Tutorial `arduino-cli` and the lauterbach_trace32_arduino_pro debugger

Motivation

Jeremy Ellis produced a video on how to start with the Lauterbach Debugger in the Arduino Portenta world. This video motivated me to show my way I work with `arduino-cli` and the Lauterbach debugger. `arduino-cli` is my favourite tool chain (together with a fancy text editor) for the Portenta development.

Goal

Step by step intro on how to use the Lauterbach Portenta Debugger together with the Arduino commandline interface **`arduino-cli`**, see https://www.arduino.cc/pro/cli

Preconditions

The `arduino-cli` environmet is installed and you know how to work with it.

The Lauterbach debugger is installed and you have also installed a valable licence key. See Sebastian Romeros Tutorial about this task.

https://www.arduino.cc/pro/tutorials/portenta-h7/por-ard-trace32

Environment

Windows 10 Home

mbedos 1.3.0

arduino-cli Version: 0.13.0 Commit: 693a045

The `arduino-cli.exe` is in the path

Arduino Portenta, updated to the actual core and bootloader (dec. 2020)

Portenta connected via USB to the PC

My root directory for Portenta development with `arduino-cli` is

`e:\projects\arduino_cli\portenta>`

Adapt this path to your environment

Step by step

Step 1: Create a Project with `arduino-cli`

Project name `portenta_basic_debug`

Open a command shell in

```
e:\projects\arduino_cli\portenta>
```

Create a new project

```
e:\projects\arduino_cli\portenta>arduino-cli sketch new portenta_basic_debug
```

Sketch created in

```
e:\projects\arduino_cli\portenta\portenta_basic_debug
```

Step 2: Code your sketch

Change to: `e:\projects\arduino_cli\portenta\portenta_basic_debug`

Open the `portenta_basic_debug.ino` file in your prefered editor and insert your code

My example code:

```
#include <Arduino.h>
#include <mbed.h>
#include <ThreadDebug.h>

UsbDebugCommInterface  debugComm(&SerialUSB);
ThreadDebug threadDebug(&debugComm, DEBUG_BREAK_IN_SETUP);

int myLoopCounter = 0;

void setup() {
  delay(100);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  myLoopCounter++;
}
```

Step 3: Compile your sketch

Open a command shell in

`e:\projects\arduino_cli\portenta\portenta_basic_debug\`

Compile `portenta_basic_debug.ino` with `arduino-cli`

```
>arduino-cli compile --fqbn arduino:mbed:envie_m7
```

Now you find all required files under

`e:\projects\arduino_cli\portenta\portenta_basic_debug\`**`build\arduino.mbed.envie_m7\`**

```
portenta_basic_debug.ino.map
portenta_basic_debug.ino.hex
portenta_basic_debug.ino.elf
portenta_basic_debug.ino.bin
```

Step 4: Upload your sketch

Double click the reset button on your Portenta board

Start a commeand shell in

`e:\projects\arduino_cli\portenta\portenta_basic_debug\`

Examine the port your board is connected

```
>arduino-cli board list
```

Result

`COM20` `Serial Port (USB) Arduino Portenta H7 (M7 core) arduino-beta:mbed:envie_m7 arduino-beta:mbed`

Upload your code

```
>arduino-cli upload -p COM20 --fqbn arduino:mbed:envie_m7
```

Step 5: Prepare the debugger

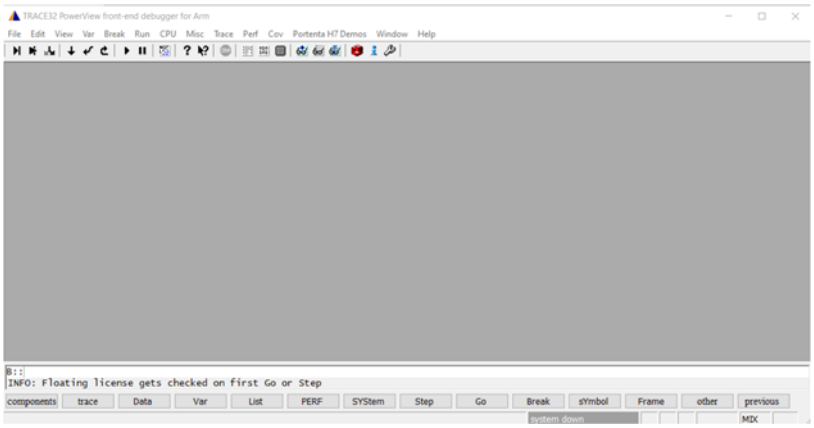
Examine the `portenta_basic_debug.ino.elf` file in the directory

`e:\projects\arduino_cli\portenta\portenta_basic_debug\build\arduino.mbed.envie_m7\`

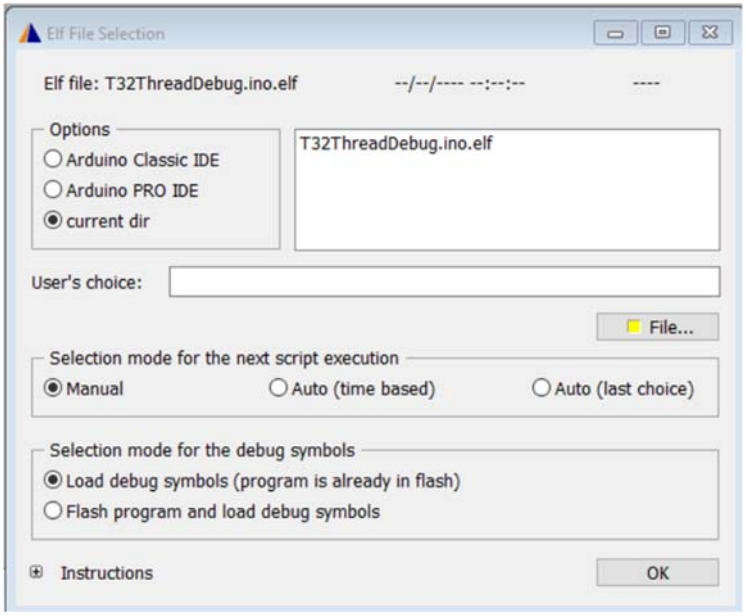
Copy the full file path to the clipboard:

`e:\projects\arduino_cli\portenta\portenta_basic_debug\build\arduino.mbed.envie_m7\portenta_basic_debug.ino.elf`

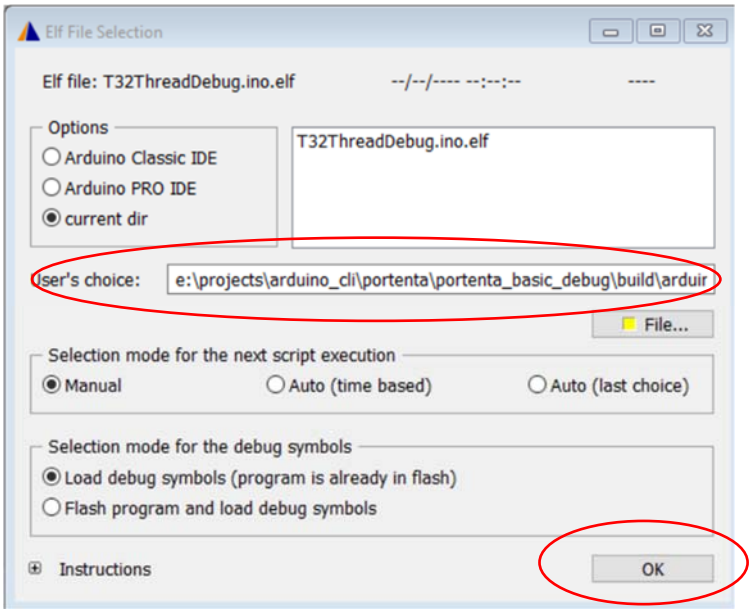
Open the t32marm debugger



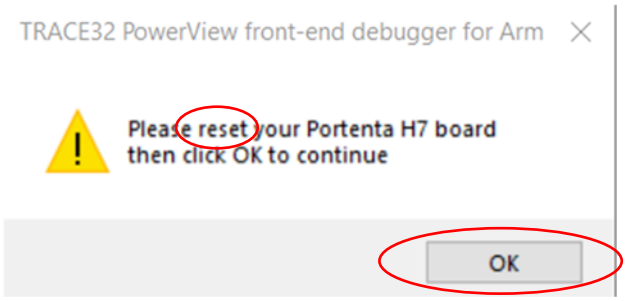
Select the Portenta H7 Demos menue and the T32ThreadDebug sub menue



Paste the full `portenta_basic_debug.ino.elf` file path to the “User’s choice:” field. Set the rest as shown in the screen shot

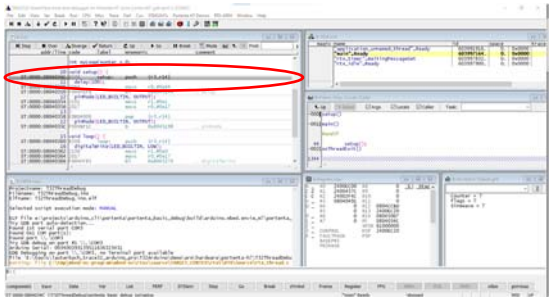


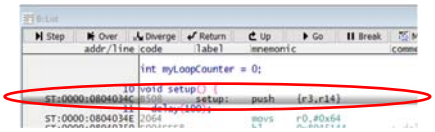
Click OK



Press **once** (**NO DOUBLE CLICK**) the Portenta reset button an then click the **OK** Button

The Debugger starts. The List Window shows the start of your sketch (setup())





Step 6: Go on with debugging and have fun!