In this paper we present an approach to mapping and scheduling of distributed embedded systems for hard real-time applications, aiming at a minimization of the system modification cost. We consider an incremental design process that starts from an already existing system running a set of applications. We are interested to implement new functionality such that the timing requirements are fulfilled, and the following two requirements are also satisfied: the already running applications are disturbed as little as possible, and there is a good chance that, later, new functionality can easily be added to the resulted system. Thus, we propose a heuristic which finds the set of already running applications which have to be remapped and rescheduled at the same time with mapping and scheduling the new application, such that the disturbance on the running system (expressed as the total cost implied by the modifications) is minimized. Once this set of applications has been determined, we outline a mapping and scheduling algorithm aimed at fulfilling the requirements stated above. The approaches have been evaluated based on extensive experiments using a large number of generated benchmarks as well as a real-life example.