Chapter 6

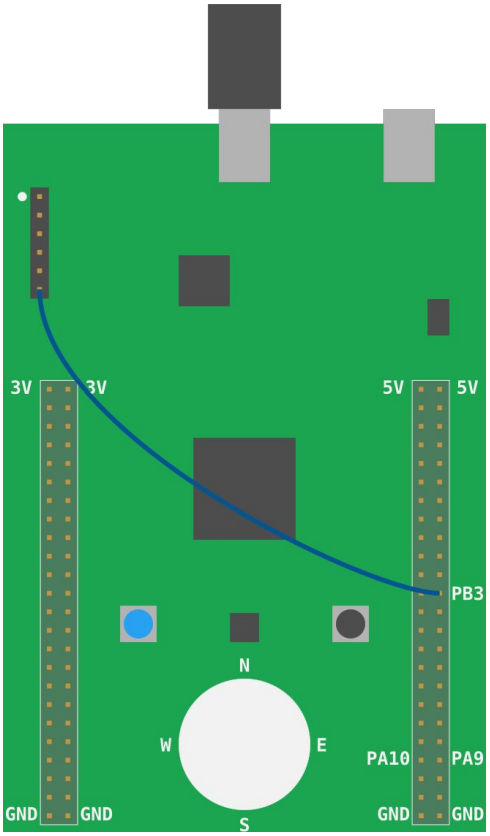
Hello World from F3

Hello World

- ➔ In this short lesson we will see how to send message to debugging host from F3
- ➔ ITM (Instrumentation Trace Macrocell) is responsible for this message sending/communication
- ➔ ITM is a communication protocol
- ➔ It's a unidirectional protocol that can be used to send message to debugging host but can't send message to MCU from host
- ➔ Being the debugging session manager, OpenOCD facilitates this message passing, received message and forward it to a file.
- ➔ **Use-case** : Application status update, logging, etc

Hello World

- ➔ To enable ITM we have fix an issue
- ➔ Connect SWO and PB3 pins (as shown)
- ➔ Once it’s done you are ready to code for this exercise



Hello World

Sample Code:

```
#![deny(unsafe_code)]
#![no_main]
#![no_std]

#[allow(unused_imports)]
use aux::{entry, iprint, iprintln};

#[entry]
fn main() -> ! {

    let mut itm = aux::init();
    iprintln!(&mut itm.stim[0], "Hello, world!");

    loop {}
}
```

Extra Work

After writing code and making an explicit connection few addition... things will also needed to be done.

1. First thing first, we have to install **itmdump i**f didn't installed in the initial lesson of Installation and configuration
2. After building and before executing program we have to open an additional(beside OpenOCD's terminal) terminal in **/tmp** directory
3. Creating a file (i.e. itm.txt) in /tmp directory
4. After creation of file creating a watch on the file (created in step 3)

Extra Work

5. Next we will build the program, flashing it and perform steps ... the point where our program stops at first breakpoint.
6. After that we will execute 2 additional commands specific to ITM
   - **(gdb) monitor tpiu config internal itm.txt uart off 8000000**
   - **(gdb) monitor itm port 0 on**
7. Next we will execute our program and once we passed this line:

```
iprintln!(&mut itm.stim[0], "Hello, world!");
```

8. We will see that on terminal we opened for itmdump, a new line appended saying "**Hello, world!**"

Panic!

- Not only **iprintln!**, but we can also use panic! macro
- Panic! macro is used for forcefully terminating the program in case of any undesired behavior of program.
- Sample code:

Summary