

## **How to compile and upload blinking LED to Olimex TMS320-P28016 using Olimex TMS320-XDS100-V3 emulator under Code Composer Studio 6.1.2?**

A couple of important hardware considerations before attempting the connection:

- a) The three jumpers of TMS320-P28016 called "GPIO18", "GPIO29", and "GPIO34" should all be set to position "1" (which means that the board would boot from flash).
- b) The jumper ARM\_JTAG\_E of the emulator TMS320-XDS100-V3 should be set to open position (not connected). This jumper should be closed only when connecting to ARM targets (not the case with the TMS320 board).

The rest of the jumpers of the both boards should be left as per default.

- c) The TMS320-P28016 board needs power supply – use either the power jack or the USB type B connector to supply power.

After a); b) and c) are set, proceed with the software setting:

1. Download and install Code Composer Studio v6.1.2 (during the installation make sure to select and install all packages related to XDS100, TMS320, and C2000 boards). I would also recommend keeping the default install path. By the time of writing this document, download locations of CCS can be found here: [http://processors.wiki.ti.com/index.php/Download\\_CCS](http://processors.wiki.ti.com/index.php/Download_CCS)
2. Download and extract the archive called sprc191.zip which contains examples for TMS320F2801x targets. The archive contains examples suitable for CCS v3.3.x and CCS v4.x.x. By the time of writing this document download location can be found here: <http://www.ti.com/tool/sprc191>
3. Install setup\_DSP280x\_v170.exe (located inside sprc191.zip) - I recommend you to keep the default installation directory again.
4. Start CCS v6.1.2 and select any workspace folder.
5. Select Project -> Import Legacy CCSv3.3 Projects...
6. Click the "Browse" button at the top-right corner of the window that pops-up and point to C:\tidcs\c28\DSP280x\v170\DSP280x\_examples\gpio\_toggle\Example\_280xGpioToggle.pjt – select "Open" and then "Next" and "Finish".
7. Ignore any warnings and messages – click "OK".
8. Right-click over the top level of the project in the explorer (at the left side of the CCS GUI) and select "Properties". A new menu would appear. You need to edit several things here (all are very important if you have some errors remember to return here):

8.1. in the "Variant" drop-down select "2801x Fixed Point"

8.2. in the field next to "Variant" select "TMS320F28016"

8.3. in the field "Connection" select "Texas Instruments XDS100v3 USB Debug Probe"

8.4. in the field "Linker command file:" select "28016\_RAM\_Ink.cmd". This is very important step since the original "2808\_eZdsp\_RAM\_Ink.cmd" of the project is not working properly with the newest compiler.

Refer to picture "project-properties.jpg" to see the options set. After you are done click "OK".

9. Right-click over "2808\_eZdsp\_RAM\_Ink.cmd" and select "Exclude from Build" (you can also delete it). If you forget to do this you would have a lot of errors and warnings later since there would be two cmd files and duplicate memory definitions in the project.

10. Right-click over the top level of the project in the explorer (at the left side of the CCS GUI) and select "Properties" again. Navigate to the sub-menu "C2000 Compiler" and click on "Include Options" delete the search path for "xdais" - refer to the picture "xdais-warning-fix.jpg". Click "OK". This fixes two warnings related to the "xdais" library which is included in the project but not really used.

11. At this point you can compile the project successfully (with one warning - "entry-point symbol other than "\_c\_int00" specified: "code\_start" ", which you can also fix if you want – it is related to the memories and requires some reading). You can also debug the code on the board via the XDS100v3 emulator. Yet, you will probably think that this is not a real blinking LED since the LED appears to be constantly turned on. This is because it blinks very fast. To fix it we do the following:

11.1 Double-click over the main file "Example\_280xGpioToggle.c".

11.2 Navigate to line 140 where you see delay\_loop() function. This is the function we need to edit in order to get a better blink. Originally it looks like this:

```
void delay_loop()
{
    short    i;
    for (i = 0; i < 1000; i++) {}
}
```

We should edit it to look like this:

```
{
    long int    i;
    for (i = 0; i < 1000000; i++) {}
}
```

12. Build the project and debug it. Use the buttons at the top to "Resume" or "Pause" the executing of the code or inspect with "Step Into" or "Step Over". Refer to picture "blinking-led-debugging.jpg". The on-board LED should blink when you have pressed F8.