

Welcome to the OhDamnGitUp wiki!

What and How we will build it

We are going to build a clock using the raspberry pi 0w, and using a web interface to set alarms and communicate with the device.

We are going to build this by separating the task into key groups,

- the time keeping and alarm checking
- web interfaces
- hardware
- documentation, testing and general do it all man

Each person will be responsible for their position and while there is no documentation or testing to do in the main system(hah), the tester can help keep everyone on track for the goal completion time of 18th may, leaving us a week or more to cover for unexpected problems as well as test the system more. The system should be made the best it can be by assigning people into the tasks available based on their skills. Members of this team have varied skills from each other such as web development experience, pi hardware experience and so on. Making each job have a person best suited to it, this should help us to ensure quality.

As far as implementation goes we will have a web interface built to take alarm settings whether to remove or add alarms, and save them to a communal file in an appropriate format. The alarm system itself will use the pi's operating system to keep track of the time and check against the list of alarms. The alarm section of the clock will operate by doing a somewhat recursive check-then-wait loop. It will start by reading the alarms into memory then checking the time of each one vs the current time, if it is within a period after the current time it will call the alarm checker again with a half sized period until it is under a minute then call said alarm after a short wait. The system will also be outputting the current time to screen output so that the user can constantly see the time. This system will mainly written in c++.

We will be ensuring build quality by maintaining one position on the team to be the somewhat everyman, a person to keep tabs on what everyone's doing and how the whole things kinda works. This position will be vital as the complexity increases and there are more than one set of eyes on all the code to help with debugging, and advanced program interactions. This program should not take too long to produce, but

with its essential nature a slightly greater than average amount of testing will surely be required. As earlier stated the 18th of may has been set as our deadline for the first fully functional version of the program.

Marketing

This clock has the target market of students, an alarm clock goes a long way for students who spend all night "studying" and still have to wake up for those annoying 9am tutorials. Having this alarm in particular be more customization than the average as well as more accessible through the web interface, it may take getting use to by the user, but being able to set the alarm without having to physically go to the alarm could be very appealing.

Every day use

The use of an alarm is an everyday thing for many people. To have a device that's sole purpose is to make sure you get up at the right time, or even just to alert you of the time so you can start or finish something is extremely helpful in daily life. Member of our group already use phone based alarm clocks in order to make sure we get to uni in time so we can guarantee we will be using this not only weekly but daily.

diagram of how the whole thing will talk to each other(basic service overview)

