LCD Alarm Clock - B04\_G01

ECE 299

Aug 18th, 2020



**Group Information**

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Name | V Number | Sections Responsible |
| B04 | Chris Hansen | V00884651 | Requirements Specification, Circuit/PCB, Testing (Display), Formatting |
| B04 | Will Lefebure | V00873484 | Exec-Sum, Constraints, BOM, Testing (U/I) |
| B04 | Kateryna Maksymenko | V00918292 | Project Goal, Conclusion/Recom,  Testing (Ambient Sensing), References |



# Executive Summary

Our lab team’s final design utilized a liquid crystal display and a keypad to provide high programmability while simultaneously reducing hardware complexity compared to alternative design paths that were considered. The heart of our alarm clock was an Arduino Uno R3, which allowed us to supply power, program complex functionality, and integrate both digital and analog components. Our team gained experience in the areas of circuit design, project management, and printed circuit board design/manufacturing throughout this project.

Contents

[Executive Summary I](#_Toc48674389)

[List of Tables III](#_Toc48674390)

[List of Figures IV](#_Toc48674391)

[I Project Goal 1](#_Toc48674392)

[II Constraints 1](#_Toc48674393)

[III Requirement Specifications 2](#_Toc48674394)

[IV Bill of Materials 5](#_Toc48674395)

[V Circuit Schematic & PCB 8](#_Toc48674396)

[VI Testing & Validation 12](#_Toc48674397)

[Display Subsystem 12](#_Toc48674398)

[User Interface Subsystem 14](#_Toc48674399)

[Ambient Light Sensing Subsystem 17](#_Toc48674400)

[VII Conclusion & Recommendations 18](#_Toc48674401)

[References 19](#_Toc48674402)

# List of Tables

Table 1: Bill of Materials………………………………………………………………. 5

Table 2: Justifying Component Choice…………………………………………………7

# List of Figures

[Figure 1: Arduino Circuit Schematic 8](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665864)

[Figure 2: Keypad Circuit Schematic 8](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665865)

[Figure 3: Photo Transistor Circuit Schematic 9](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665866)

[Figure 4: Piezo Alarm Buzzer Circuit Schematic 9](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665867)

[Figure 5: LCD Circuit Schematic 10](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665868)

[Figure 6: Front View of 3D PCB 10](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665869)

[Figure 7: Alarm Clock Case 11](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665870)

[Figure 8: Back of 3D PCB 11](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665871)

[Figure 9: Alarm Clock PCB 12](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665872)

[Figure 10: HY1602E Block Diagram and Power Supply 13](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665873)

[Figure 11: Tinker Cad Basic LCD Test 14](https://dydr.simuldocs.com/files/ydrjkj2dtr1wy79rwpz5gczwhw2matgd/bktmnm4mr8800/bm0fqh1108800/bm0g6qn4r8800/Simul/Final_Report_HLM.docx#_Toc48665874)

# I Project Goal

Students in the 2020 ECE 299 summer session were tasked with designing and prototyping an electronic alarm clock that would display the current time and activate an alarm at a user-defined time. These alarm/clock times could be independently modified, and the brightness of the display would adjust according to the ambient light conditions. Students could also include a snooze feature for bonus marks, all though this feature was not required.

# II Constraints

Several project constraints were observed by this semesters ECE 299 class, both from the requirement specifications set out by the instructors, and the unique social -distancing circumstances created by the COVID-19 pandemic.

The first type of constraints encountered was hardware based, such as the required use of an Arduino UNO R3 as the primary micro-controller of our design. Similarly, students were also constrained to use certain software such as KiCAD (for PCB design) and TinkerCAD (for both circuit simulation and enclosure modeling). On a national level electronic equipment is subject to regulations such as RoHS (Restriction of Hazardous Substances) compliance, which influenced our team’s material selection process. Milestone deadlines posted on Coursespaces also acted as the major time constraints for our project.

# III Requirement Specifications

|  |
| --- |
| ***Design Requirements*** |
| * User to Set Time * Display Real time * Allow User to set alarm at desired time * Allow user to change alarm time * Allow user to snooze and turn off alarm * Change the brightness of alarm base of irradiance light in room * Display Real time * Speaker for alarm |
| ***Enclosure Specifications*** |
| * Opening for power cable to connect onto Arduino R3 * Minimizing physical dimensions |
| ***Electrical Specification*** |
| ***Arduino UNO R3*** |
| * Constant 5V power supply from a USB connection to the Arduino Uno R3 as specified by the data sheet for operating voltage [1] * Protection for each I/O pins from the Arduino board to keep current less then 40mA which is the absolute maximum rating as seen in the ATMega data sheet [1] * Maximum current from the VCC and GND pins are 200mA * The total amps calculated for the VCC and GND pins is 67mA as calculated from equation (7) * The maximum amps for each I/O pin are seen in the load calculation figure [1] |
| ***Piezo Alarm Buzzer*** |
| * For the Piezo Alarm Buzzer is rated 5V and 10mA as the maximum current, as specified in the data sheet [6] * For protecting the circuit between the Arduino board pin 13 and the Piezo Alarm buzzer we rated a 500-ohm resistor from the equation (1) |
| ***Photo Transistor*** |
| * The illuminance for full daylight is 10752 lux which is 1.57 mW/ [2]. For the photo transistor we needed a linear reading for irradiance from 0 to 1.57 mW/. To do this we must have a minimum amp less than 2.5mA. We use the equation (1) to calculate the resistance to be 2k ohm. * The current for irradiance of 1.57 mW/ is 2.3 mA according to a datasheet of a phototransistor [3]. * The value of voltage output can be calculated using equation (3) with I = 2.3 mA and R = 2k Ω. is 4.6 V. * The can be converted to digital domain using equation (6) which is 941. |
| ***NPN*** |
| * For the NPN transistor it has a rating of 5V [4] * For the control voltage we use the equation (5) to calculate the |
| ***Keypad*** |
| * For the keypad we used the internal pull-up resistor to protect the I/O pins |
| ***LCD*** |
| * For the main power supply for the LCD is rated for 5V and 25mA with a max of 40mA. [7] * For the power supply for the LCD backlights is rated for 5V and 160mA. [7] * For the LCD power supply for backlight we need a minimum 125ohm to protect the I/O pins found from the equation (1) with using V=5V and I=40mA. * For the main power supply for the LCD we calculate the minimum resistance of 200ohms from equation (1) with using V=5V and I=40mA. * For the main power supply and LCD backlight resistor, we used to be 220ohm resistor which has a rating of 3W to account for the resistor accuracy. |

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |
|  | (3) |
|  | (4) |
|  | (5) |
|  | (6) |
|  | (7) |

where:

V is the voltage of the load in volts

I is the current of the load in amps

P is the power of the load in watts

is the value from Analog read function

is the value from Analog write function

R is the resistance of the load in Ω

n is total number of devices connected to the Arduino Uno R3

is the total current at GNDS

when k=1, 2, …, n is the total current for each device connected to Arduino Uno R3

# IV Bill of Materials

Table 1: Bill of Materials

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Label in schematic | Component Description | Part number | Cost/unit quantity | Source of cost information |
| ARD1 | Arduino Uno R3 | 763004-  200050 | CA$22.95 | <https://www.sparkfun.com/products/11021> |
| KeyPad | 4x4 Keypad | 3844 | CA$9.36 | <https://www.digikey.ca/en/products/detail/adafruit-industries-llc/3844/9561536> |
| DS1 | LCD Display | HY1602E | CA$5.33 | <https://www.yoycart.com/Product/530342109609/> |
| Q1 | 800nm Phototransistor | PT480E00000F | CA$0.75 | <https://canada.newark.com/sharp/pt480e00000f/phototransistor-side-view/dp/97K7891> |
| Q2 | BJT Transistor | 2N3903 PBFREE | CA$0.86 | <https://www.digikey.ca/en/products/detail/central-semiconductor-corp/2N3903-PBFREE/4806875> |
| R3/R4 | 220 Ω [ +/- 1% , 3W ] | 43F220E | CA$2.44 | <https://www.digikey.ca/en/products/detail/ohmite/43F220E/823229> |
| R2 | 2k Ω [ +/- 1% , 0.4W ] | MBB0207VD2001BC100 | CA$1.03 | <https://www.digikey.ca/en/products/detail/vishay-bc-components/MBB0207VD2001BC100/7351940> |
| R1 | 500 Ω [ +/- 5% , 1W ] | AC01000005000JA100 | CA$0.97 | <https://www.digikey.ca/en/products/detail/vishay-bc-components/AC01000005000JA100/596588> |
| J1 | 8 Wire Terminal | OSTTC080162 | CA$2.76 | <https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTTC080162/614555?s=N4IgTCBcDaIKIBEwDYAMyC0A5BIC6AvkA> |
| LS1 | Piezo [ 5v, 10mA ] | TP124005-2 | CA$1.34 | <https://www.digikey.ca/en/products/detail/db-unlimited/TP124005-2/10288171> |

Table 2: Justifying Component Choice

|  |  |
| --- | --- |
| Label in schematic | Functional reason for selecting this component |
| ARD1 | Constrained to Requirement Specifications. |
| KeyPad | High programmability for low pin usage. |
| DS1 | Affordable and compact design. |
| Q1 | Proper wavelength and affordable. |
| Q2 | High Breakdown and frequency. |
| J1 | Allows flexible placement of Keypad in enclosure. |
| LS1 | Affordable, loud volume (75dB). |
| R1 | Meets power requirements, low +/- range, and affordable. |
| R2 | Meets power requirements, low +/- range, and affordable. |
| R3/R4 | Meets power requirements, low +/- range, and affordable. |

# V Circuit Schematic & PCB

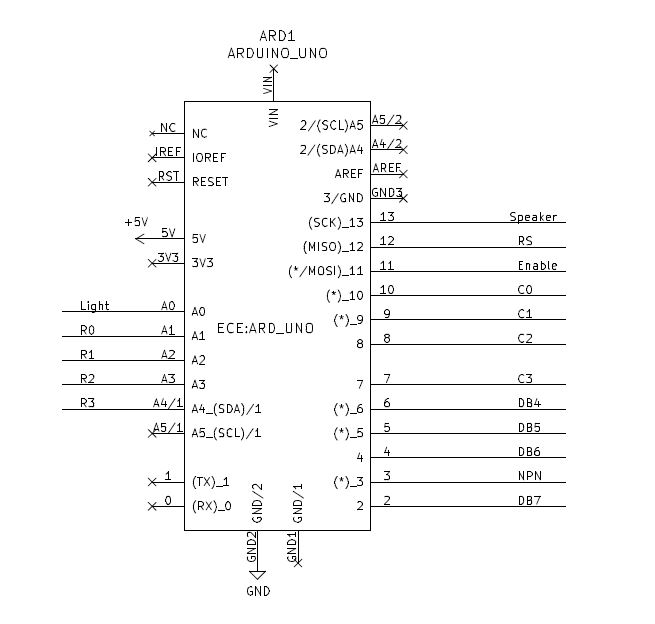
 In the constraints, the Arduino UNO R3 is the main source of control for the alarm clock. We used the Arduino to monitor and control five essential components of the alarm clock. As we see in the Figure 1, the Arduino board uses I/O pins A1-A4 and 7-10 to connect to the keypad.

Figure : Arduino Circuit Schematic

For our PCB we decided to use a board terminal block to have our keypad connected as seen in the Figure 2. The reason for this was to minimize the PCB size and allow us to have the keypad mounted on any location inside the enclosure without the limitation of being directly attached to the PCB.

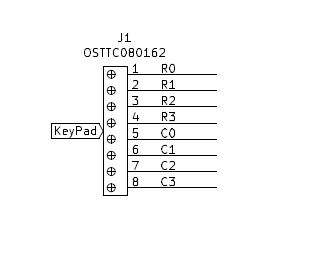


Figure : Keypad Circuit Schematic

The next component for our alarm clock was the photo transistor. As seen in figure 3 on the next page, it is connected to a 2kΩ resistor to produce a linear change in voltage proportional to the level of ambient light detected in the environment.

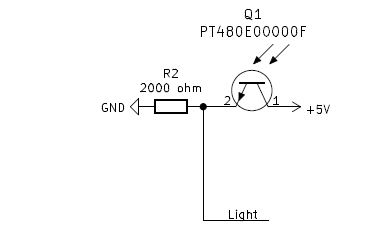


Figure : Photo Transistor Circuit Schematic

The I/O pin A0 was used to read analog signals and connect it to the photo transistor as seen in the photo transistor diagram. A simple circuit was made for the speaker with a 500Ω resistor to control the current and connect it to a speaker and ground as seen in Figure 4.

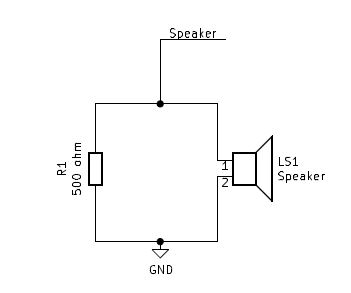


Figure : Piezo Alarm Buzzer Circuit Schematic

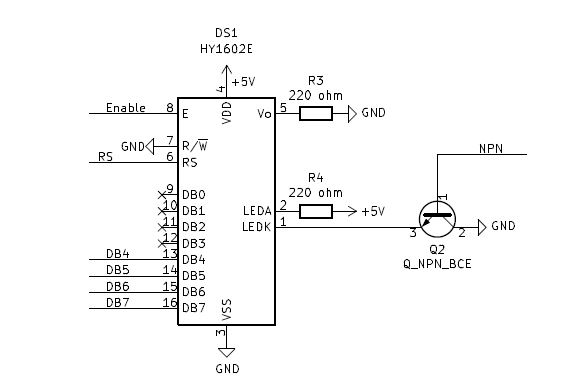
 The I/O pin 13 was used on the Arduino Uno R3 to send a digital signal to control the sound produce by the speaker. Next, we decided to use an LCD screen instead of a seven segment display to allow us to display alarm at the same time as the main time. This can be seen in Figure 5.

Figure : LCD Circuit Schematic

With this design we had 4 connections to the Arduino board using I/O pin 2,4,5,6 to control what is displayed on the LCD. We also connected enable and the RS connection from the LCD to the Arduino board. Finally, we added a 220Ω resistor to V0 to protect the LCD and Arduino board. With the back light of the display we decided to add a 220Ω resistor to have the LCD and Arduino protected with the NPN transistor off. When you start sending a signal from the Arduino to the NPN transistor it will change the current entering the LCD to change the background brightness.

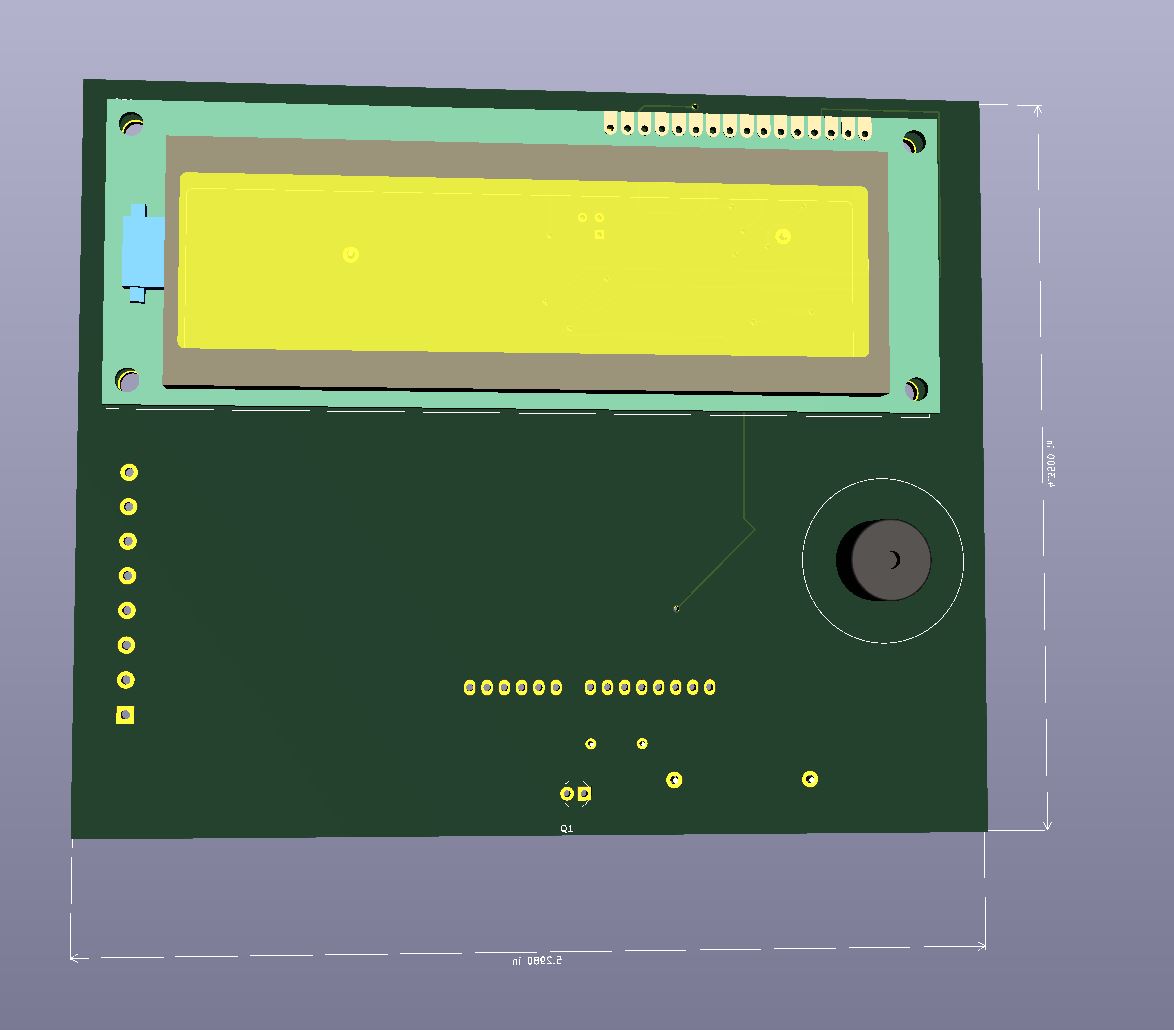
With the implantation of the circuit into a PCB we decided to have the LCD, photo transistor, and speaker on the front as seen in the Figure 6.

Figure : Front View of 3D PCB

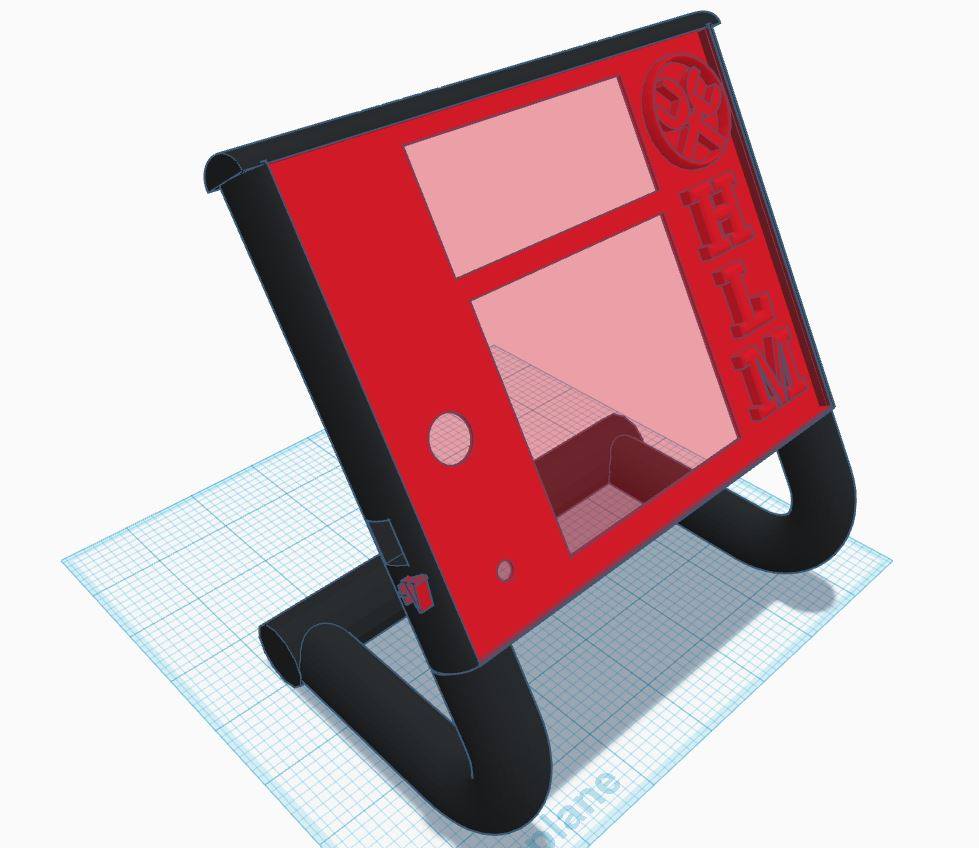
 This was done to minimize the PCB size and allow the LCD, photo transistor and speaker to fit through holes of the case and causing the alarm clock to look smooth and simple on the front as seen by the Figure 7.

Figure : Alarm Clock Case

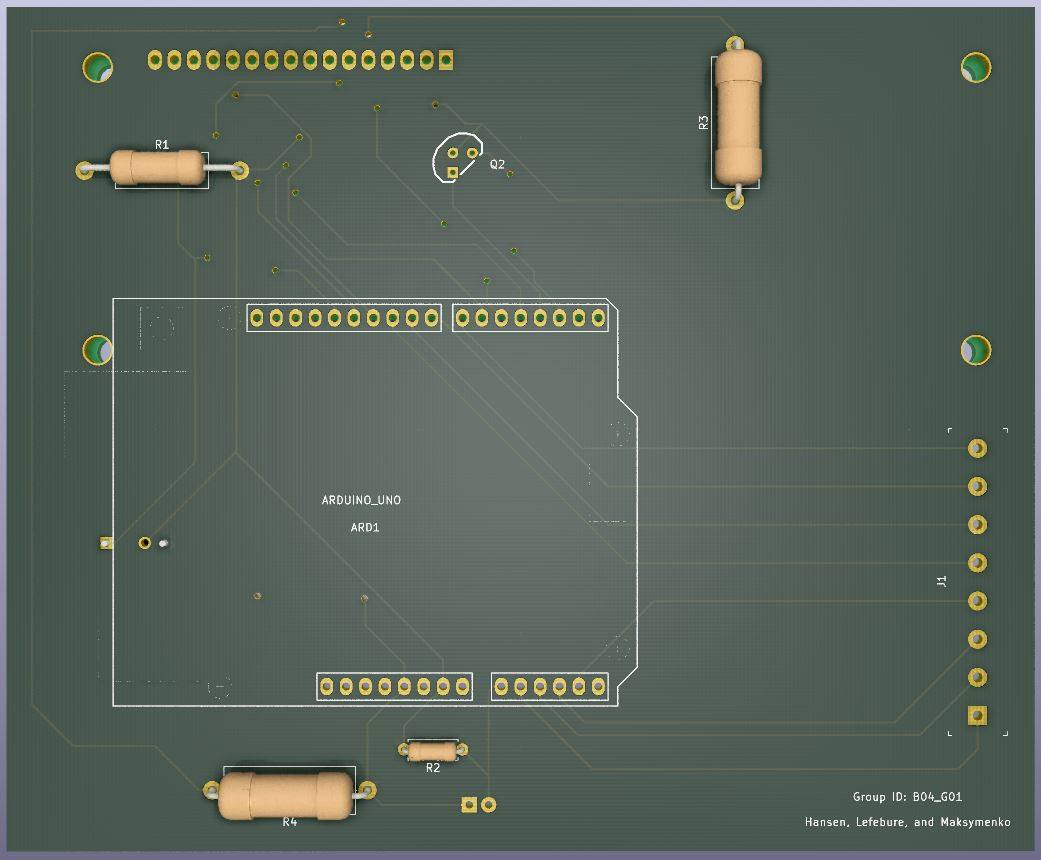
 For the remaining components it was decided to have it on the back side of the   
PCB as seen in the Figure 8.

Figure : Back of 3D PCB

As we see on the PCB Figure 8 and Figure 9, we used a board terminal block to connect the keypad to the PCB to minimize the PCB size and allow easy placement of the keypad on the case.

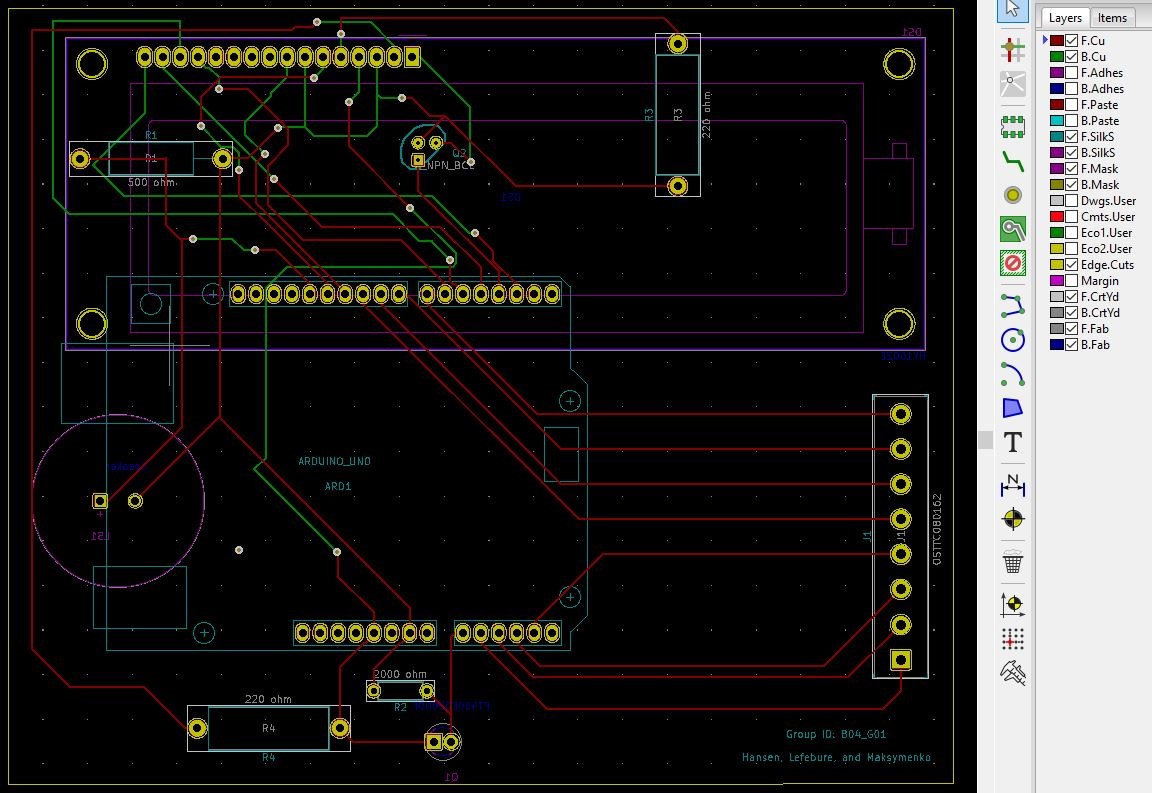


Figure : Alarm Clock PCB

# VI Testing & Validation

## Display Subsystem

The HY1602E LCD comes in a 16 pin package. It has the following pin configuration:

|  |  |  |
| --- | --- | --- |
| Pin No | Symbol | Description |
| 1 | LEDK | Power supply for Backlight (-) |
| 2 | LEDA | Power supply for Backlight (+) |
| 3 | VSS | Ground for logic |
| 4 | VDD | Power supply for logic |
| 5 | Vo | Power supply for LCD drive |
| 6 | RS | Register Selection |
| 7 | R/W | Read/Write selection |
| 8 | E | Enable signal for LCM |
| 9-16 | DB0-DB7 | Data Bus lines |

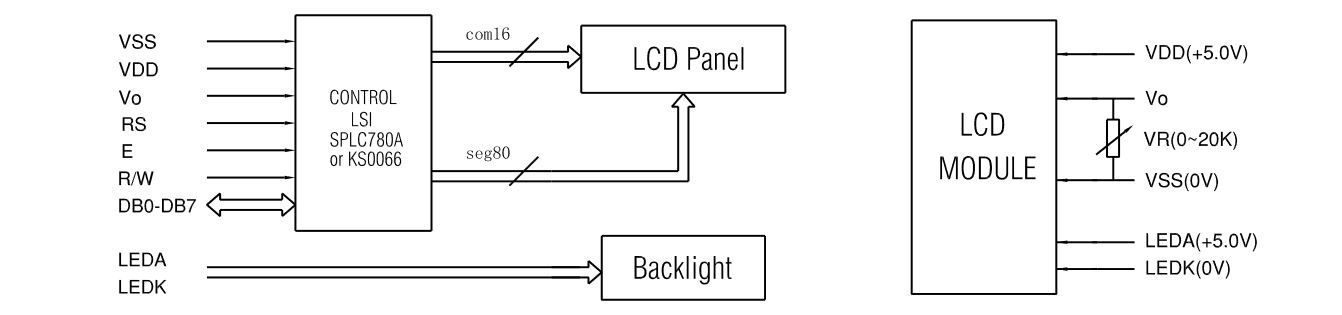
The Hy1602E LCD has the following block Diagram and Power Supply:

Figure : HY1602E Block Diagram and Power Supply

The HY1602E LCD has the following Electrical Characteristics:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Item | Symbol | Test Condition | Min | Typ. | Max. | Unit |
| Operating Voltage | VDD | Ta=25C | --- | 5.0 | --- | V |
| Operating Voltage for LCD | Vlcd | Ta=25C | --- | 5 | --- | V |
| Supply Current | Ldd | Ta=25C, VDD=5V | --- | 2 | 3 | mA |
| Supply Current for Backlight | If | Ta=25C, Vf=4.2V | --- | 160 | --- | mA |

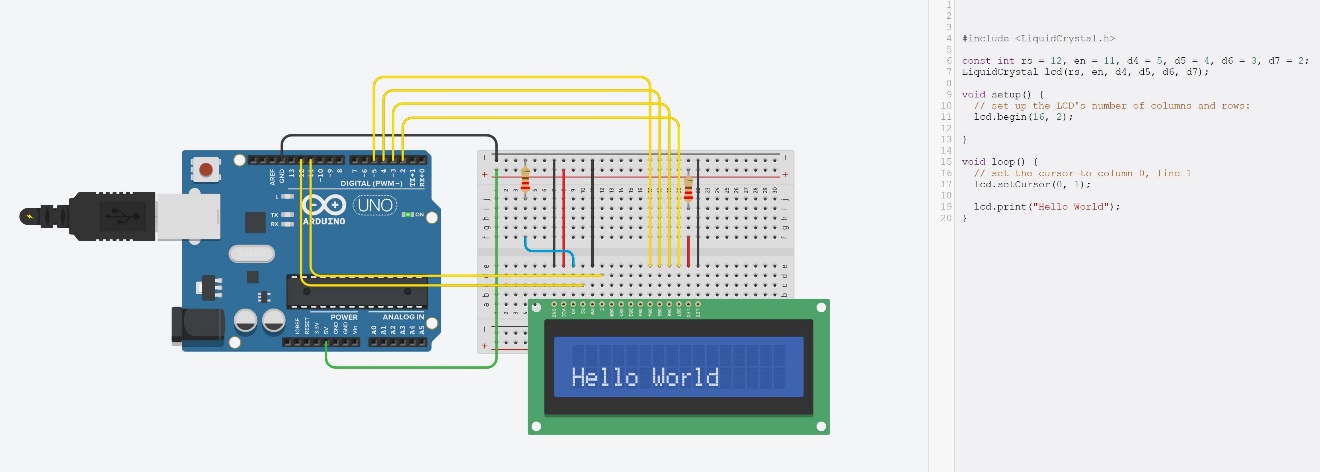
As in accordance to the pin configuration the R/W is active high, so for enabling the LCD to work, this pin is connected to the ground. The VDD is the supply voltage for driving the logic circuit within the LCD and should be 5V as seen in the electrical characteristics []. The VSS will be connected to ground. The LEDA is the main power supply for the back lights and is set at 5V and have has a typical current at 160mA. With the constraint of are Arduino board we set the current less then 40mA and use an NPN transistor to control the current to increase and decrease the brightness. For pin DB0-DB3 will not be used for the Data Bus line. The Vo pin will be connected to a 220-ohm resistor and the ground to keep the current limited for the Arduino board. To test the HY1602E LCD we will connect the VDD to a 5V power supply and have R/W, and VSS connected to ground to have them at data register and read state. Next, we will connect the LEDA with a 220-ohm resistor to a 5V power supply and LEDK to GND. With this set up we will see the LCD turned on with the back lights on. Connecting the DB4-DB7, RS and Enable to the Arduino board to allow us to test the print feature. On the Arduino board we enter the code seen in figure 7 to confirm the LCD prints the following message “Hello World”.

Figure : Tinker Cad Basic LCD Test

## User Interface Subsystem

In order to test the user interface, a comprehensive test procedure was developed. This test verifies the functionalities of time setting and alarm setting in both 12hr and 24hr formats, as well as the implemented optional snooze feature. The clock code that runs the U/I is available on GitHub for viewing and testing purposes [5]. The circuit can be simulated in TinkerCAD, which served as the main environment in which the testing was conducted.

|  |
| --- |
| **U/I TEST**  **12hr Time**  **Set Time 12hr:**   1. Press ‘B’ to enter the “Time Set” menu. *( Button Press works? [ Yes / No ] )* 2. Enter at least 4 numbers to select a time. *( Button presses registered? [ Yes / No ] )* 3. Press ‘C’ to toggle AM/PM. *( AM/PM Toggles? [ Yes / No ] )* 4. Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   **12 Hr Rollover:**   1. Press ‘B’ and enter the time 12:55 PM.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the clock transitions from 12:59 PM to 01:00PM.   *( Successful 01:00 PM Rollback? [ Yes / No ] )*   1. Press ‘B’ and enter the time 12:55 AM.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the clock transitions from 12:59 AM to 01:00AM.   *( Successful 01:00 AM Rollback? [ Yes / No ] )*  **AM/PM**  **AM/PM Rollover:**   1. Press ‘B’ and enter the time 11:55 AM.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the AM/PM symbol toggles as the clock transitions from 11:59AM to 12:00PM. *( AM/PM toggles at Noon? [ Yes / No ] )* 2. Press ‘B’ and enter the time 11:55 PM.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the AM/PM symbol toggles as the clock transitions from 11:59PM to 12:00AM. *( AM/PM toggles at Noon? [ Yes / No ] )*   **24/12**  **Toggle 24/12hr:**   1. Press ‘C’ to toggle the clock into 24hr mode.   *(AM/PM indicator disappeared? [ Yes / No ] )*   1. Press ‘B’ and enter the time 20:00.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Press ‘C’ to toggle the clock into 12hr mode. NOTE: If the button is pressed at [20:**XY**] where **X** & **Y** signifies the second’s digits at the time of pressing, the clock should then read [08:**XY** PM] after switching to 12hr mode. *( Clock displays 08:****XY****? [ Yes / No ] )*   **24hr Time**  **Set Time 24hr:**   1. Press ‘C’ to toggle the clock into 24hr mode.   *(AM/PM indicator disappeared? [ Yes / No ] )*   1. Enter at least 4 numbers to select a time. *( Button presses registered? [ Yes / No ] )* 2. Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   **Midnight Rollover:**   1. Press ‘B’ and enter the time 12:55 PM.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the clock transitions from 23:59 to 00:00.   *( Successful 00:00 Rollback? [ Yes / No ] )*    **12hr Alarm**  **Set Alarm 12hr:**   1. Press ‘A’ to enter the “Alarm Set” menu. *( Button Press works? [ Yes / No ] )* 2. Enter at least 4 numbers to select a time. *( Button presses registered? [ Yes / No ] )* 3. Press ‘C’ to toggle AM/PM. *( AM/PM Toggles? [ Yes / No ] )* 4. Press ‘A’ to confirm the alarm time. *( Alarm correctly set? [ Yes / No ] )* 5. Press ‘#’ to prime and display the alarm. *( Alarm icon displaying? [ Yes / No ] )*   **Alarm:**   1. Press ‘A’ and enter an alarm for 11:59 PM.   Press ‘A’ to confirm the alarm time. *( Alarm correctly set? [ Yes / No ] )*   1. Press ‘#’ to prime and display the alarm. *( Alarm icon displaying? [ Yes / No ] )* 2. Press ‘B’ and enter the time 11:50 PM.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the alarm sounds when the clock strikes 11:59 PM. *( Alarm sounds at set time? [ Yes / No ] )* 2. Press ‘#’ to disarm the sounding alarm. *( Alarm silences and the icon disappears? [ Yes / No ] )* 3. Press ‘#’ to prime and display the alarm. *( Alarm icon displaying? [ Yes / No ] )* 4. Press ‘B’ and enter the time 11:50 AM.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the alarm does not sound when the clock strikes 11:59 AM. *( Alarm recognizes AM/PM and doesn’t sound? [ Yes / No ] )*   **Alarm Snoozed:**   1. While an alarm is sounding, press ‘ \* ’ to snooze it for a pre allotted number of seconds. *( Alarm successfully snoozed? [ Yes / No ] )*   **24hr Alarm**  **Set Alarm 24hr:**   1. Press ‘C’ to toggle 24/12hr format and enter 24hr mode. *( 24/12 Toggles and AM/PM icon disappears? [ Yes / No ] )* 2. Press ‘A’ to enter the “Alarm Set” menu. *( Button Press works? [ Yes / No ] )* 3. Enter at least 4 numbers to select a time. *( Button presses registered? [ Yes / No ] )* 4. Press ‘A’ to confirm the alarm time. *( Alarm correctly set? [ Yes / No ] )* 5. Press ‘#’ to prime and display the alarm. *( Alarm icon displaying? [ Yes / No ] )*   **Alarm Triggered:**   1. Press ‘A’ and enter an alarm for 19:45.   Press ‘A’ to confirm the alarm time. *( Alarm correctly set? [ Yes / No ] )*   1. Press ‘#’ to prime and display the alarm. *( Alarm icon displaying? [ Yes / No ] )* 2. Press ‘B’ and enter the time 19:30.   Press ‘B’ to confirm the Set Time. *( Time correctly set? [ Yes / No ] )*   1. Observe and confirm the alarm sounds when the clock strikes 19:30. *( Alarm sounds at set time? [ Yes / No ] )* 2. Press ‘#’ to disarm the sounding alarm. *( Alarm silences and the icon disappears? [ Yes / No ] )*   **Alarm Snoozed:**   1. While an alarm is sounding, press ‘ \* ’ to snooze it for a preset number of seconds. *( Alarm successfully snoozed? [ Yes / No ] )* |

## Ambient Light Sensing Subsystem

For testing the ambient light sensing subsystem, the current that goes though the LCD must be known. This current then can be converted to irradiance and compared to the irradiance of the daylight conditions.

The maximum current for the LCD with backlights on is 160 mA but it was limited using a current limiting resistor of 220 Ω with voltage input 5 V

Since an NPN transistor is used, the current flowing through the LCD display is 10 mA [4]. If the current is 10 mA, looking at the datasheet the irradiance is 6.2 mW/ [3]. The irradiance used for this project is below 6.2 mW/ under daylight conditions, so the brightness will be changing with ambient light.

# VII Conclusion & Recommendations

Final prototyping of this project achieved all mandatory production and functional milestones, on deadline. A secondary objective was also met such as the incorporation of a “snooze” button.

With a three-member design team, it was decided to divide work equally with regards to coding, hardware and testing. As each team member had differing areas of expertise, the workload was distributed with those proficiencies in mind. The division of responsibilities enabled focused work on our respective goals, culminating towards completing the project as a whole.

The project was made during a global pandemic which not allowed us to make a physical prototype of the alarm. While developing the prototype in online environment, the biggest challenge was working in Tinkercad because of its timing delay. When the code for setting the time, setting the alarm and brightness was combined, the timing in the simulation became slow.

Given another opportunity to do this project, more time could be spent modifying or redesigning the enclosure. If the alarm clock would be prototyped, there would be not enough space for putting all the components inside. The dimensions considered were only PCB, not including soldered components and all the wires.

# References

[1] Atmel. “ATMega 328p Datasheet,” mouser.com. [Online]. Available: https://www.mouser.com/pdfdocs/gravitech\_atmega328\_datasheet.pdf. [Accessed: August 17, 2020].

[2] National Optical Astronomy Observatory. “Recommended Light Levels (Illuminance) for Outdoor and Indoor Venues,” noao.edu. [Online]. Available: https://www.noao.edu/education/QLTkit/ACTIVITY\_Documents/Safety/LightLevels\_outdoor+indoor.pdf. [Accessed: August 17, 2020].

[3] Sharp. “PT480E00000F Datasheet,” canada.newark.com. [Online]. Available: https://canada.newark.com/sharp/pt480e00000f/phototransistor-side-view/dp/97K7891. [Accessed: August 17, 2020].

[4] Central Semiconductor Corp. “2N3903 Datasheet,” digikey.ca. [Online]. Available: <https://www.digikey.ca/en/products/detail/central-semiconductor-corp/2N3903-PBFREE/4806875>. [Accessed: August 17, 2020].

[5] Lefebure, W. 2020. “Clock\_Code\_Final.INO” [Online]. Available: <https://gist.github.com/frenchylastname/b65accbbd28c263db61967c7cb53f0bf>. [Accessed: August 17, 2020].

[6] DB Unlimited. “TP124005-2 Datasheet,” digikey.ca. [Online]. Available: https://www.digikey.ca/en/products/detail/db-unlimited/TP124005-2/10288171. [Accessed: August 17, 2020].

[7] Microchip. “HY1602E Datasheet,” yoycart.com. [Online]. Available: https://www.yoycart.com/Product/530342109609/. [Accessed: August 17, 2020].

# Appendix A – Enclosure Details

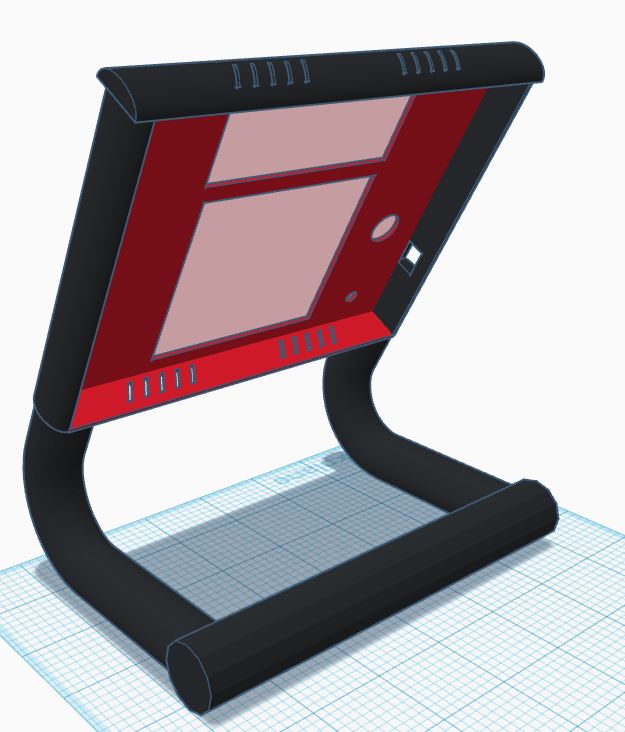


Figure 13: Back of Clock Enclosure

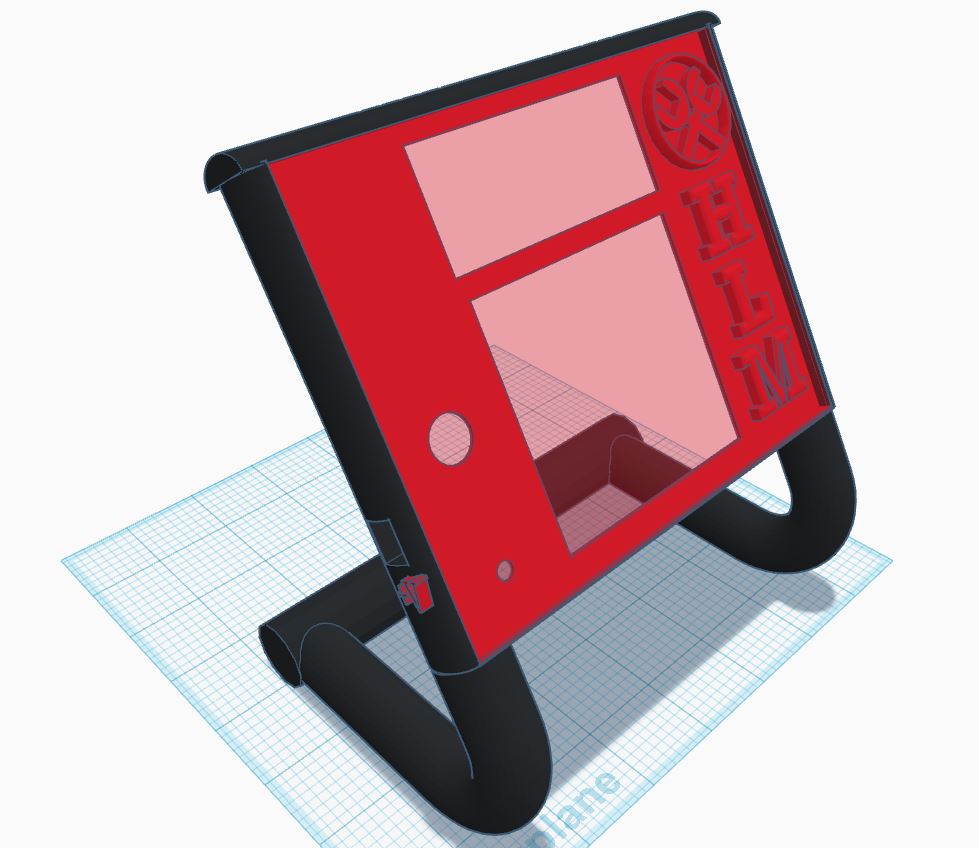


Figure : Front of Clock Enclosure

# Appendix B – PCB

Figure 14: Front/Back PCB

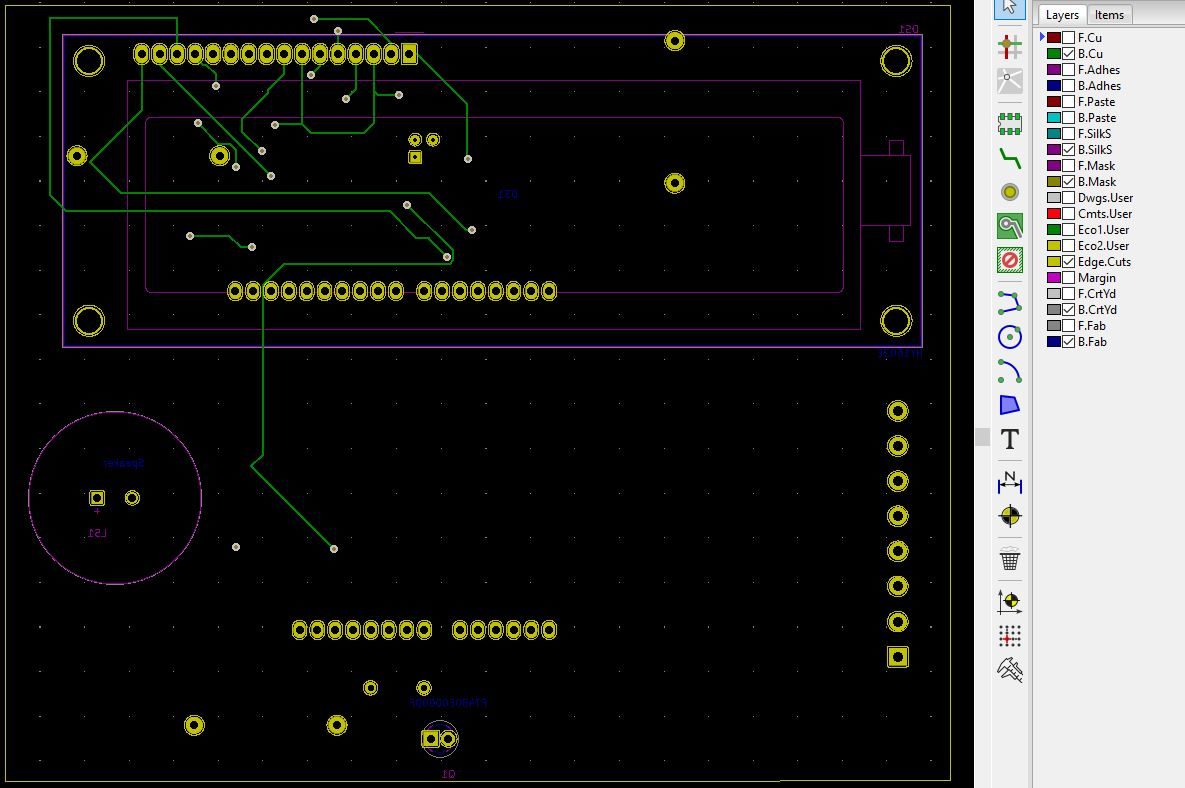
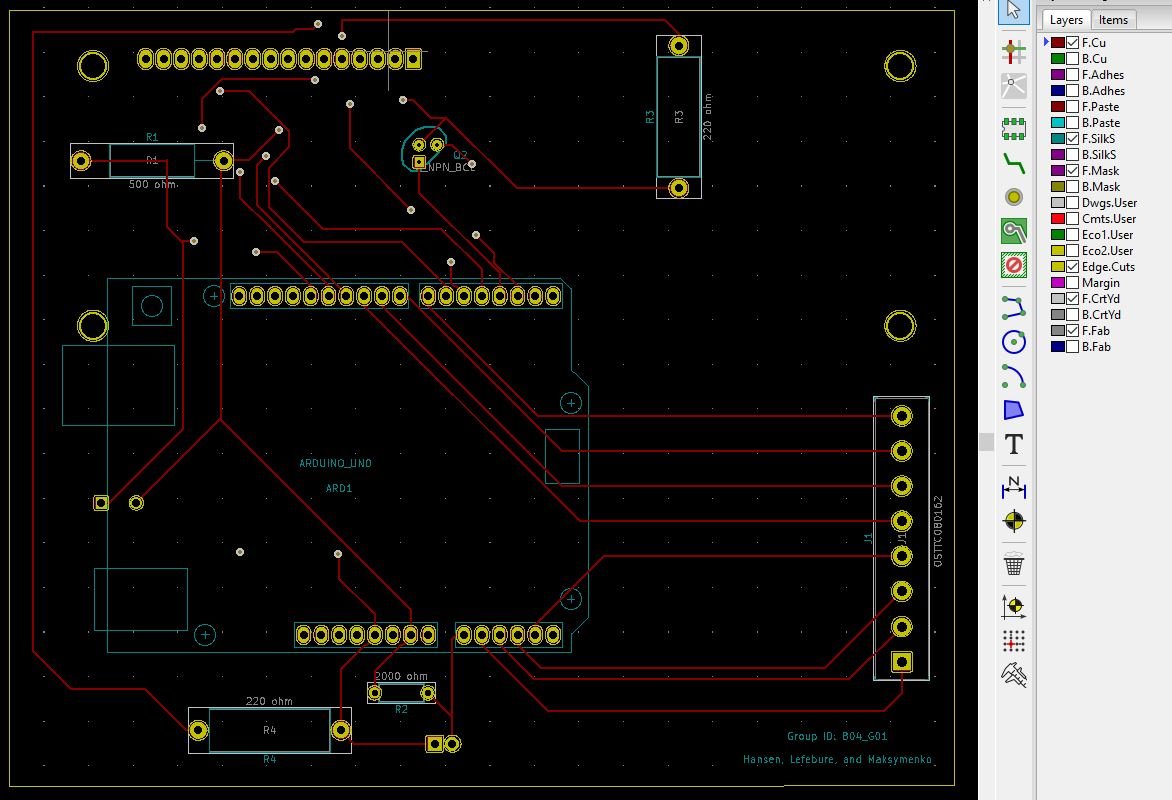


Figure 15: Back PCB

Figure 15: Front PCB