One Clock To Rule Them All Design Document

Tristan Hari, Tasman Thenell, and Scott Metzsch

Abstract

The purpose of this document is to provide an in-depth analysis on each of the three sections that were written for the tech review. More specifically, for our project our three main categories were Software, Hardware, and Case. This document will include the entirety of how our project will work, using several viewpoints from the IEEE 1016 standard. The use of this document will be as a road map for how the rest of your year will play out, so in turn we will make it with the notion that it will be revisited many times throughout the year.

For this review each of us were responsible for writing about 3 different sections. Here are the assigned sections for each individual.

Tristan Hari

- Micro Controller
- Letter Grid Layout
- Testing

Tasman Thenell

- Light Emitting Diodes
- Power Source
- Existing Software Libraries

Scott Metzsch

- RTC
- Fabrication Tools
- Materials

A. Micro Controller

The technology we are using for our micro controller is the MPBLABXpress Evaluation board. The reason why this will be our sole selection is simply that it is what was chosen for us to use by our client. That being said it is an excellent choice. There are many projects that have been created using this micro controller, from things like other clocks, to home monitoring systems. It has many pins for usage and runs very efficiently with power. The creators also provide a cloud based IDE [1] that is very easy to use and load new programs onto the board with, which is a feature exclusive to the board we are using. A few of the competing products on the market right now is the Adafruit trinket [2] mini micro controller, and the infamous raspberry pi [3]. The Adafruit device is excellent as it is very small and can do all of your favorite 5v logic programs. The only issue faced with this device in the light of our project is that it does not have enough power throughout capabilities to properly control the LED array that we will be using. Also it would require some sort of board to be placed on, and that would lead to additional hardware needing to be bought, and potentially increasing the size past where we would have liked it. The raspberry pi is also an excellent product. There and thousands of project examples that are out there for the pi as is. The only reason we decided not to proceed with the raspberry pi is mainly that is more than what we need. The raspberry pi is an entire tiny computer that can run Linux flavors and do many things. But with that in mind, they are much more expensive as are the parts to expand the capabilities.

B. RTC

We are working to create a clock that will tell time with words instead of hands or numbers. To keep track of the time we will be using a real time clock module that is powered by a battery to keep track of the time so that even if the clock loses power time will still be kept track of. We will be using the RTC module to know when to update the face of the clock to show a new time.

For our project Victor has supplied us with 3 KeyeStudio DS3231 Clock Modules. The RTC is powered by a small battery that will keep track of time even if the main power source is turned off or disconnected. The main benefit of this clock is that is has an integrated temperature-compensating crystal oscillator and that there are pins we can attach wires to instead of having to solder each wire to the board. In RTCs time is kept by using a small quartz crystal that has a resonance of a precise frequency. This frequency is used to keep track of time and there can be small changed based on the temperature that the crystal is at. In our RTC there is an adjustment made that will allow us to get more accurate time keeping when the RTC is within recommended temperatures. Between 0 degrees Celsius and 40 degrees Celsius the clock is accurate to 2ppm [4], or in other words it will have two errors every 1 million seconds. With this accuracy we will only lose around 0.1728 seconds per day or 63.072 per year.

An alternative to the KeyeStudio RTC is the SparkFun Real Time Clock Module. Compared to the KeyeStudio RTC the SparkFun RTC has two main weaknesses, the first is that the clock is not have any crystal temperature correction and does not have pins to connect wires into. Not having crystal temperature correction has main issues when you have your clock changing temperatures with the weather and seasons. The SparkFun RTC has an advertised accuracy of 20ppm which is 10 times the errors that you have in the KeyeStudio RTC. A 20ppm error will lead to 1.728 seconds loss over a day and 630.72 or 10 minutes 30 seconds over the course of a year. This lack of temperature correction is a strike against the SparkFun RTC. The second weakness of the SparkFun RTC is the lack of built in pins for connecting the RTC to the logic board. The intent of the SparkFun RTC is to solder wires onto the RTC and then to the logic board to connect the two parts, but with SparkFun RTC being only 20mmx20mm and roughly the side of a quarter [5] soldering 5 wires onto the board can be slightly difficult. Also having pins instead of soldering wires to boards makes it easier to fix mistakes make connecting the correct wire to the correct place on the logic board.

A third option for the RTC could be the SparkFun DeadOn RTC Breakout, this RTC has some advantages over the SparkFun Real Time Clock Module but still is behind the KeyesStudio RTC in terms of ease of instillation. The SparkFun DeadOn RTC is similar to the KeyesStudio RTC since they both have a temperature-compensated crystal oscillator to keep track of time, this reduces the error to 2ppm [6] from 20ppm with the non-temperature-compensated SparkFun RTC. But where the SparkFun DeadOn RTC is still behind the KeyesStudio RTC is in connectivity. With the DeadOn RTC you still need to solder wires onto a board the size of a quarter, but instead of 5 wires there are 7 connections on the DeadOn RTC.

For our project Victor made a great choice for the RTC. The KeyesStudio RTC is accurate, easy to connect to the board, and the cheapest of the 3 boards coming in at $6.80 on KeyesStudios store page. With the KeyesStudio RTC winning in all 3 categories that have importance to our project, (connectivity, accuracy, and price), it is the correct RTC to choose for our project.

C. *Fabrication Tools*

To create the case of our clock we have many tools, but the main 3 we are looking at using are a 3d printer, laser cutter, and CNC. Each tool has its strengths and weakness that will impact our decision on what to use for all the different parts that will go into our clock.

The first tool we can use for our project is a 3d printer. Most modern 3d printers use either ABS (Acrylonitrile Butadiene Styrene) or PLA (Polylactic Acid) which are both thermoplastics. 3d printers create objects by melting down this plastic and applying layer after layer to form the walls and internal lattice of the object. In the settings of the 3d printers you are able to change the thickness of the walls and pattern that the machine prints for the hollow inside the object. Both the wall thickness and internal pattern determine the strength, weight, and robustness of the object as well as increasing of decreasing the time it takes to print the object. Each 3d printer also has a limitation that comes from the size of the printing area. We imagined our clock to be around 12 inches by 12 inches and most 3d printers would not be able to print pieces that large and if they were able to it would take a very long time to print. 3d printing can be used in our project for any parts we may need that are not easily found at a hardware store or are not flat objects. We envision possibly using the 3d printer to help create mounting point for the micro controller, guides for LED strips, or other small parts we find cannot find at hardware stores.

The second tool we are able to use for our project is a laser cutter. Many laser cutters use a laser beam and focus the beam to create a focus point that can melt, burn, or vaporize the material at the focus point. The process is similar to using a magnifying glass to focus the sun onto burn objects. Laser cutters are good for cutting acrylic, wood, and if powerful enough can cut metals. Consumer laser cutters like the ones we have access to are usually limited to cutting plastics and woods. Since laser cutters cut using heat and concentrated beams there is a great deal of heat generated during the process and this needs to be accounted for when choosing materials and thicknesses of materials. Thicker materials or harder woods need to be cut at slower speeds and thus cause the material to heat up more and can leave burn marks and can warp the material you are cutting. There is some marking techniques and cooling tools that help to reduce the amount of heat and burning that can affect the cut of a material. We envision using a laser cutter if we want to make the housing of our clock acrylic or out of a softer or thiner pieces of wood.

The last tool that we have access to is a CNC (Computer Numerical Control) router. CNCs use specific types of drill bits to cut materials out and create 2d or 3d shapes. CNCs are very similar to laser cutters in their cutting process but CNCs have the benefit of generating less heat to make to remove material. Similar to the laser cutter the cut speed depends on the hardness of the material and the speed and depth of the cut need to be adjusted accordingly. If your material is a soft wood like pine, you will be able to have each pass of the CNC be faster and/or deeper compared to if you are a harder wood like maple or oak. We envision using the CNC router to create the face and back of the clock. It will be a good tool to cut out each letter on the face of the clock and for removing material from the back to isolate the light from each letter.

Each tool that we have access to can be used in our project and each has strengths and weaknesses over the other tools. We will use either the laser cutter or CNC for most of the large pieces of the clock since they are faster than the 3d printer, and the 3d printer will be perfect for small pieces or mounting hardware of electronic components.

D. Materials

For our project we have 3 possibly types of materials to use for the case of our clock, acrylic, wood, and ABS/PLA. Each material will have strengths and specific uses in our project.

One possible materials we can use in our clock is acrylic. Acrylic is a plastic that you can purchase in sheets or different colors, varying thickness, and physical characteristics. Proposed uses of acrylic are the face of the clock and case of the clock. Side panels and the face of the clock would be easy to cut with either the laser cutter or the CNC. We could also choose an acrylic that is opaque and this will help to isolate light from the LEDs from bleeding into other letters on the clock. If we wanted to show off our cable management and electronics that go into our clock, transparent acrylic can be a great choice. Acrylic can also come in mirrored sheets that can create unique ways to display the words on the face of the clock. The one concern with using acrylic is that mounting objects to it could be a little difficult. Mounting parts and components to an acrylic clock case would have to be done with glues since drilling and screwing into acrylic has a tendency to crack or split the material if done improperly. Acrylic is also versatile in that we would be able to cut it with either the laser cutter of using the CNC if we find the correct bit and are careful not to crack the sheet.

Wood is another material that we have thought about using to create the case of the clock. There are many options and choices of woods that we can choose for this project. Cheaper or softer woods are a good choice for prototyping stage of the project since they are easier to cut and decrease the price of each prototype version. Hardwoods are a good choice for the final product because of their durability and aesthetic appeal. The downside is that the cut time of hardwoods will be greater and the price can be more expensive compared to softer woods. Woods will be easier to mount components into since we are able to drill into the woods with more confidence that the material will not split or crack. Depending on the hardness of the wood we will be able to use either the laser cutter or the CNC to cut out the case of the clock.

ABS and PLA are popular thermoplastics used in filament 3d printers and can be used to print out small parts needed for our clock. These plastics are heated up by the 3d printer, extruded, and then cool back to a solid state. ABS filament is better when strength, flexibility, and higher temperature resistance are important to the object, while PLA has higher printing speeds, thinner layers, and sharper printed corners. We intend to print small parts and mounting hardware for our clock so either ABS or PLA would meet the requirements we need of the part.

For the case of the clock we will be either choosing wood or acrylic for the body and face of the clock. In both scenarios we will use a cheaper material for prototyping of the clock and then choose a more premium material for the final product. ABS or PLA will be used for mounting hardware and any other small parts that we need to create our clock.

E. Letter Grid Layout

When looking at the grid layout there are infinite possibilities to pursue for what could be considered optimal or the best for our design. That being said we will be comparing a few existing layouts for their potential of different displays. The first one being the most popular, and one of the basis ideas for our project, the QWLock [7]. The way their layout is, is to have each row with cascading potential priority in sentence structure to tell the time correctly. So they have created in layers the exact words needed to spell out the correct time in the correct order. Also They have whole words connected, not leaving spaces between letters. This was a design decision to make the end product more legible but less apparently random and scattered looking while unlit. A second grid design was one we found on an Adafruit tutorial page [8] that was centered around a micro layout. This one was only 8x8 layout, using the absolute minimum words and letters possible to communicate the time. This was a creative design, however it loses the ability to tell time in five minute intervals, which would inherently break one of our requirements, so we will not proceed with a letter grid like this one, and will require more space than 8x8. Another layout design was by Mike Gualtieri [**?**]mikegrid), who listed out some instructions about how to do a layout with any words of your choosing, and how to organize them in a good hierarchical fashion. This is likely to be the method we will use to proceed, as following Mikes layout algorithm is not only open source and free to apply, but allows us the freedom of word choice and layout. The only constricting piece to this method is that it doesnt strictly adhere by any means to letter by letter layout, or by ending in a square grid. To achieve that may require some additional work on the method.

F. Light Emitting Diodes

One of the crucial parts of the clock is providing an interesting and legible clock face. In order to achieve these goals, light emitting diode (LED) backlighting has been chosen to illuminate the letters of the clock face. Using LEDs isnt as simple as just choosing a type of light bulb, there are a strict set of requirements the final selection will have to meet. To this end, we have selected three types of LEDs to test against the following requirements.

The LED requirements center around brightness, ease of use, flexibility, and power requirements. The display is required to be readable at 10 feet but without using an amount of power that would require adding additional separate power supplies. In addition, to facilitate being a programmable display, each LED needs to be separately controllable in a way that isnt overly complicated. Lastly, flexibility is desired as several stretch goals call for various color capabilities as well as brightness control.

The first type of LED under consideration is a small LED strip purpose built as a back light. An example of this product can be found on Adafruit with product ID 1626. [9] The entire module is a thin LED sandwiched in diffusing material that creates a small strip of light rather than a single distinct point. These strips score well in power requirements, using only 20mA at around 3 volts but dont provide many of the other desired features. These modules dont contain any internal brightness control, lack the ability to produce different colors, and would require the creation of a custom addressing system to allow for individual programming.

The second type of light being reviewed is a programmable RGB seven segment display (Adafruit ID 1399). [10] These units provide several features that would be helpful for the project including flexible color and brightness. The color capabilities are particularly intriguing because one planned feature is a having the clock face have a gradient of colors to shift through based on time of day. The other specifications of these display units pose some of the same problems as the LED backlight described above. While these have individual control capabilities, they require twenty one pins for a full range of control per display unit. In addition to the complicated wiring, a separate control unit would need to be included to control these units. Lastly, the power requirements are higher than single LED solutions because these units have seven times as many LEDs to power.

The last type of display unit under consideration is a programmable LED module. One common example of this type of unit is the NeoPixel from Adafruit. [11] These units provide many useful features like a unified programming interface with an internal driver which allows for full color and brightness control with a large number of pins. One very interesting feature is the ability to tie multiple units together and control them all individually through a common bus. Each module uses four pins to communicate with the controller. Fully programmable brightness and red blue green color spectrum round out a tempting set of features.

The project team led to agree with the product suggestion provided by the client. Programmable LED modules with internal drivers are the best solution for powering the clock display. Unparalleled ease of use and flexibility fully meet the proposed feature set of the display including color schemes and variable brightness without requiring a standalone additional controller for the display lighting. In order to get the most out of these units, the team has already acquired several samples to begin experimenting with to learn the API.

G. Power Source

Two separate avenues are being considered for powering the clock project. The most simple approach would be a fixed permanent wall connection. While this would work, most clocks that can function as wall clocks operate off of battery power to prevent the need for visually distracting wiring.

The final power specification isnt yet known for the project but based on early projections and the hardware choices for LEDs and microcontroller, the target power supply needs to provide five volts at several amps. The exact amperage isnt known at this time due to the display resolution and subsequently the number of LEDs not yet being known. This isnt a primary concern as all of the following products and solutions for power have some degree of flexibility or come in multiple power ratings.

The first technology under consideration is a permanent fixed wall connection as the power supply. Since the microcontroller and several of the best contenders for power supply regulating boards use micro USB, it would be possible to tap the existing range of micro USB wall adapters meant for charging android phones and products like the Raspberry Pi. Exact power requirements arent a concern as these adapters exist in a wide range of amperage ratings and specifications.

The first potential battery based power solutions is also based on an existing range of hardware. Like the five volt power adapters mentioned before, there exist a very wide range of five volt battery packs designed for use as portable phone chargers and universal usb power sources. These portable battery packs come in a variety of dimensions and power capacities. Of course these wouldnt be the only answer for power but such packs can be charged via standard USB wall adapters which would be simple to include.

The other battery based solution being considered is to design a custom solution using an array of batteries and prebuilt voltage regulating hardware. An example of this type of power supply setup can be seen in the BattBorg, an addon designed for the Raspberry Pi. [12] This option would be the most flexible but would require the largest amount of hardware development to deploy.

Unlike other categories of the literature review, the only definite selection being made at this time is the use of a standard microUSB power source. While battery power would provide an appealing feature for a wall clock, the feasibility of battery power is currently unknown as the LED arrangement and overall system power draw arent yet known. Without such numbers, figuring out a battery based solution for power isnt possible. In addition, we have already reached an understanding with our client that battery power would be a stretch goal rather than a main focus of the project.

H. Existing Software Libraries

While the physical appearance of the clock is very important, all of the timekeeping functionality and the ability to display the time is powered by a software library running on the board. In this case, the chosen board is an MPLAB Xpress IDE board which uses a combination of direct hardware control and c programming to control it. Since the board supports C and C based assembly, the selection of software available is very large. A homemade word clock has been done before many times before and we will be discussing the capabilities provided by three different software implementations of the word clock functionality.

The first version of word clock software is written in C for Ardruino by Doug Jackson and Scott Bezek. Their software, released under the GNU General Public License, is most of a starting point than a library. [13] This software provides nothing more than the core function of a word clock including predefined word sets and corresponding LED control, as well as simple controls for changing the time. The library lacks any other advanced features and only drives a single color of LED. Another setback is the almost complete lack of documentation for this software.

The second set of software under consideration is a more robust suite of clock control software from a project on the website Elektronika. Also released under a permissive license, many additional features are included above and beyond the core clock functionality. [14] Standout features include temperature sensor reading and temperature display support as well as a scrolling display of the date. The other important feature of this software is a semblance of documentation. While still sparse, the code is documented in a readable fashion.

The final set of clock software is from the QloneTwo team. Released under the Apache license, this control software also written for Arduino is aimed at being a perfect copy of the function of the original word clock. Nothing more than simple time support is included but an intriguing feature is strong support for button input. [15] Other than this feature, the QloneTwo library doesnt provide any additional functionality over the other two libraries. This library also lacks any form of in depth documentation.

If faced with the choice of using only off the shelf existing hardware, the library from Elektronika is the most well documented and provides the most features which line up with project requirements. Since the project focus is software design, the best course forward would be to take advantage of the useful bits of existing code under open licenses and expand it with a more flexible framework. The core concept missing from all the software libraries that were reviewed is any ease in plugging additional features or display modes into the library. To meet this goal and to facilitate various stretch goals, the best software choice is a new implementation based on the existing Elektronika implementation.

I. Testing

The world of engineering testing and development has many routes to take, but the end goal is to achieve testing that finds most bugs in the product before it reaches the user. Culminating the accumulated knowledge from the past testing experiences of people in our group, as well as reviewing testing standard online we have a fairly decided test flow. The method of our development cycle will be done via agile. We have a zen board extension for our github that allows us to create user stories and epics as tickets, and further increase our ability to efficiently create unit tests and produce a quality product. The other two large development processes are waterfall and spiral. Waterfall involves a strict step by step project breakdown with very strict requirements from start to finish. This does not match our capstone project very well, as the product itself is very open ended in design by nature. Upon first meeting with the client he provided us with a general concept of what he would like the end product to be, and in turn asked us to design the details. Without strict project requirements, waterfall style development simply will not function as intended [16]. A lot of development cycles use the spiral method, which has been found to be very effective when dealing with projects that involve the customers input for iterations of the design, as ours does. However, the reason we decided to go with agile over spiral is that spiral lends itself more to larger groups and teams, with a larger product to be developed. We, as a small team of only three dont need all of the checks and balances that spiral development would have us do, and it would truly slow us down in the long run. Agile lends itself especially well with our project because we interact with the customer often, have mandatory weekly updates that function as secondary scrums, and have quickly changing requirements and tests. All this in mind we will be proceeding with the agile style.

References

[1] MPLab, “Cloud based ide and examples.” https://www.microchip.com/mplab/mplab-xpress.

[2] Adafruit, “Adafruit trinket.” https://www.adafruit.com/products/1501.

[3] R. Pi, “Raspberry pi.” https://www.raspberrypi.org/.

[4] KeyeStudio, “Ks0166 keyestudio ds3234 high precision real time clock moduel.” http://www.keyestudio.cc/h-nd-132.html.

[5] SparkFun, “Sparkfun real time clock moduel.” https://www.sparkfun.com/products/12708.

[6] SparkFun, “Sparkfun deadon rtc breakout - ds3234.” https://www.sparkfun.com/products/10160.

[7] “Qwlock, howpublished =.”

[8] “Adafruit mini grid, howpublished =.”

[9] adafruit, “White led backlight module - small 12mm x 40mm.” https://www.adafruit.com/products/1626.

[10] adafruit, “Rgb 7-segment digit - 1” tall digit.” https://www.adafruit.com/products/1399.

[11] Adafruit, “Neopixel mini 3535 rgb led.” https://www.adafruit.com/products/2686.

[12] PiBorg, “Battborg - power your raspberry pi from aa batteries.” https://www.piborg.org/battborg.

[13] Scottbez1, “Sleek word clock.” http://www.instructables.com/id/Wordclock/.

[14] J. Sajta, “/projects/word clock.” http://www.elektronika.ba/841/word-clock/.

[15] Wouterdevinck, “Wordclock.” http://www.instructables.com/id/Wordclock/.

[16] “Testing differences, howpublished =.”