**4 Foundations of the embedded programming**

**4.1 The working environment**

Programming the piccolo f28069 was not an easy task as it required direct register reads and writes in order to start working. There were no pre-implemented function definitions in this model as it was in its more expensive forebeareers. Therefor, the team needed to either start the programming sequence on bare register addresses or to look harder.

Fortunetely, there were some examples given by Texas Instruments for theese kind of events. In thoose instances the developers made the processor do many things, including an ePWM module configuration and such.

Digging deeper into the software model presented, a register mapping relying on section definitions in memory was discovered. Such a configuration can be – between reasonable frames - used for easing a developers work, as one does not need to accost every register by their address. The software model in this project is built up on theese foundations as previously defined typedefs were also accessible and used. They were linked to the corresponding memory section through the compiler.

Furthermore, the examples provided numerous .h and .c files for commonly used functions and variables, dedicated to each and every component the processor has. Using theese helping hands, the project library was expanded with includes and source files. Finally, a blank project file was added to the working set for implementing the main function.

Later, maintaining a clean order in the main file, dedicated .c files were created for each used pheripheral, and a shared .h file for their function and variable defines. This way for instance the initialization of the pheripherials were coded elsewhere, not in the main file (it only contained the corresponding function call), thus leaving it more transparent.

**4.2 System Control and Interrupts**

As mentioned in section 1it works this and this way…