

Step 1

- What can we say about readability of this code?
What are the skills required to write such a program?
 - Regarding the application domain, could you characterize the expressivity?
The configurability of the code to change pins or behavior?
Its debugging capabilities?
 - Regarding the performance of the output code, what kind of parallelism is expressed by the use of the DDRx registers?
 - What if we add additional tasks in the micro-controller code, with the same frequency? With a different frequency?
-
- The readability is not good, mainly due to the manipulation of the PINs using vectors of bits. The skills required are not very high provided you can write C and have the right documentation regarding the ports configuration and usage.
 - The expressivity is the highest because we have access to all the possibilities of the machine, provided we have a documentation for all the system primitives. The configurability is very high The debugging is difficult due to the operations being low-level: if the program does not have the desired effects it could for example be due to a bit that was written to the wrong value, or a write operation that was not conducted due to a wrong writing mode and which would fail silently; all errors which are not on the same conceptual level as the desired output.
 - The pins of a single register can all be modified at the same time with binary operations, which allows to jump from any state of the register to the other. This is a type of memory parallelism which affects the state of the components.
 - It is possible to add any number of tasks on the same frequency due to the sequential nature of the code. Adding tasks with different frequencies would be difficult because it would require computations in order to synchronize different clocks. For example if we have a task T1 every 200ms and another T2 every 500ms, a possibility is to wait 100ms every task T1 in order to do T2. This is not scalable as is.