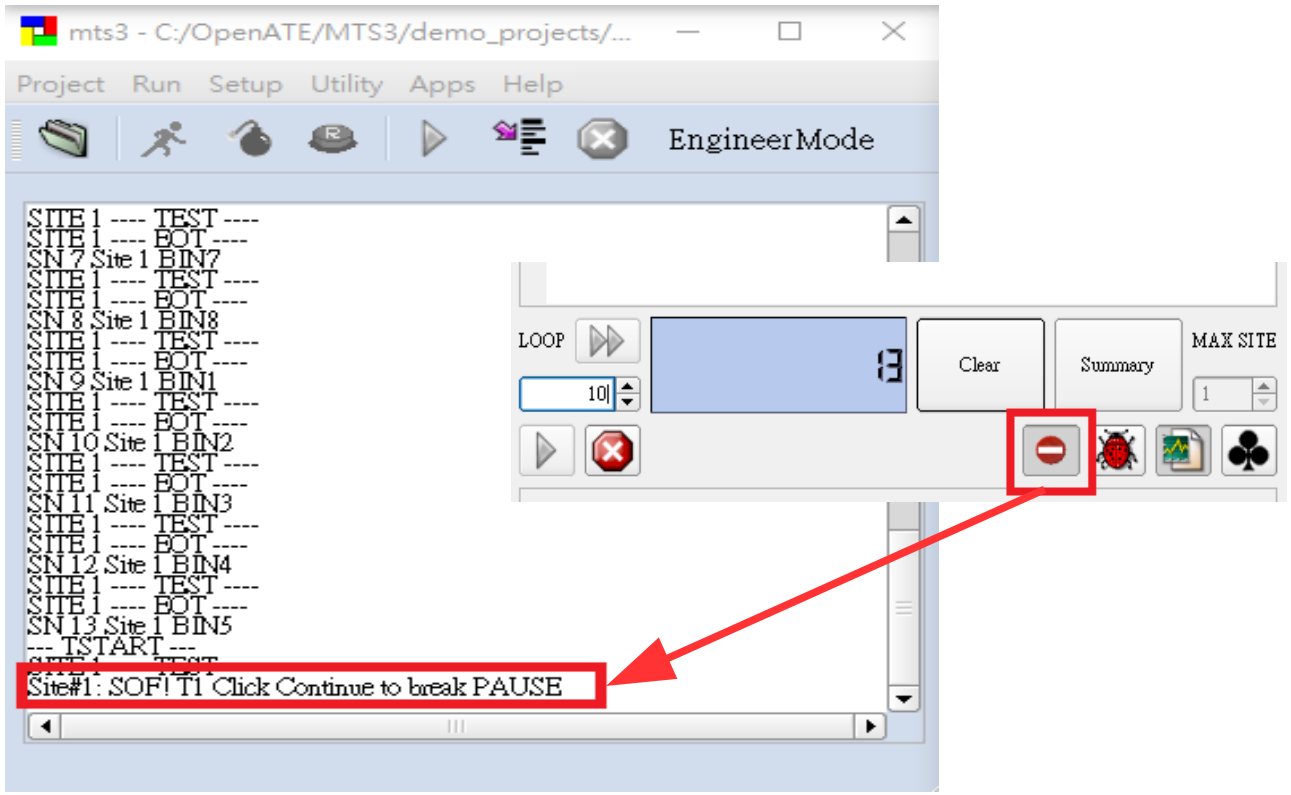
Step6

- Now, we will test mts3_pause() function.
- You must press “Pause” button and press “Tstart” button to run the test program once.

GLOBAL | INIT | TEST | EOT | PWD | LIB

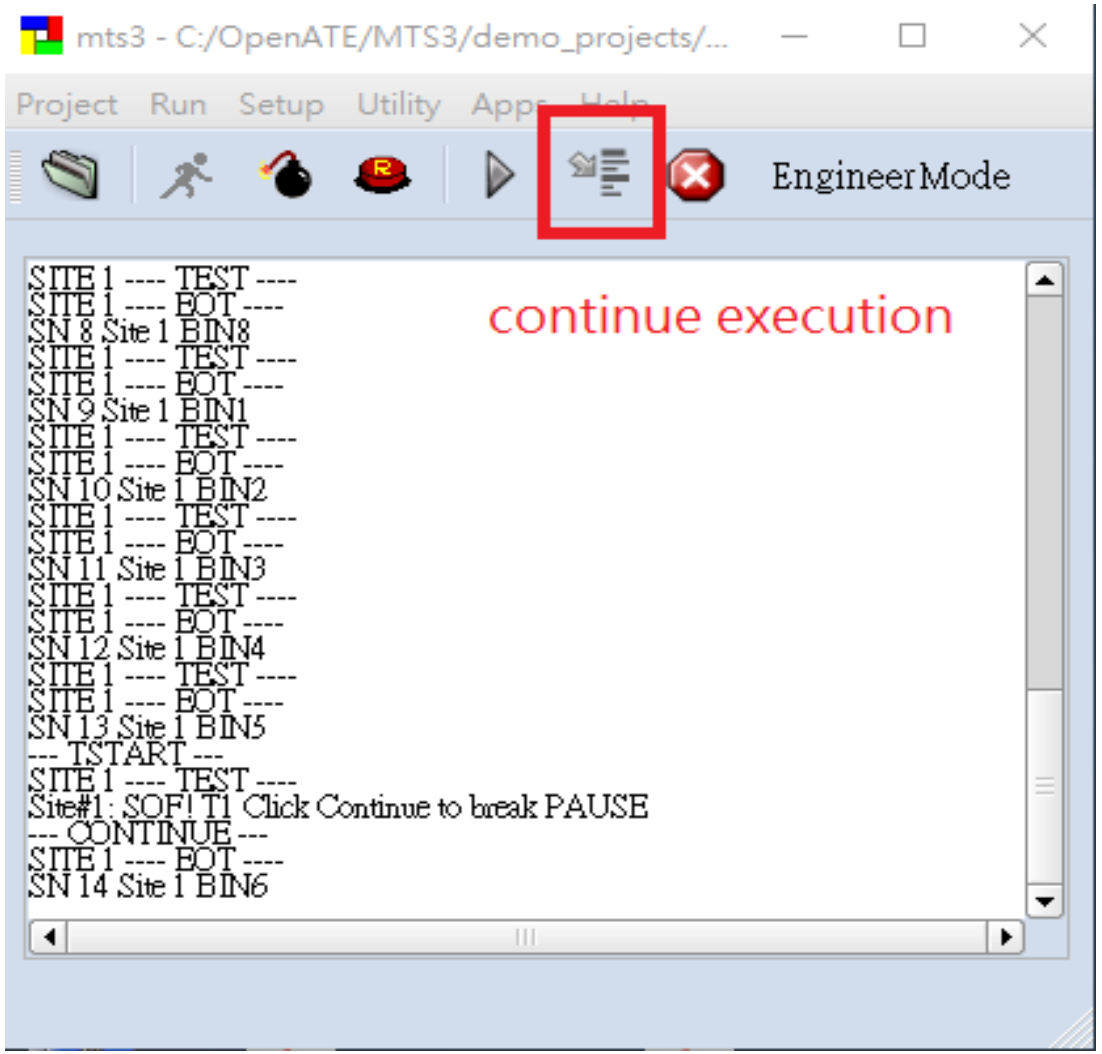
```
int bin,site;
mts3_start_timer();

if(mts3_get_datalog())
mts3_msg( "SITE %d ---- TEST ----", site=mts3_getSite() );
if(i>8) i=1;
mts3_pause("T1 Click Continue to break PAUSE");
bin=i++;
mts3_bin(bin );
```

Pause execution of test program until user hits continue button.



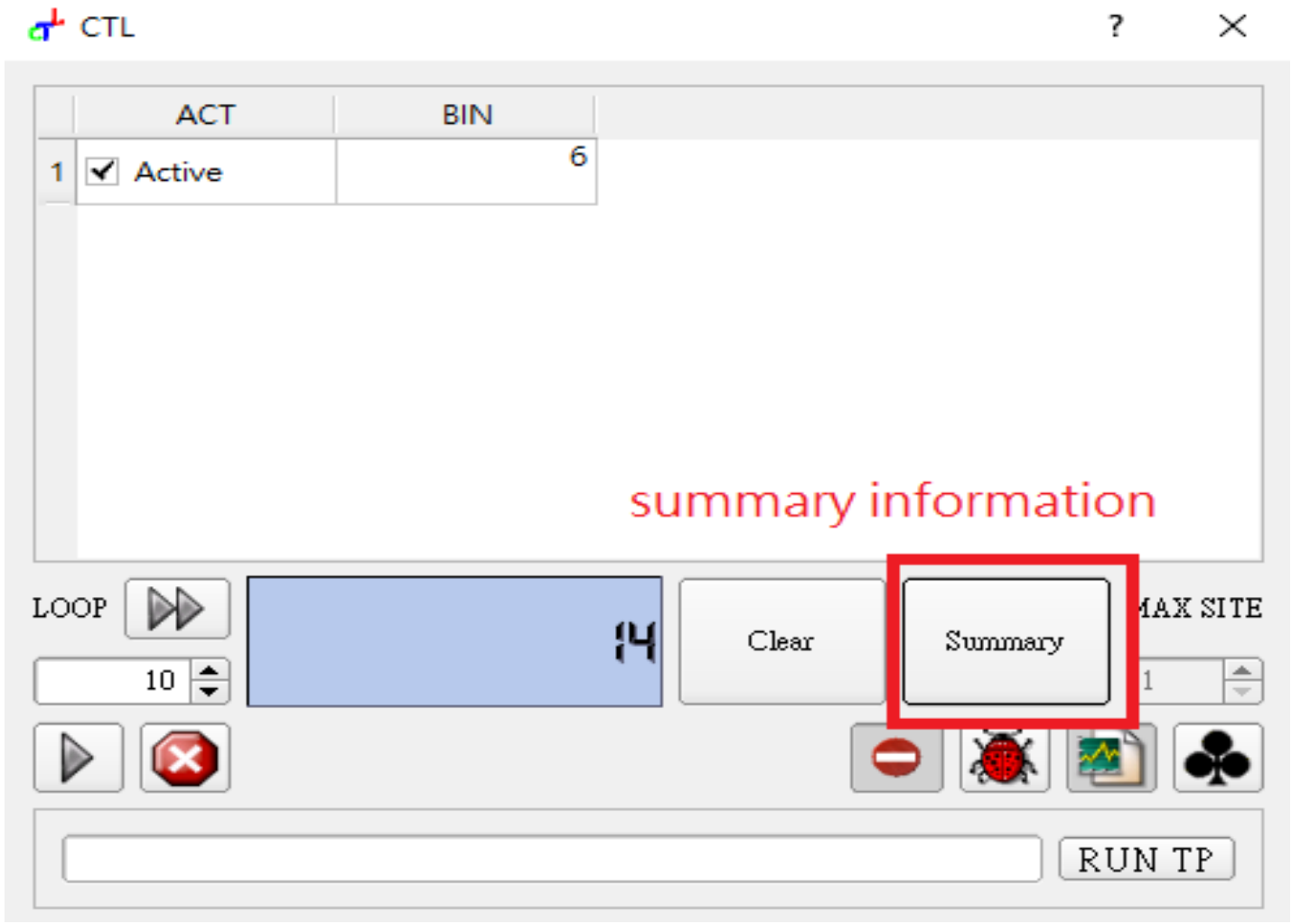
Step7

- Press “Continue” button to continue execution.



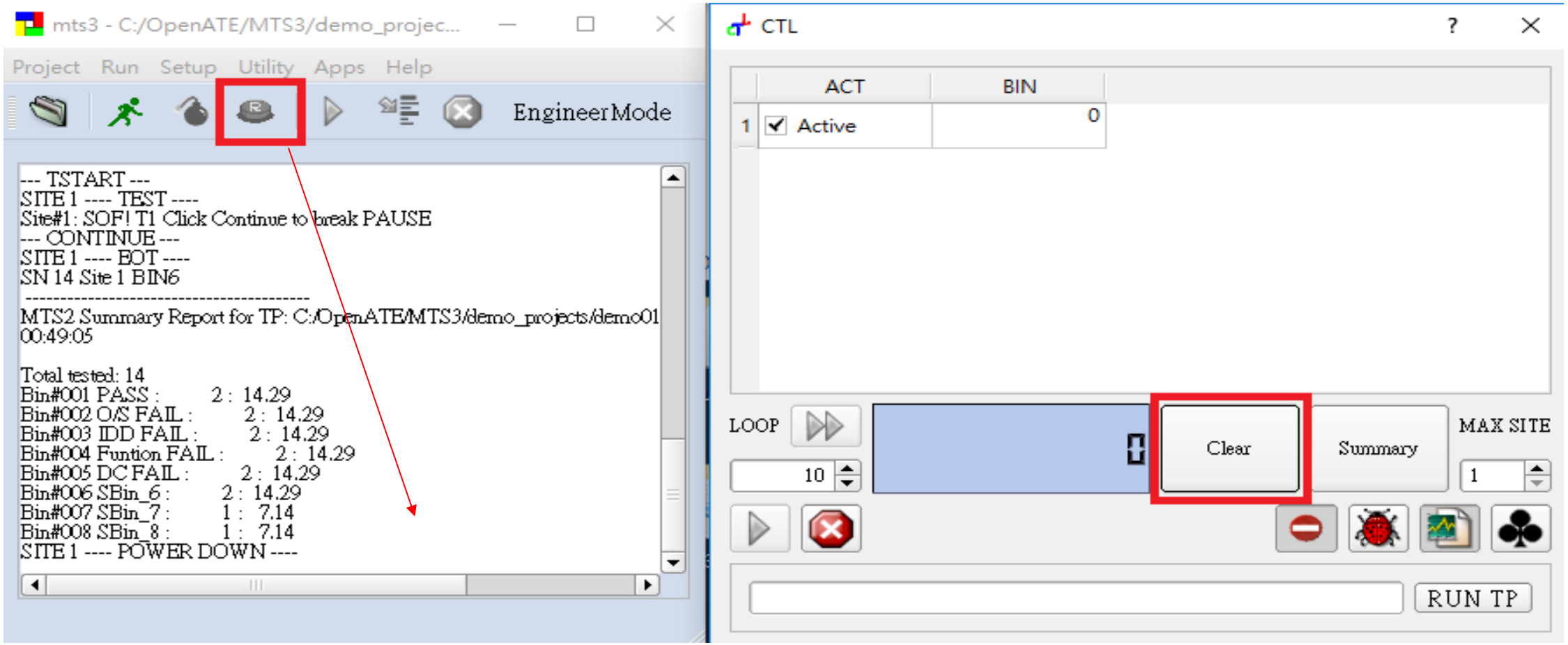
Step8

- Press “Summary” button to see summary information.



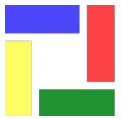
Step9

- Press “Reset” button to finish execution.
- Press “Clear” button to clear the number of times.



Demo Project2

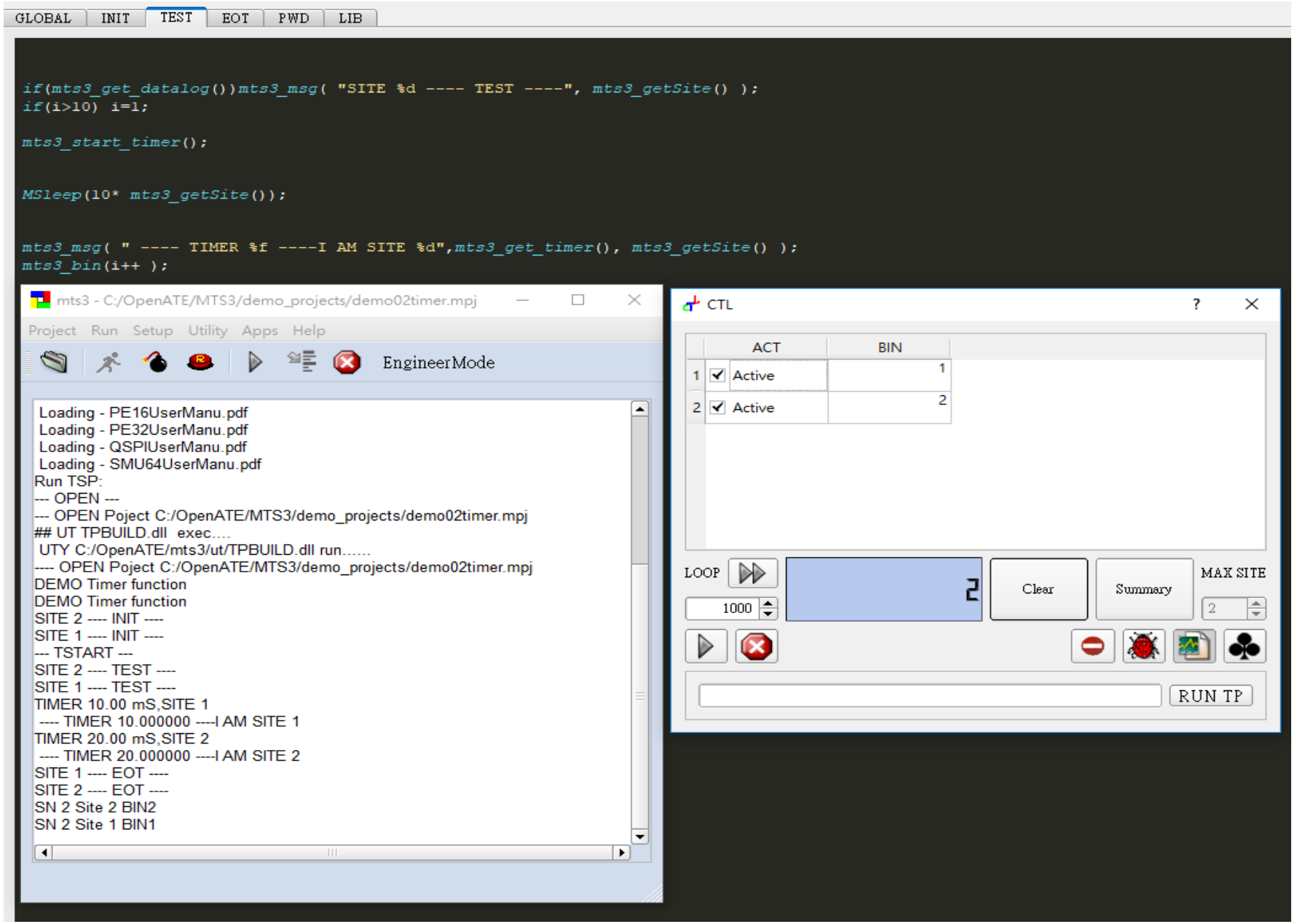
Step-by-step



OpenATE Inc.

The open solution for IC tester

- Demo
  TIMER



OpenATE Inc.
The open solution for IC tester

Introduction to API

- void mts3_start_timer();
- double mts3_get_timer();
  - These two functions are used to measure the time elapsed from one point to another in the test program.
    - mts3_start_time() : Mark current time.
    - mts3_get_timer() : Return the time elapsed (in ms) since last call to mts3_start_time().
    - If datalog is on, the time will also be printed in the datalog pane.

Step1

- Open MTS3
- Open project “C:\OpenATE\MTS3\demo_projects\demo02timer.mpj”
- Open TPBUILD <Utility → TPBUILD >

Step2

- You can press “Datalog” button to see some information.

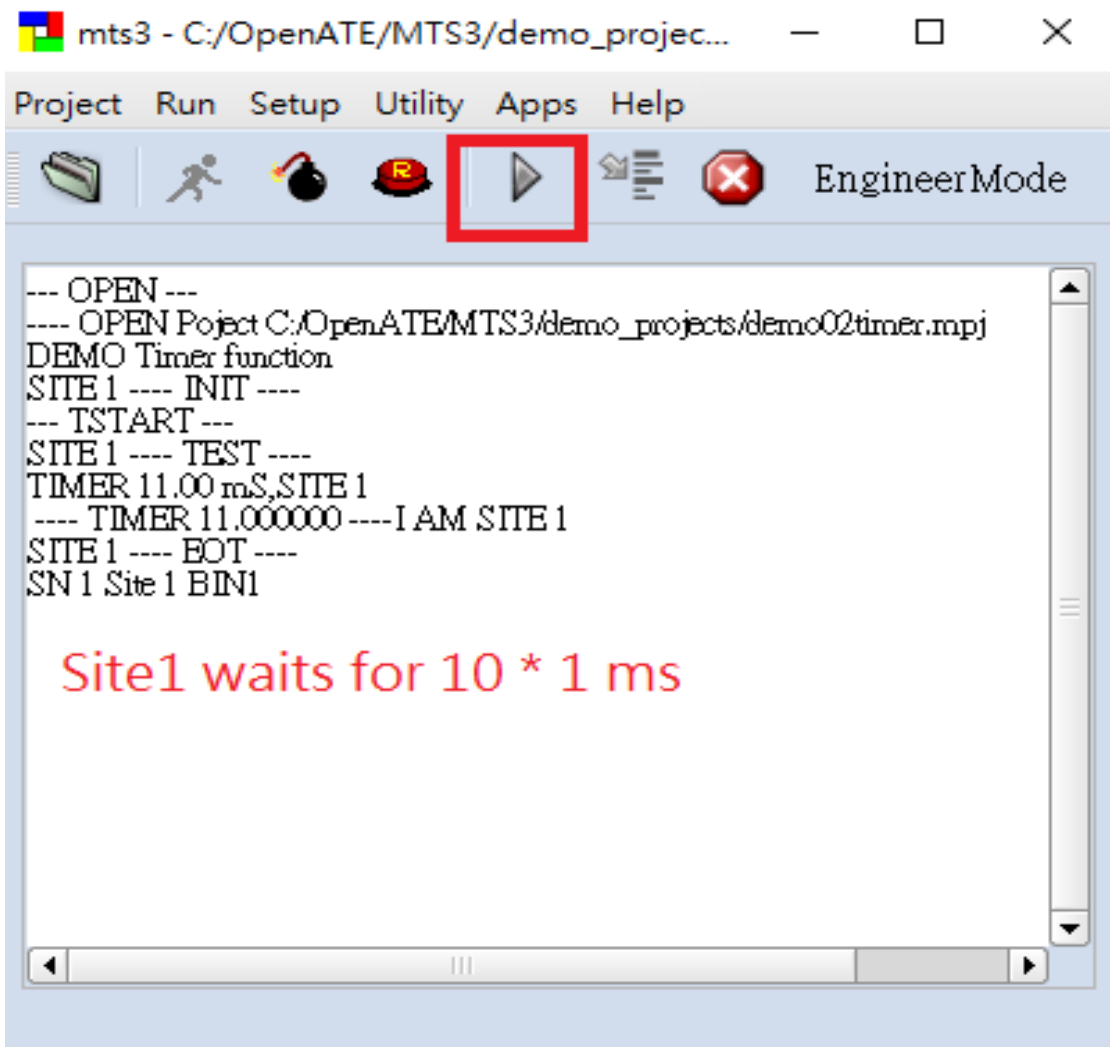
Step3

- Press “Exec” button to initialize.

Step4

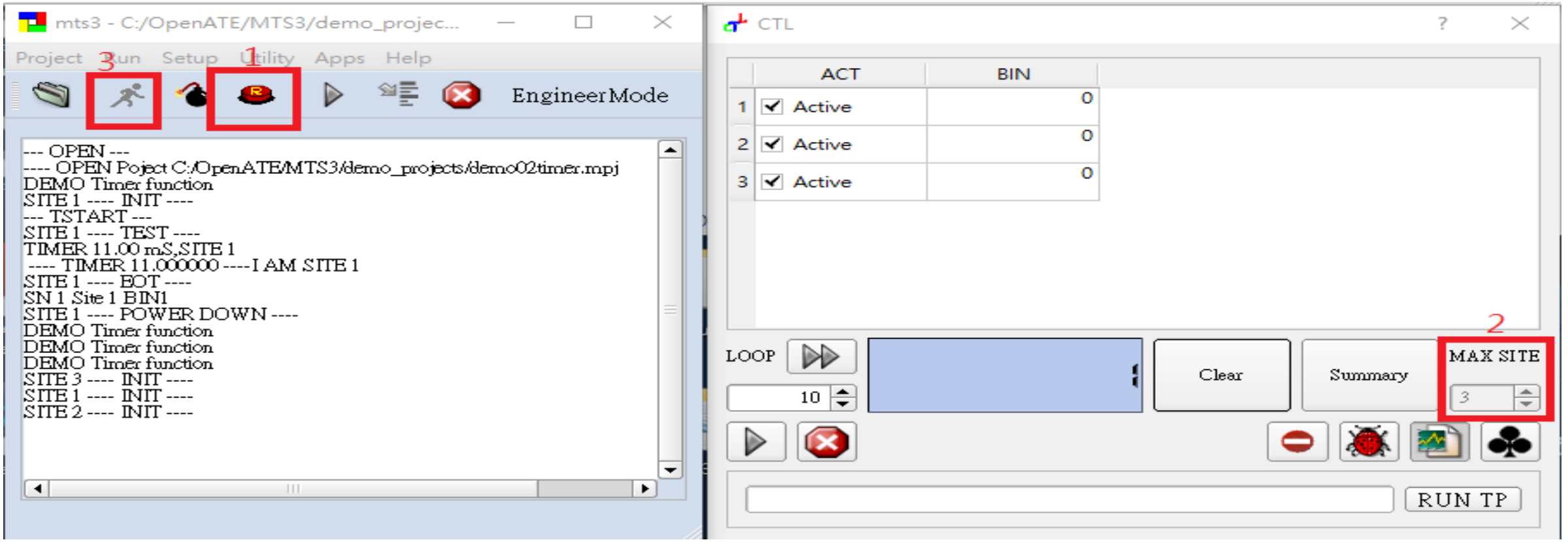
- Press “Tstart” button to run the test program once.

```
if(mts3_get_datalog())mts3_msg( "SITE %d ---- TEST ----", mts3_getSite() );
if(i>10) i=1;

mts3_start_timer();


MSleep(10* mts3_getSite());


mts3_msg( " ---- TIMER %f ----I AM SITE %d",mts3_get_timer(), mts3_getSite() );
mts3_bin(i++ );
```

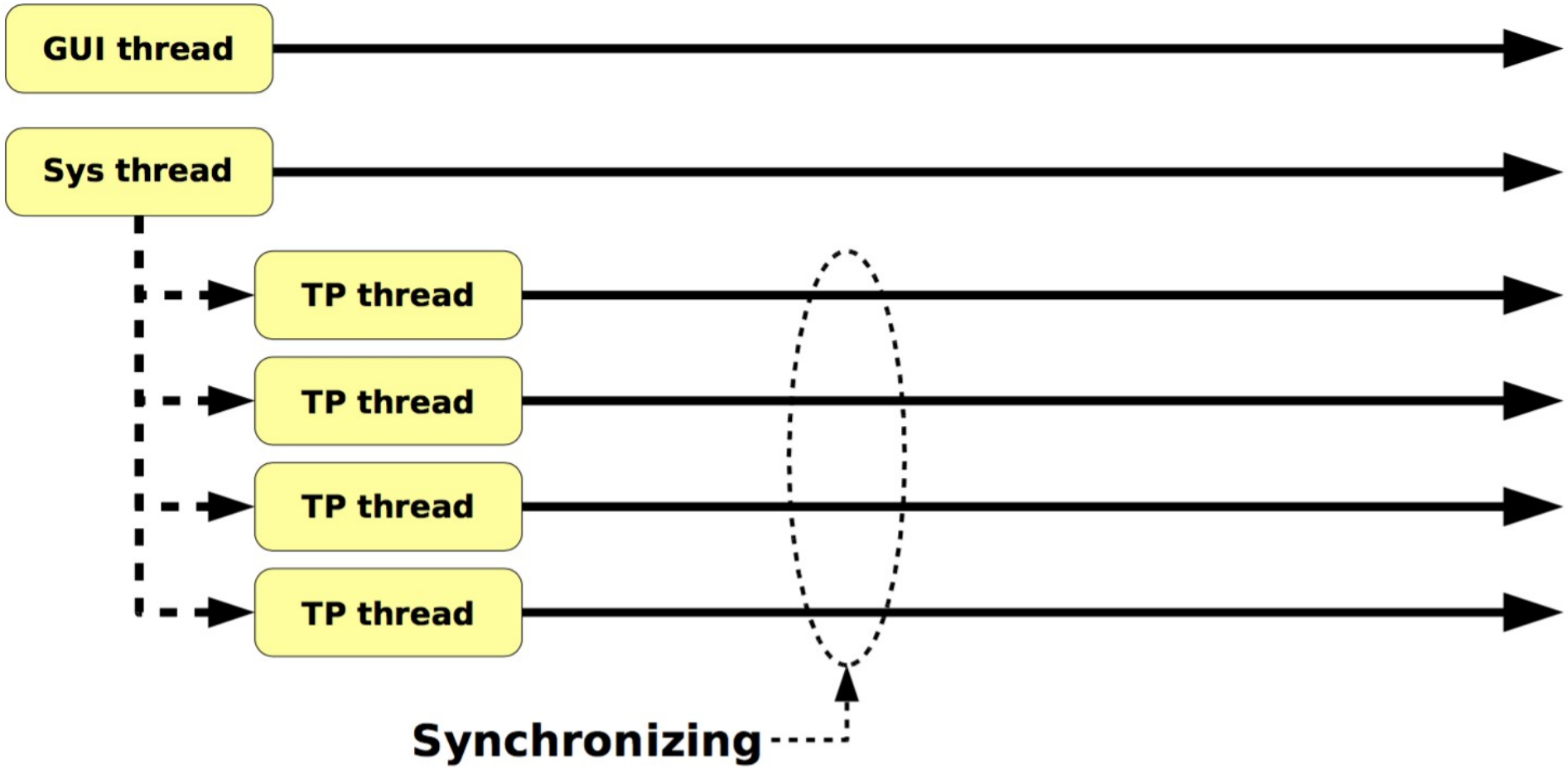


Step5

- Press “Reset” button to finish execution.
- Increase the number of sites.
- Press “Exec” button to initialize.

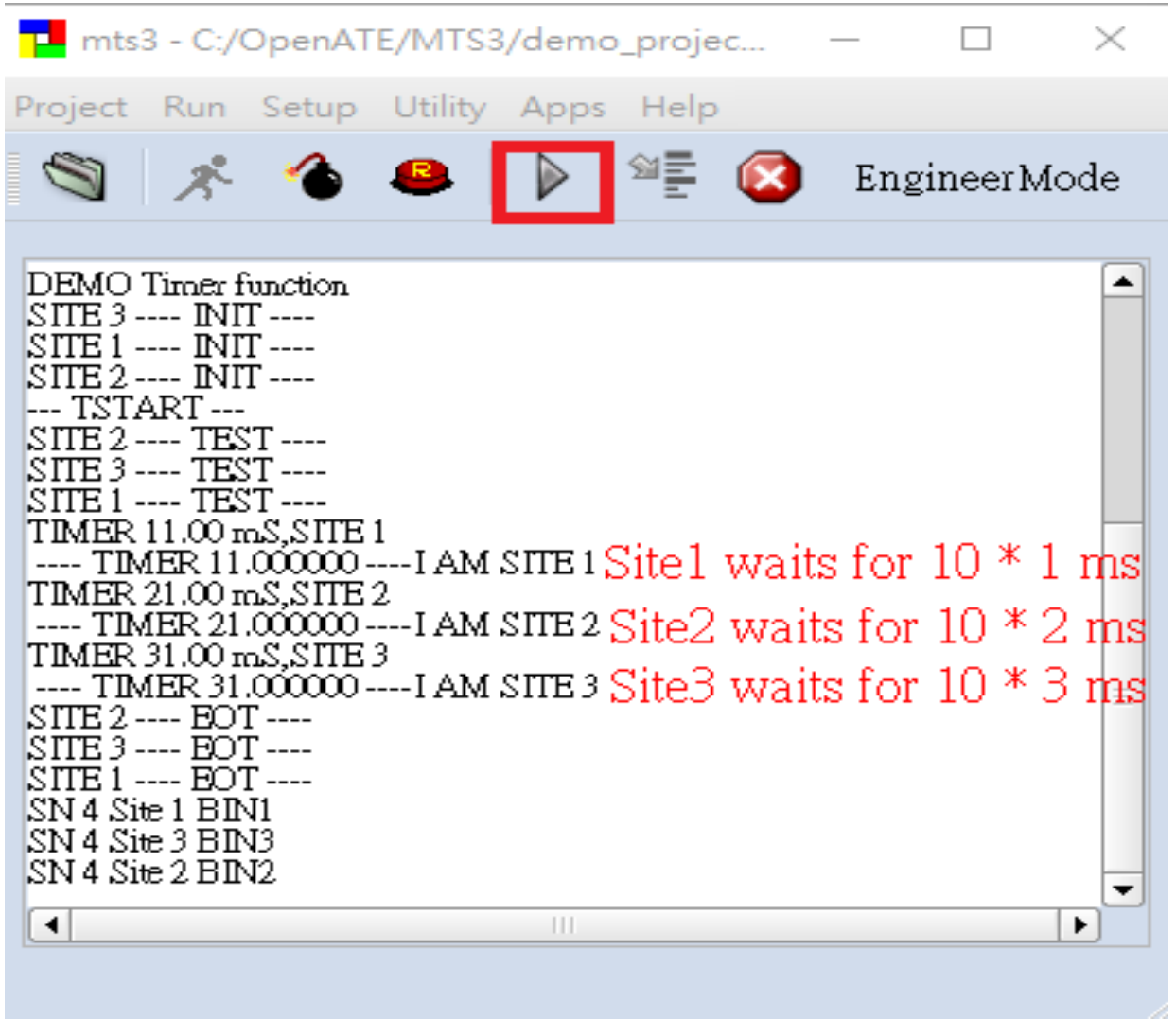


Threading


GUI thread
Sys thread
TP thread
TP thread
TP thread
TP thread
Synchronizing

Step6

- Press “Tstart” button to run the test program once.



Demo Project3

Step-by-step

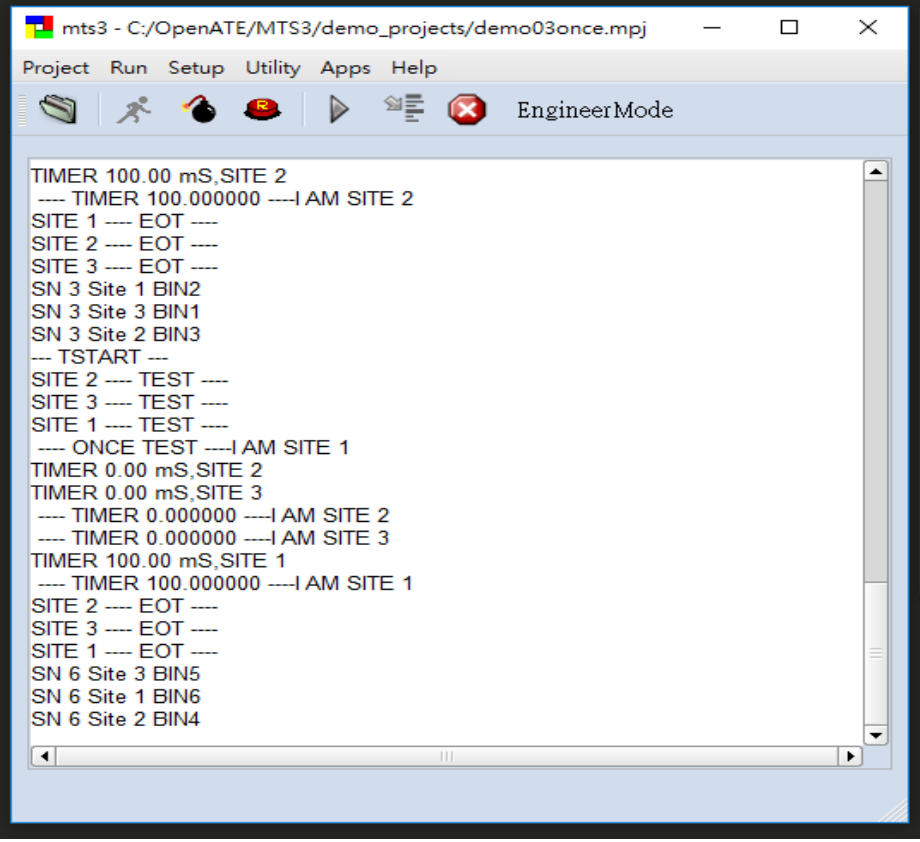
Demo Project_3

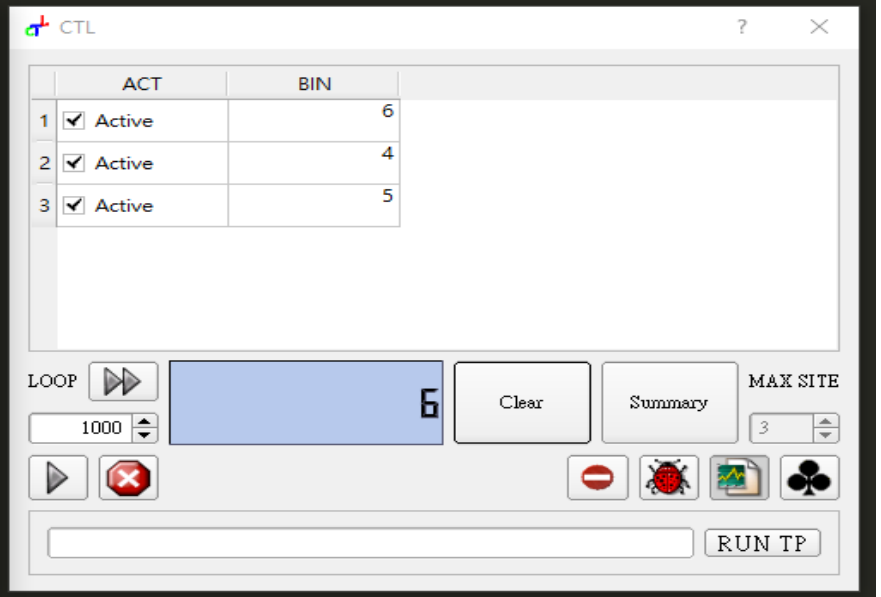
Demo

-

GLOBAL | INIT | TEST | EOT | PWD | LIB

```
if(mts3_get_datalog())mts3_msg( "SITE %d ---- TEST ----", mts3_getSite() );
if(i>10) i=1;

mts3_start_timer();

if(mts3_once()){
mts3_msg( " ---- ONCE TEST ----I AM SITE %d", mts3_getSite() );
MSleep(100);
}
if(mts3_get_debug())mts3_once();
mts3_msg( " ---- TIMER %f ----I AM SITE %d",mts3_get_timer(), mts3_getSite() );
mts3_bin(i++ );
```





OpenATE Inc.
The open solution for IC tester

Introduction to API

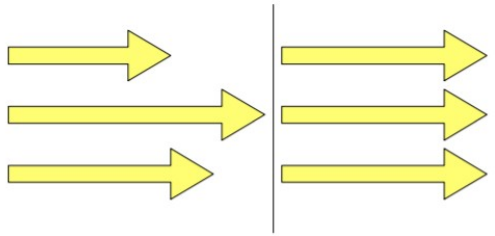
- bool mts3_once();
  - Sometimes for a multi-site application, a piece of code may need to be executed only once by one thread.
  - Usage:

```
if ( mts3_once() ) {
      // statements in this block will be executed only once by one thread.
 }
```

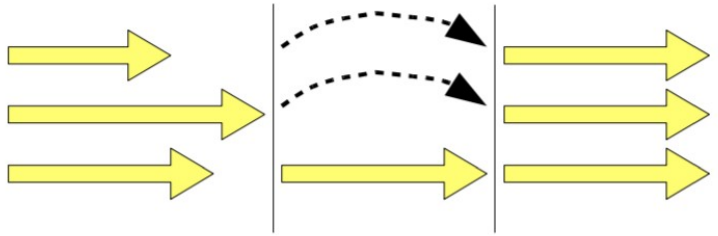
  - mts3_once() is also used to sync the execution among test threads.

Synchronizing

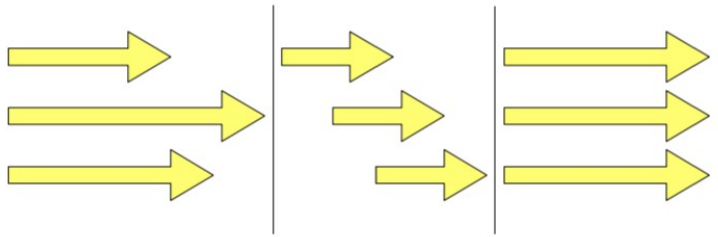
Synch



Once



Sequence



Step1

- Open MTS3
- Open project “C:\OpenATE\MTS3\demo_projects\demo03once.mpj”
- Open TPBUILD <Utility → TPBUILD >

Step2

- You can press “Datalog” button to see more information.

Step3

- Increase the number of sites. 

- Press “Exec” button to initialize.

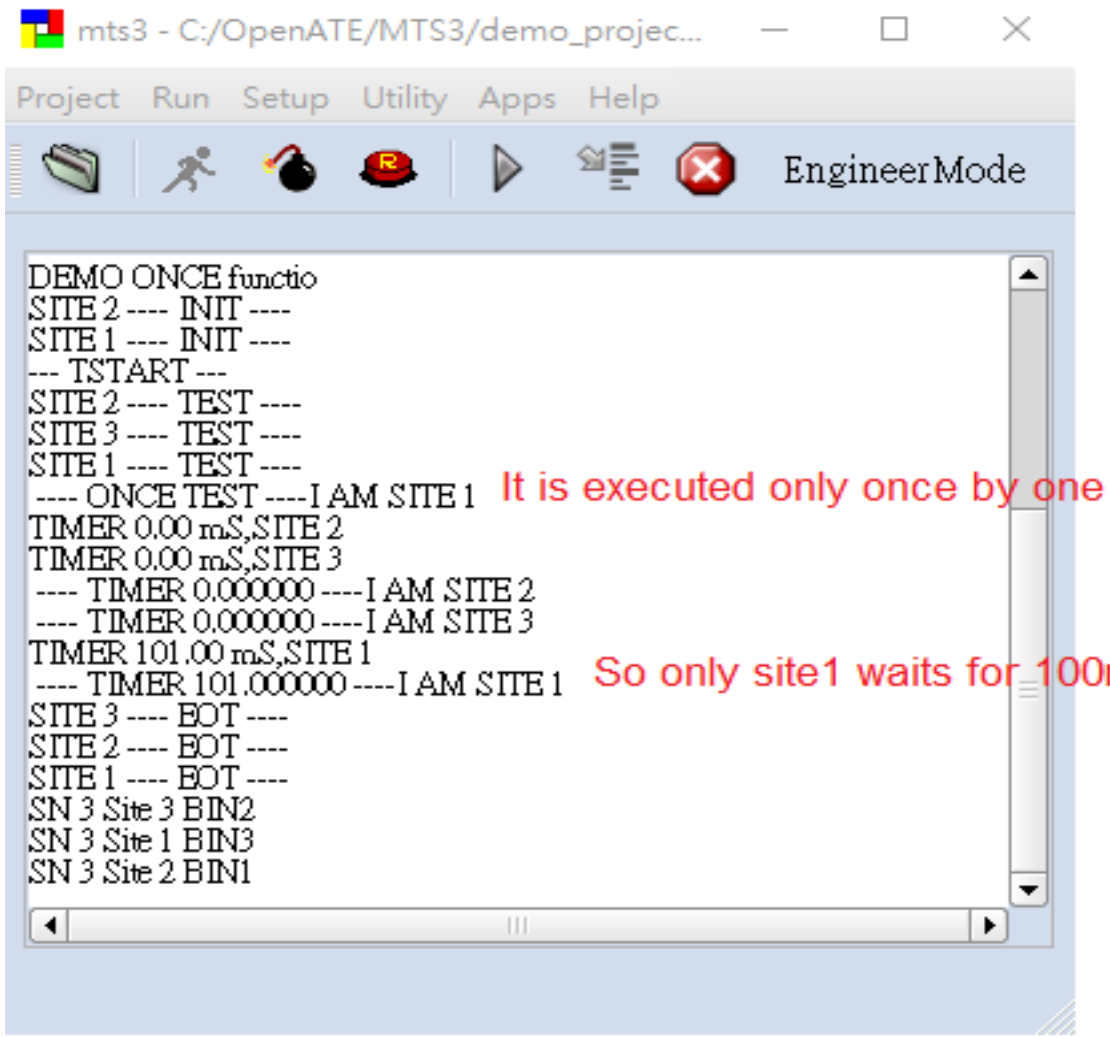
Step4

- Press “Tstart” button to run the test program once.

GLOBAL | INIT | TEST | EOT | PWD | LIB

```
if(mts3_get_datalog())mts3_msg( "SITE %d ---- TEST ----", mts3_getSite() );
if(i>10) i=1;

mts3_start_timer();

if(mts3_once()){
mts3_msg( " ---- ONCE TEST ----I AM SITE %d", mts3_getSite() );
MSleep(100);
}
if(mts3_get_debug())mts3_once();
mts3_msg( " ---- TIMER %f ----I AM SITE %d",mts3_get_timer(), mts3_getSite() );
mts3_bin(i++ );
```

Statements in this block will be executed only once by one thread.



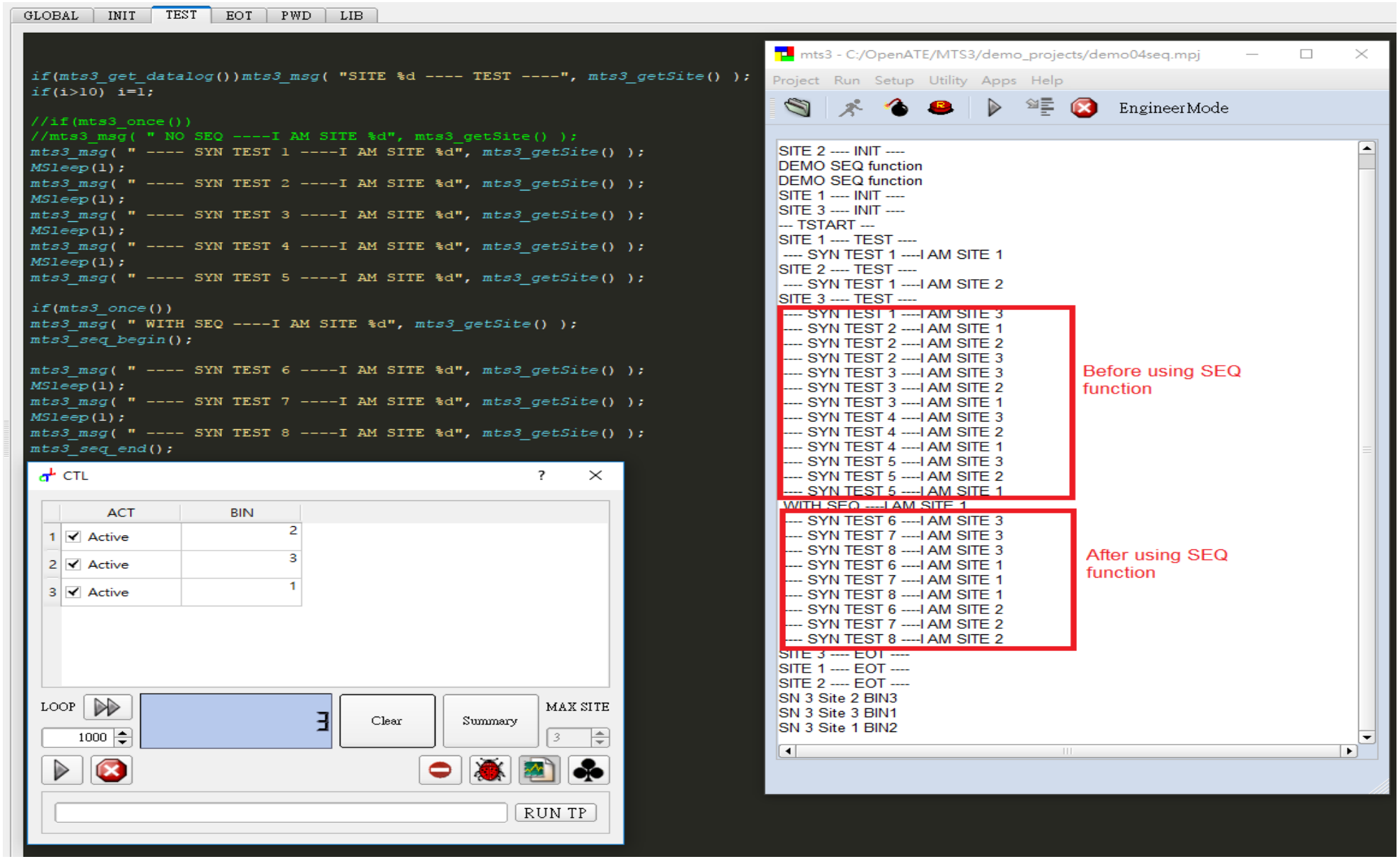
Demo Project4

Step-by-step

Demo Project_4

Demo

-



OpenATE Inc.
The open solution for IC tester

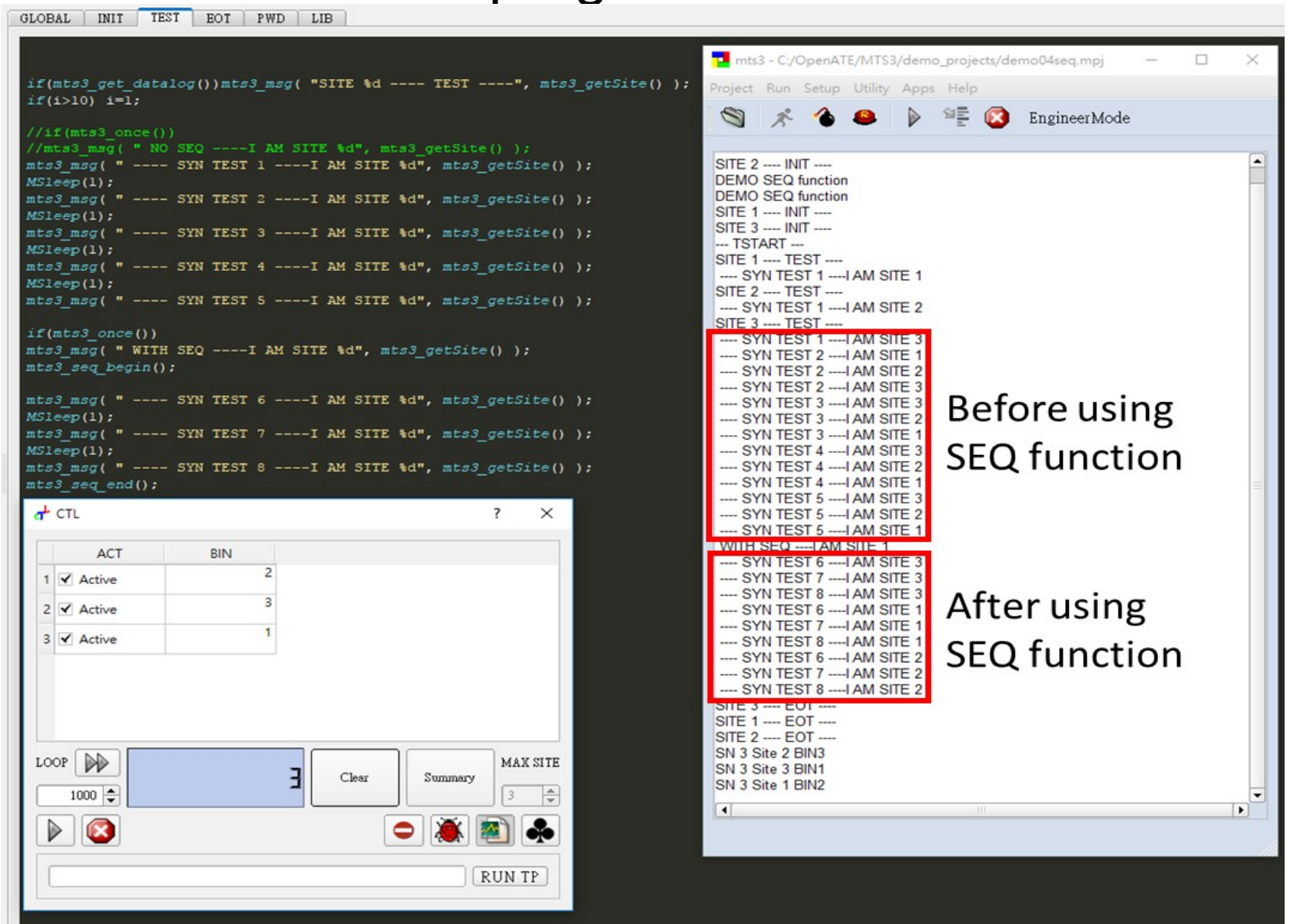
Introduction to API

- void mts3_seq_begin() / void mts3_seq_end();
  - Sometimes for a multi-site application, a piece of code may not able to be executed concurrently by more than one thread at the same time.
  - Usage:

```
mts3_seq_begin(); // ---- Point A
// statements in this block will be executed by one thread at a time.
mts3_seq_end(); // ---- Point B
```

Step4

- Press “Tstart” button to run the test program once.



OpenATE Inc.
The open solution for IC tester

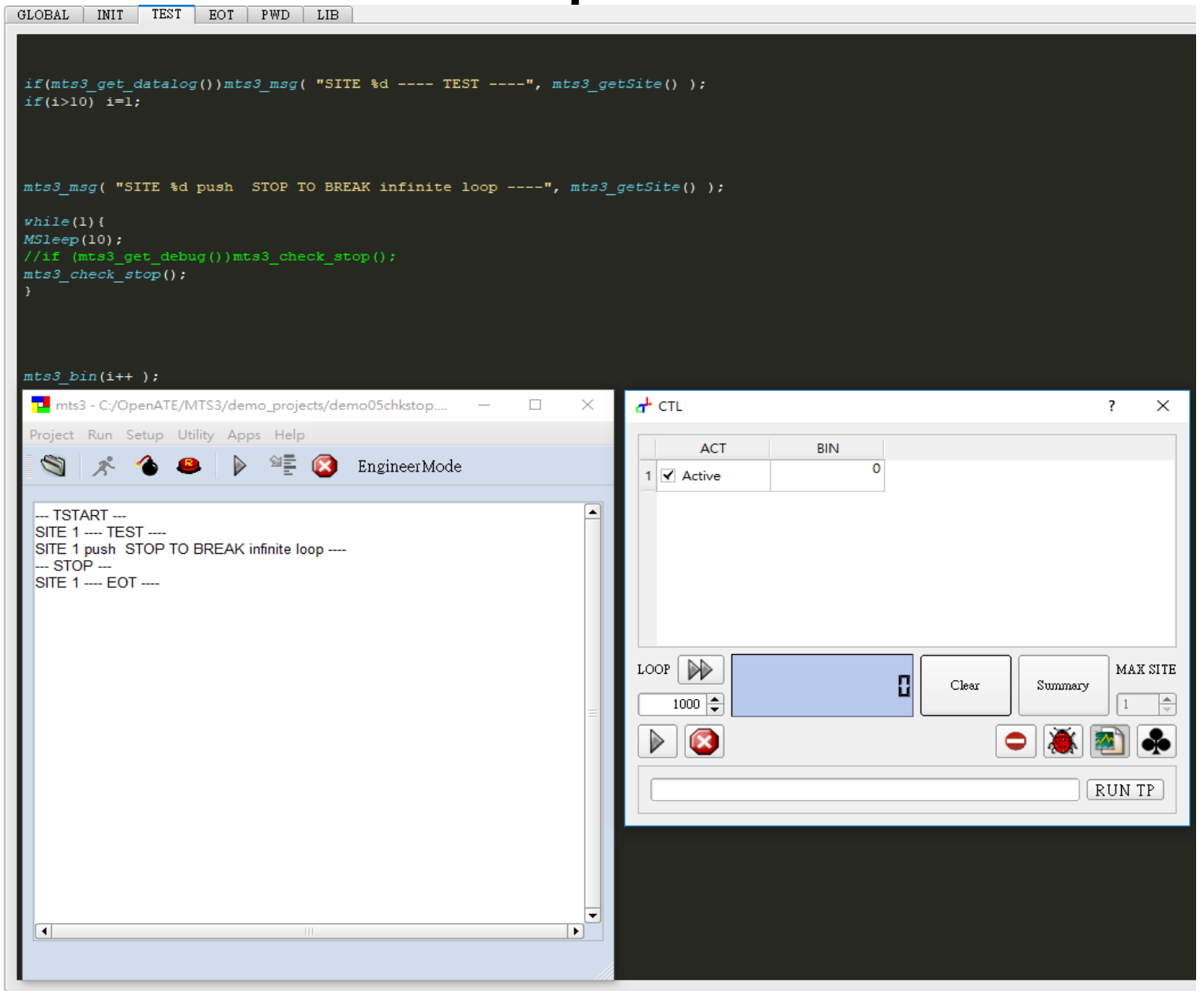
Demo Project5

Step-by-step

Demo Project 5

Demo

-



OpenATE Inc.
The open solution for IC tester

Introduction to API

- void mts3_check_stop();
  - In user's test program, add this function call in a infinite loop, so that system can interrupt the loop when user click 'stop'.
  - while (1) { if (hardware_ready()) exit; }
- Here we put a call to mts3_check_stop() in the loop, so that user can hit stop button to stop the infinite loop:
  - while (1) {
    if (hardware_ready()) exit;
    mts3_check_stop();}

Step1

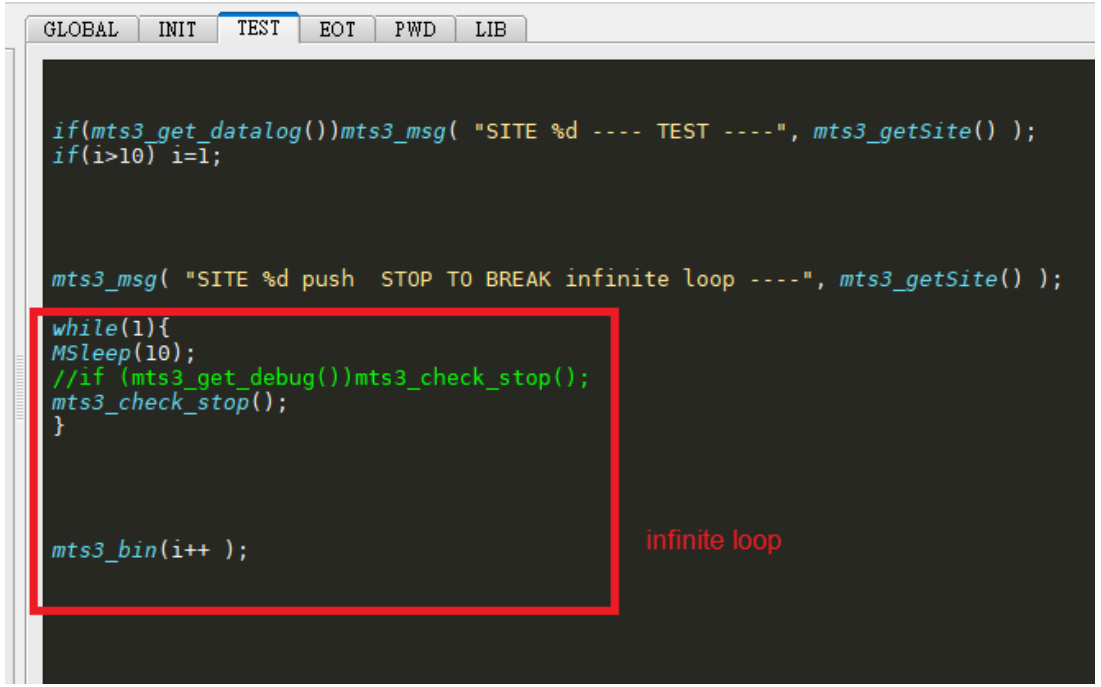
- Open MTS3

Open project “C:\OpenATE\MTS3\demo_projects\demo05chkstop.mpj”

- Open TPBUILD <Utility → TPBUILD >

Step2

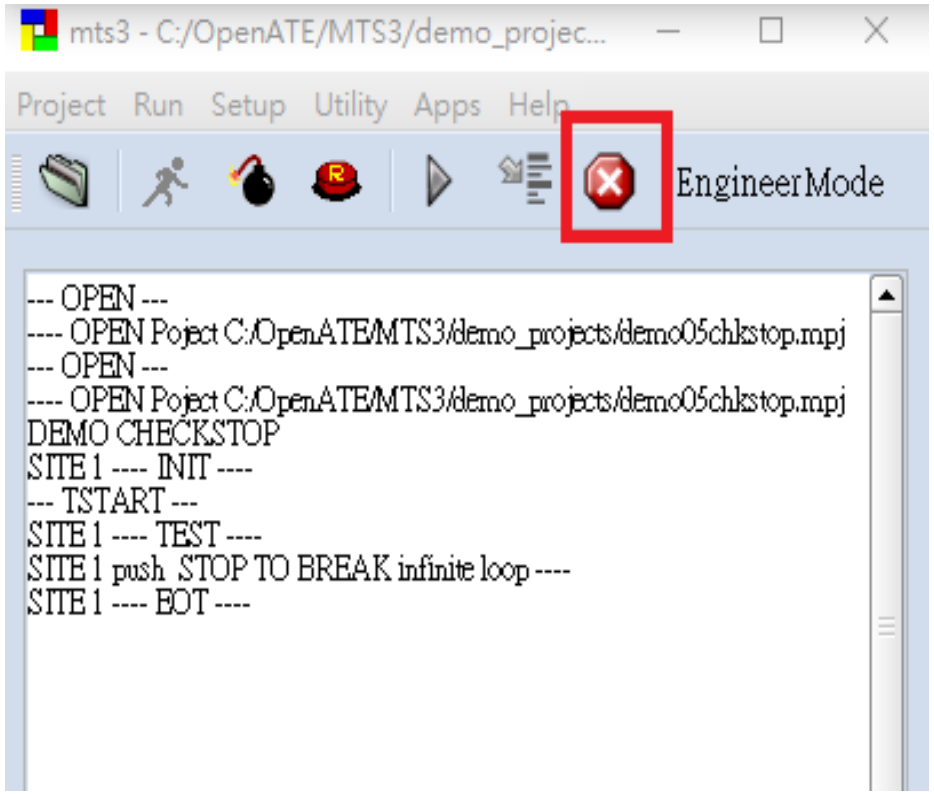
- Press “Exec” button to initialize.
- Press “Tstart” button to run the test program once.



OpenATE Inc.
The open solution for IC tester

Step3

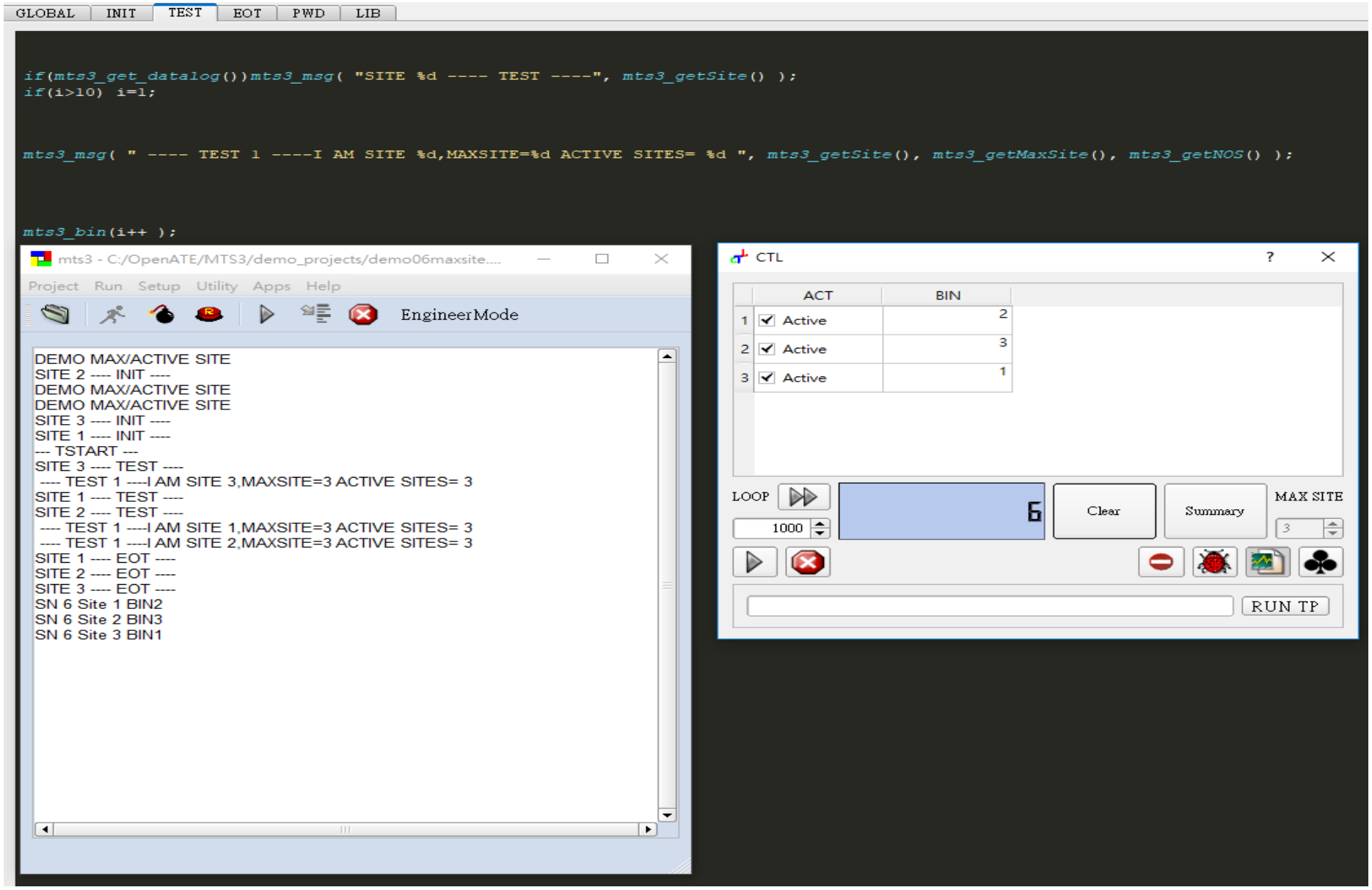
- Push STOP to break infinite loop



Demo Project6

Demo Project_6

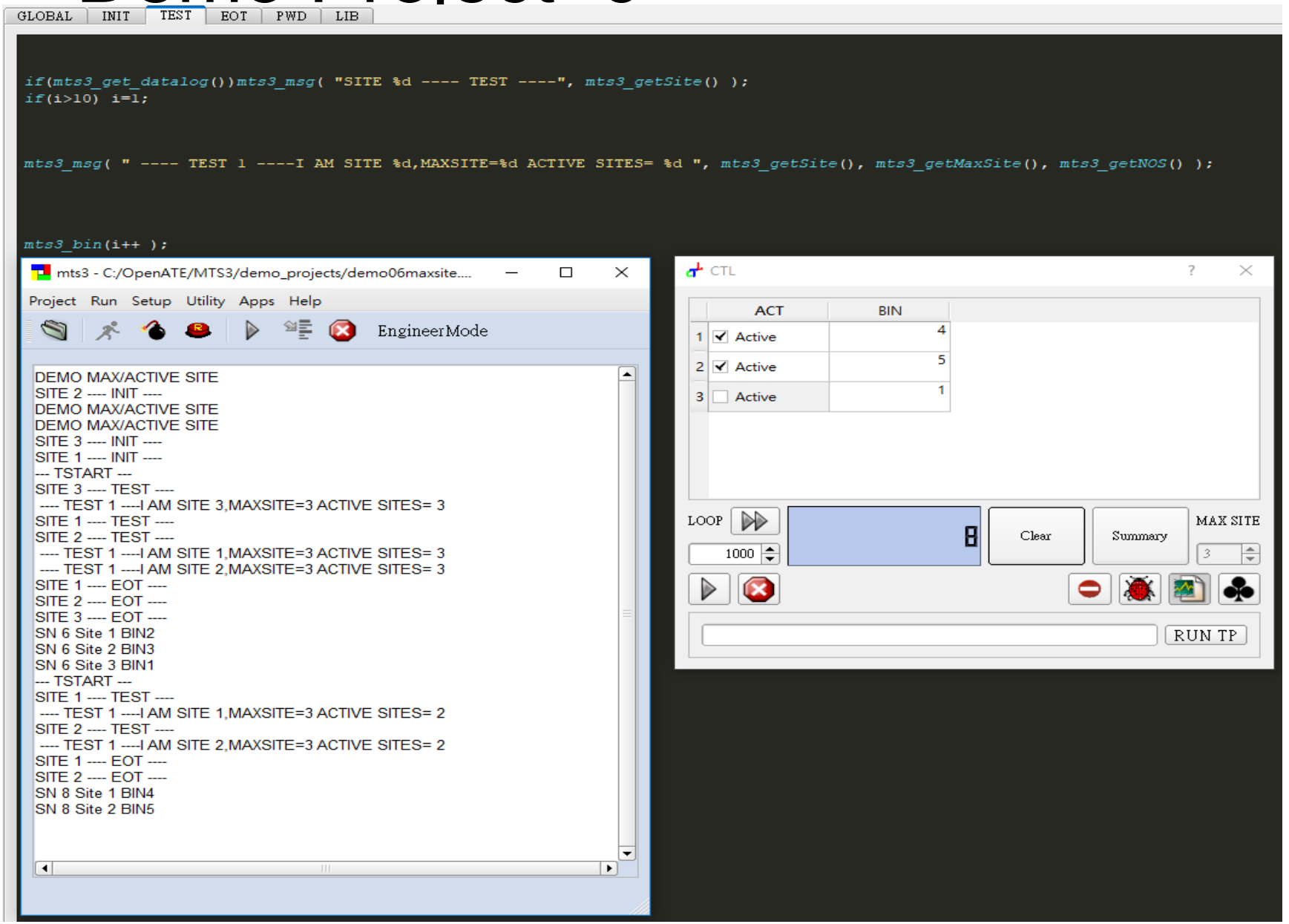
- Setup
  - Three sites
- Max site = 3
- Active site = 3



OpenATE Inc.
The open solution for IC tester

Demo Project 6

- Setup
  - Three sites
  - Deactivate third site
- Max site = 3
- Active site = 2



OpenATE Inc.
The open solution for IC tester

Introduction to API

- int mts3_getMaxSite();
  - Return maximum sites available by system.

- int mts3_getNOS();
  - Return number of sites for current test configuration.

Reference

- OpenATE official web
  - http://www.openate.com/

  mts3
- mts3_user.manual