Converting resistor networks to a breadboard and non-breadboard model to find the best modelling option for circuit integrity and thermal management. (ARCHIVED IB RESEARCH PAPER)

Elian Rieza

https://elianrieza.dev

Contents

1 Introduction and Aim

As an aspiring computer engineer and computer scientist, I've been modifying and manipulating circuits and microcontrollers in my free time. Recently, I've been printing out my own printable circuit board (PCB) based on microcontrollers for hobbyist projects and a crucial part of modelling to-be-printed boards was choosing what method allows better thermal dissipation. This is because as my model runs for longer, the temperature change will be greater and at a point, the running temperature of the prototype will be high enough to be dangerous and could cause third-degree burns or possibly cause combustion in components.

This IA will use "thermal dissipation" and "change in temperature" interchangeably to refer to the measure of heat energy dissipated in the circuit models.

On the other hand, as these microcontrollers act as a small computer which execute programmed tasks and carry information through currents (Eland, 2024), the information being carried through currents should be limited and ideally shouldn't short circuit and interfere. To ensure this, resistor networks are used which, for the sake of the microcontroller, permit circuit integrity by controlling minimal amounts of current flow and eliminating short circuit and ensure data transfer is "sane" as binary information is sent through currents (Das & Yang, 2017). However, if I want to modify this, I'd need to model these resistor networks separately. Resistor networks (also known as resistor arrays) generate a great amount of heat as time passes thus my reasoning in choosing this question as there's a trade-off in circuit integrity and thermal dissipation. There are two key ways these resistor networks can be modeled.

The two type of models I primarily use are breadboard models, a breadboard being a plastic case with metal strips beneath which allows extremely small currents and relatively small voltages to pass through and conduct through other connected components, namely resistors. On the other hand, another way to model PCBs before printing is with alligator clips and their wires. This model takes more space but is more cost-effective than buying a breadboard. Either way, both conduct electric currents similarly and both are insulated by polyvinyl chloride (BFP, 2019) but conduct differently with breadboards using copper and alligator clips usually varying. However, these pale in comparison to PCBs which are usually printed using 5.5Oz copper conductive traces (3M, 1994) and is insulated with a solder mask, epoxy and silicon layer to ensure heat dissipation to be as even and equal as possible. These different material makeups all allow current to pass through but all deal with the heat generated from the current differently but I will only be calculating the heat dissipated in conductive material for this IA.

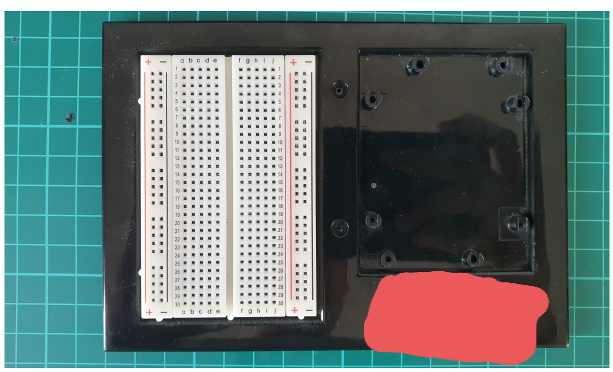


Figure 1: A breadboard

The main aim of the IA is to find the most optimized way to model a prototype for electronic components that I use in terms of thermal dissipation and circuit integrity. I will analyze the ESP-32, ESP-8266 and the Arduino Uno's resistor networks, determine the total resistance for each loop present then calculate their total temperature dissipated every minute - this represents their intended dissipation in a printed PCB model. I'll then proceed to model these circuits with a breadboard and without a breadboard respectively, using a multimeter to calculate the current flow with the same circuit I modeled the ideal thermal dissipation with. I'll analyze the resulting values then evaluate and conclude my investigation.

2 Context and Methodology

Initially, I needed to find the schematics of each microcontroller I needed to use to determine the resistor networks present in the microcontrollers. As all 4 components' schematics were online, calculating and modelling the printed circuits were easy (see Appendix C). Additionally, I measured the length of conductive tracing, also known as PCB wire, based on the schematic diagrams which provided a 1:1 ratio.

Microcontroller	Total conductive trace length on an ideal PCB (cm)
ESP-8266	9.18
ESP-32	31.6
Arduino Uno	49.05

Table 1: Intended length of resistor network conductive trace in the microcontrollers used

After finding the resistor networks for each board, the next step is attempting to find the total energy dissipated from each loop based on a derived form of Ohm's Law or, $V = IR$.

Definition 2.1 (V)**.** Let V be the voltage in a loop, measured in volts.

Definition 2.2 (I)**.** Let I be the current in a loop, measured in amps (amperes).

Definition 2.3 (R)**.** Let R be the total resistance in a loop, measured in ohms (Ω).

Definition 2.4 (P)**.** Let P be the power dissipated in a loop, measured in watts.

Definition 2.5 (H)**.** Let H be the heat energy dissipated in a loop, measured in joules.

Definition 2.6 (t)**.** Let t be the time in seconds, equivalent to 60 seconds.

$$\textbf{Equation 1} \implies V = IR \quad \text{(Ohm's Law)}$$

$$I = \frac{V}{R} \quad \text{(Rearranged for current)}$$

$$P = VI \quad \text{(Power in a circuit formula)}$$

$$P = V \cdot \frac{V}{R} \quad \text{(Substitute } I \text{ from Ohm's Law)}$$

$$\implies P = \frac{V^2}{R} \quad \text{(Simplify)}$$

$$H = Pt \quad \text{(Energy in a circuit formula)}$$

$$H = \frac{V^2}{R}t \quad \text{(Substitute } P \text{ from Power formula)}$$

$$H = \frac{(IR)^2}{R}t \quad \text{(Substitute } V \text{ from Ohm's Law)}$$

$$\textbf{Equation 2 (DeCross et. al., 2018)} \implies H = I^2Rt \quad \text{(Total heat energy dissipated (in Joules).)}$$

The resulting value for the H formula represents the total energy dissipated as heat energy at a constant time $t = 60$ seconds. However, the resulting value H represents the heat energy transferred and not the change in temperature, thus, the equation must find the change in temperature as I am attempting to find the amount of heat transferred between materials. This is because I'll first be calculating the ideal temperature change on a PCB, whose wires are made of 5.5-oz copper wires, then on a breadboard, whose wires are made of phosphor bronze and connected with wires made of 22-AWG (American Wire Gauge) copper wire, then on alligator clips, whose model I used also used a 22-AWG copper wire.

Definition 2.7 (m)**.** Let m be the mass of the conductive trace per meter without the insulation, measured in kilograms.

Definition 2.8 (c)**.** Let c be the specific heat capacity of the material of the conductive trace, measured in Joules per kilogram per Celsius ($J/kg/°C$).

Definition 2.9 (ΔT)**.** Let ΔT be the change in temperature in a loop, measured in degrees Celsius ($°C$).

$$H = mc\Delta T \quad \text{(Equation to find heat transferred)}$$

$$\textbf{Equation 3} \implies \Delta T = H \div mc \quad \text{(Total heat energy dissipated in a circuit)}$$

2.1 Circuit matricization

My investigation requires finding the thermal dissipation in a resistor network where each loop in the circuit has a separate amount of resistance and voltage but follows Kirchhoff's circuit laws otherwise. Using the ESP-8266 as an example, I first created an adjacency matrix for the ideal ESP-8266 circuit where each junction - or a point where three or more wires meet - and each battery acts as a vertex, a simple, weighted, directed graph can be found with each branch's weight representing the resistance flowing through the branch and the direction of the graph representing the current flow respective to the battery terminals.

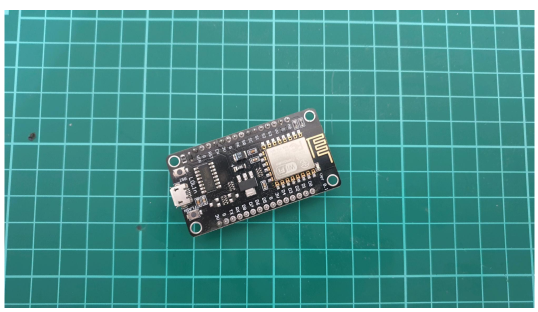
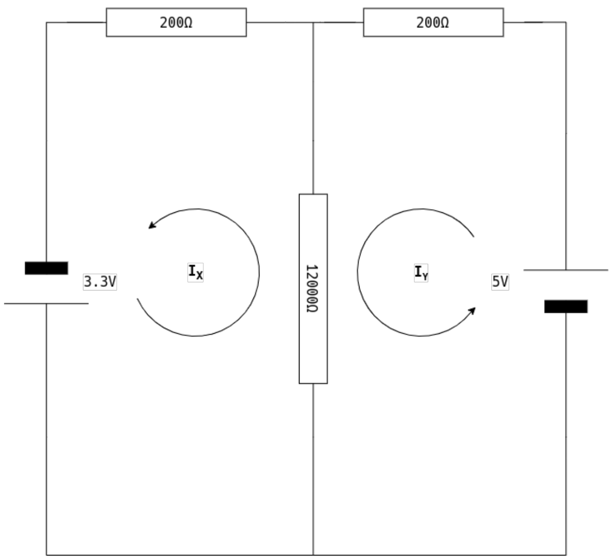


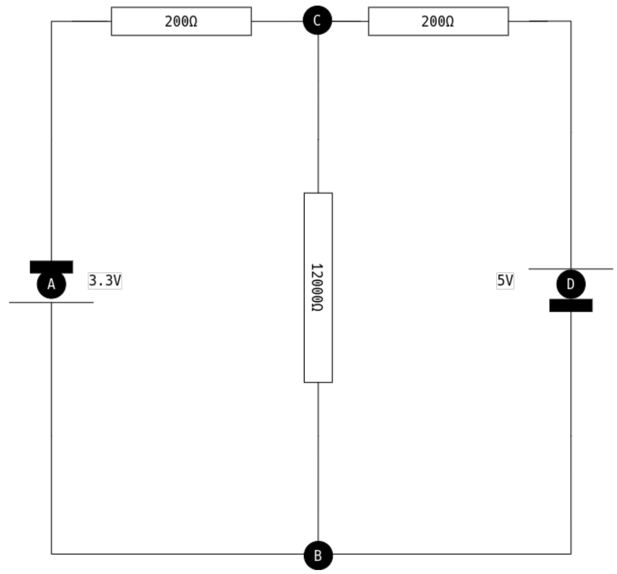
Figure 2: The ESP-8266

Definition 2.10 (A)**.** Let A be the weighted, directed adjacency matrix of the ESP-8266 resistor network.

The 4x4 adjacency matrix A for the graph above represents the mapped resistance between vertices. Note that there are branches that are connected but with no resistors and thus cannot be assigned a value. These can be represented with the special value -1 on the adjacency matrix to represent an electrical circuit connection with 0 resistance so it will need to be substituted as 0 when being calculated a weight in an algorithm because while a path is present between nodes with a -1 value, it holds no weight.



(a) Illustration of the resistor network in the ESP-8266



(b) Vertices in the ESP-8266

Figure 3: Simplified representation of the ESP-8266

2.2 Verifying circuit integrity

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 12000 & -1 \\ 200 & 12000 & 0 & 0 \\ 0 & 0 & 200 & 0 \end{bmatrix}$$

Definition 2.11 (A')**.** Let A' be the unweighted, directed adjacency matrix of the ESP-8266 resistor network where all non-zero values are substituted as 1.

Definition 2.12 (n)**.** Let n be the number of loops in any given resistor network model.

Definition 2.13 ($\sum_{i=1}^{n+1} A'^i$)**.** Let $\sum_{i=1}^{n+1} A'^i$ be the maximum-possible-routes matrix for the ESP-8266, where $\sum_{i=1}^{n+1} A' = A'^1 + A'^2 \dots A'^{n+1}$.

To confirm the construction of a circuit without any short circuits - also known as the circuit integrity - the maximum-possible-routes of the unweighted directed matrix $n+1$ edges from the battery vertex back to itself should equal 1, where n is the amount of loops present in the circuit. This is because A^{n+1} represents that each node only has exactly 1 way to get to itself. A' is constructed where the non-zero values (including the special value -1) of matrix A is substituted with the value 1 to represent an electrical circuit connection between the two nodes.

$$A' = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

In this case, the ESP-8266 resistor network has 2 loops meaning that to confirm the integrity, the minimum-possible-walks matrix should be summed up to A^3 where $A' + A'^2 + A'^3 = \sum_{i=1}^{3} A'$.

$$\sum_{i=1}^{3} A' = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & 1 & 2 & 1 \\ 2 & 3 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 2 & 1 & \mathbf{1} \end{bmatrix}$$

As demonstrated, the route to and from vertex A and to and from vertex D only have 1 possible route in $\sum_{i=1}^{3} A'$. Any diagonal value from a battery vertex greater than 1 suggests that there are different currents arriving at the wrong battery terminal, causing a short circuit, invalidating the circuit's integrity.

2.3 Ohm's Law matricization

Matrix A cannot be used by itself to calculate the total resistance of a resistor network. If we instead treated the battery sources solely as vertices to represent the voltage and used the k-Nearest Neighbour (k-NN) algorithm to calculate the resistance per loop - where the beginning vertex and the final vertex of the pathfinding algorithm is the same vertex and is. In the case of the ESP-8266, we would receive two resistance values pertaining to the resistance in loops I_X, represented by the voltage flow from A and I_Y, represented by the voltage flow from D. In other words, in order to only take the total resistance of the loops, a cyclic but not strongly connected cycle must be found with T_A where each cycle found using k-NN must exclusively only start at A or D. In this case, the loops form a partially complete, cyclical and connected graph which can be built outside the whole resistor network. This is called a partite and this is referred to a n-partite graph.

Definition 2.14 (ΣX_R)**.** Let ΣX_R be the total resistance of loop X

Definition 2.15 (ΣY_R)**.** Let ΣY_R be the total resistance of loop Y

k-NN Algorithm starting at Vertex A

AB = -1
BC = 12000
CA = 200
$\Sigma X_R \leftarrow$ 12200
$\because$ **0 + 12000 + 200 = 12200**

Note that as `AB` $= -1$, the value of `AB` should be considered as 0 when calculating ΣR_R.

k-NN Algorithm starting at Vertex D

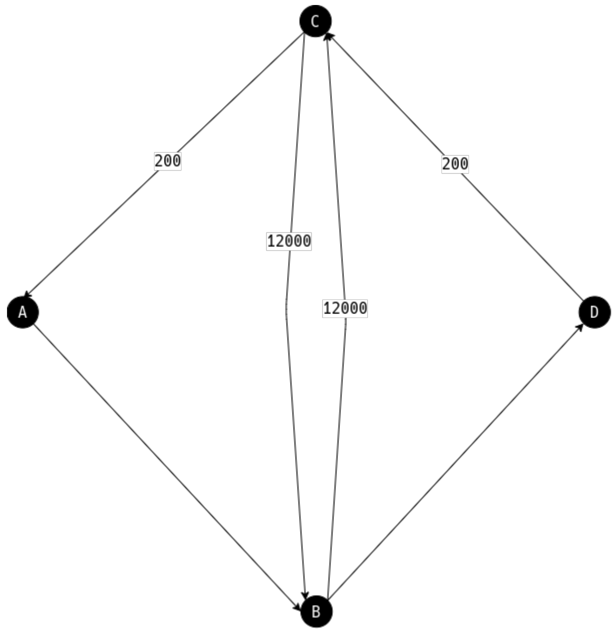
DC = 200
CB = 12000
CD = -1
$\Sigma Y_R \leftarrow$ 12200
$\because$ **200 + 12000 + 0 = 12200**

As above, `CD` should be considered as 0 when calculating ΣY_R. This suggests that for every n batteries, there will be n loops and as a consequence that circuit will have a $n \times n$ R matrix which represents the each resistance per loop in an n-partite graph.

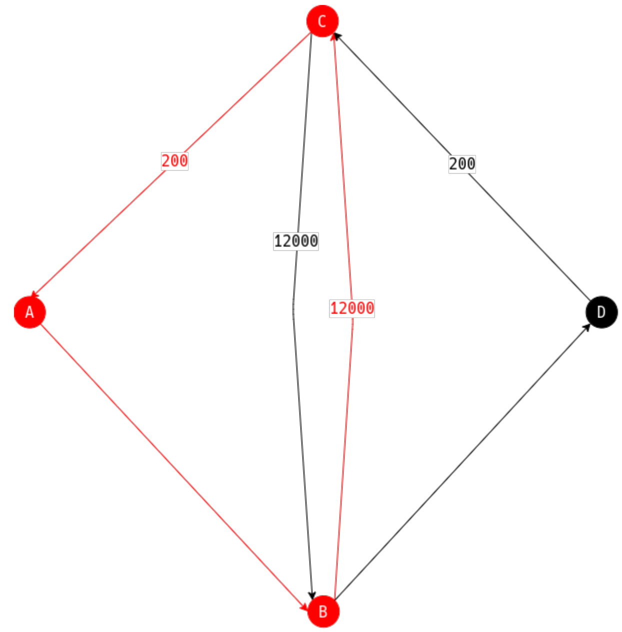
Definition 2.16 (R_A)**.** Let R_A be the total resistance matrix of each loop in the ESP-8266 resistor network.

The resulting total resistance for both loops are 12200Ω. These can be put in a separate 2x2 weighted adjacency matrix R_A as the result $\Sigma X_R \leftarrow 12200$ represents the minimum weight cycle to and from vertex A, thus the total resistance in loop X and $\Sigma Y_R \leftarrow 12200$ represents the minimum weight cycle to and from vertex D, thus the total resistance in loop Y. Therefore, R_A represents the total resistance weighting of each independent loop.

$$\implies R_A = \begin{array}{c} \begin{array}{cc} \text{Loop X} & \text{Loop Y} \end{array} \\ \begin{bmatrix} \Sigma X_R & 0 \\ 0 & \Sigma Y_R \end{bmatrix} \end{array}$$



(a) Illustration of the adjacency matrix A in the ESP-8266



(b) Visual representation for k-NN starting at A

Figure 4: Simplified representation of the ESP-8266

$$\therefore R_A = \begin{array}{c} \begin{array}{cc} \text{Loop X} & \text{Loop Y} \end{array} \\ \begin{bmatrix} 12200 & 0 \\ 0 & 12200 \end{bmatrix} \end{array} \; Ohms$$

With respect to Kirchhoff's laws, the current flow of the circuit will always be from the positive terminal to the negative terminal of the battery. Current is also a scalar value meaning that if a negative current is calculated, it indicates a direction opposite to the assumed positive flow and the absolute value of the calculated current should be used as current is a scalar quantity. However, if a branch between two loops, such as that of loops I_X and I_Y in figure 3a experiences one loop flowing opposite to that of another loop using the branch, a negative resistance value must be associated with the matrix solving for I_A so that I found the total current flowing through a loop using equation 1. The value of I needs to be found using each circuit's loop's resistance and voltage to be matricized.

Equation 4 $\implies rI = V$

- r represents the resistance matrix of the entire circuit
- I represents the current matrix of each loop
- V represents the voltage of each loop

r is different from R as r also contains the resistances of each loop with respect to any adjacent loop resistances. Recall that in figure 3a there are two loops going in opposite direction. If we take the first loop I_X, the current flows through the same branch of I_Y which flows through a 12000Ω resistor. The current will then flow through the 200Ω resistor. This means that for loop I_X, the total resistance for the loop is 12200Ω. However, as there is a 12000Ω that current flow I_Y passes through in the opposite direction, I_X will also experience a difference in voltage of $-12000I_Y$. This applies vice versa as well. I_Y passes through a 12000Ω resistor that I_X flows through in the opposite direction therefore I_Y will also experience a difference in voltage of $-12000I_X$.

Definition 2.17 (I_A)**.** Let I_A be the ideal (on a PCB) total current column of the ESP-8266 resistor circuit containing H_X and H_Y, measured in amps.

Definition 2.18 (I_X)**.** Let I_X be the total current in loop X of the ESP-8266 resistor circuit, measured in amps.

Definition 2.19 (I_Y)**.** Let I_Y be the total current in loop Y of the ESP-8266 resistor circuit, measured in amps.

Definition 2.20 (r_A)**.** Let r_A be the resistance matrix of the entire circuit with respect to any adjacent loop resistances for the ESP-8266.

$$r_A = \begin{bmatrix} 12200 & -12000 \\ -12000 & 12200 \end{bmatrix}$$

These values can be substituted in equation 4 where r is a 2×2 matrix of the values of I_X and I_Y their adjacent resistances and where V is the voltage of each loop.

$$\begin{gathered} r_A \times I_A = V_A \\ \begin{bmatrix} 12200 & -12000 \\ -12200 & 12200 \end{bmatrix} \times \begin{bmatrix} I_X \\ I_Y \end{bmatrix} = \begin{bmatrix} 3.3 \\ 5 \end{bmatrix} \\ \begin{cases} 12200 I_X - 12000 I_Y = 3.3 \\ -12000 I_X + 12200 I_Y = 5 \end{cases} \\ I_X = 0.02071487603, I_Y = 0.02078512397 \quad \text{(Using GDC)} \\ \implies I_A = \begin{bmatrix} 0.02071487603 \\ 0.02078512397 \end{bmatrix} Amps \\ \approx I_A = \begin{bmatrix} 0.021 \\ 0.021 \end{bmatrix} Amps \end{gathered} \tag{1}$$

The resulting I_A represents the ideal current of each loop as components of the whole circuit. If any resulting I is negative, the absolute value should be taken. Due to the resolution of the UNI-T UT33C+ being 2 significant figures, I_A will be rounded to 2 significant figures if being used as a standalone matrix. Otherwise, the unrounded I_A will be used for calculations.

2.4 Modelling on and off a breadboard

Based on figure 3a and matrix A, I can now wire up a prototype based on the ideal circuit with and without a breadboard.

Definition 2.21 (I_{Aw})**.** Let I_{Aw} be the current column of the ESP-8266 resistor circuit modeled without a breadboard, measured in amps.

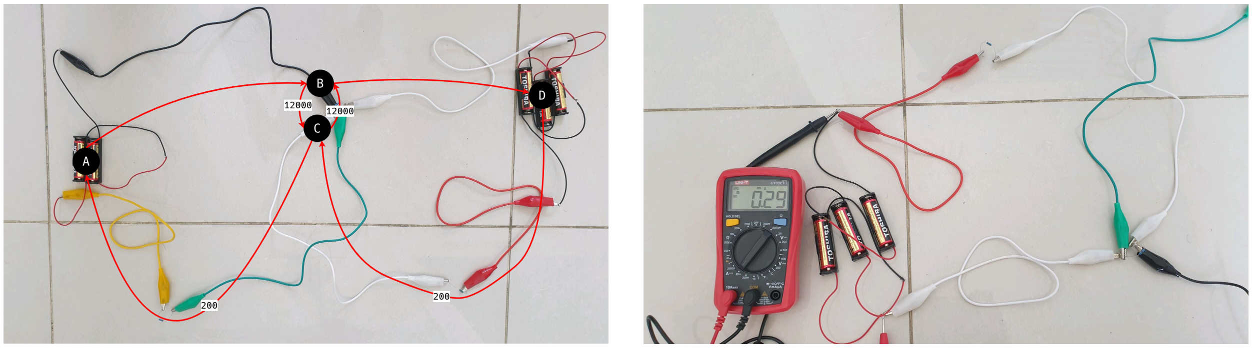
Definition 2.22 (I_{Ab})**.** Let I_{Ab} be the current column of the ESP-8266 resistor circuit with a breadboard, measured in amps.

Using a UNI-T UT33C+ multimeter (see figure 6) with the Ammeter setting on with resolution 20m (reading multiplied by 0.1A) (UNI-T, 2022), I calculated the current between the two terminals (see figure 5b, 6b). In this case, the resulting current for loop I_x was 0.024 A while the resulting current for loop I_Y is 0.021A. Both values can be inserted into a 2×1 column matrix I_{Aw}.

$$I_{Aw} = \begin{bmatrix} 0.027 \\ 0.029 \end{bmatrix} Amps$$

After this, a breadboard model can now be modeled. Like the previous, I calculated the current between the two terminals on both loops I_X and I_Y (see figure 6).

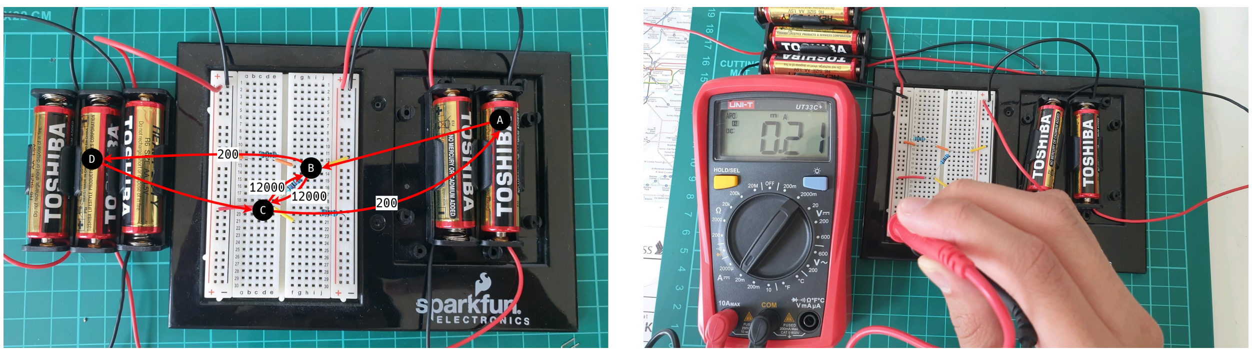
$$I_{Ab} = \begin{bmatrix} 0.021 \\ 0.024 \end{bmatrix} Amps$$



(a) Graph 5a overlayed on the ESP-8266 aligator clips resistor network

(b) ESP-8266 resistor network with alligator clips on loop I_Y being measured

Figure 5: Prototyping the ESP-8266 with alligator clips



(a) Graph 5a overlayed on the ESP-8266 breadboard resistor network

(b) ESP-8266 resistor network on a breadboard's loop I_Y being measured

Figure 6: Prototyping the ESP-8266 resistor network on a breadboard

2.5 Finding the difference in current flow

As I_A represents the ideal current flow for the PCB and I_{Aw} and I_{Ab} represents the current flow in the non-breadboard model and breadboard model respectively, the current difference between the ideal PCB and the modelled non-breadboard and breadboard models should be taken into consideration.

Definition 2.23 (δI_{Aw})**.** Let δI_{Aw} be the current difference column of the ideal ESP-8266 resistor circuit and non-breadboard model (I_{Aw}), measured in amps.

Definition 2.24 (δI_{Ab})**.** Let δI_{Ab} be the current difference column of the ideal ESP-8266 resistor circuit and breadboard model (I_{Ab}), measured in amps.

$$\delta I_{Aw} = |I_A - I_{Aw}|$$

$$\delta I_{Aw} = |\begin{bmatrix}0.02071487603\\0.02078512397\end{bmatrix} - \begin{bmatrix}0.027\\0.029\end{bmatrix}|$$

$$\delta I_{Aw} = \begin{bmatrix}0.006285112397\\0.00821487693\end{bmatrix} Amps$$

$$\delta I_{Aw} \approx \begin{bmatrix}0.0063\\0.0082\end{bmatrix} Amps$$

$$\delta I_{Ab} \approx \begin{bmatrix}0.0029\\0.0032\end{bmatrix} Amps$$

2.6 Calculating energy dissipated and thermal change

After finding the resulting ideal current of each loop, with and without a breadboard, I'd now need to calculate the amount of heat energy dissipated by all components in the loop. From the earlier derived equation 2 ($H = RI^2t$), I needed to rearrange the equation as multiplying 2×1 I^2 with 2×2 R would give a dimension error. This means that equation 2 would need to be rearranged so that H is a 2×1 matrix and I^2 is postmultiplied by R.

Definition 2.25 (H_A)**.** Let H_A be the ideal (on a PCB) heat energy dissipated column of the ESP-8266 resistor circuit containing H_X and H_Y, measured in joules.

Definition 2.26 (H_X)**.** Let H_X be the total heat energy dissipated in loop X of the ESP-8266 resistor circuit, measured in joules.

Definition 2.27 (H_Y)**.** Let H_Y be the total heat energy dissipated in loop Y of the ESP-8266 resistor circuit, measured in joules.

Definition 2.28 (R_X)**.** Let R_X be the resistance of loop X of the ESP-8266 resistor circuit, measured in ohms.

Definition 2.29 (R_Y)**.** Let R_Y be the resistance of loop Y of the ESP-8266 resistor circuit, measured in ohms.

$$\therefore H_A = R_A I_A^2 t$$
$$\implies \begin{bmatrix} H_X \\ H_Y \end{bmatrix} = \begin{bmatrix} R_X & 0 \\ 0 & R_Y \end{bmatrix} \begin{bmatrix} I_X^2 \\ I_Y^2 \end{bmatrix} \times t$$

To now find the ideal amount of heat energy dissipated H_A the values of R_A and I_A would need to be substituted and solved.

$$\implies \begin{bmatrix} H_X \\ H_Y \end{bmatrix} = \begin{bmatrix} 12200 & 0 \\ 0 & 12200 \end{bmatrix} \begin{bmatrix} 0.02071487603^2 \\ 0.020785123976^2 \end{bmatrix} \times 60$$
$$H_A = \begin{bmatrix} 314.1056571 \\ 316.1056571 \end{bmatrix} Joules \quad \text{(Using GDC)}$$

Definition 2.30 (H_{Aw})**.** Let H_{Aw} be the total heat energy dissipated of the ESP-8266 resistor circuit modeled without a breadboard, measured in joules.

Definition 2.31 (H_{Ab})**.** Let H_{Ab} be the total heat energy dissipated of the ESP-8266 resistor circuit with a breadboard, measured in joules.

Now, finding the amount of heat energy dissipated for the breadboard model H_{Ab} and the non-breadboard model H_{Aw} would need I_{Ab} and I_{Aw} be substituted in equation 2 respectively.

$$H_{Aw} = \begin{bmatrix} 533.628 \\ 615.612 \end{bmatrix} Joules \quad \text{(Using GDC)}$$

$$H_{Ab} = \begin{bmatrix} 322.812 \\ 421.632 \end{bmatrix} Joules \quad \text{(Using GDC)}$$

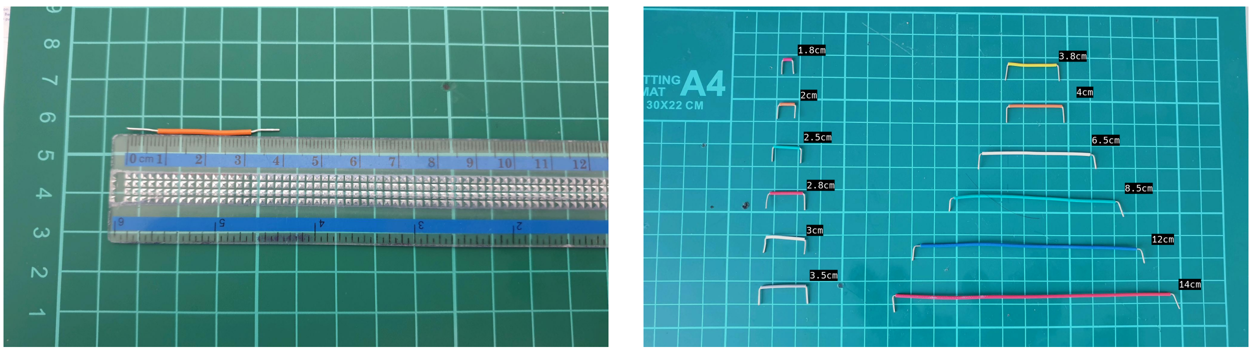
Once we have the matricized values of the total amount heat energy dissipated for an ideal PCB, the breadboard model and the non-breadboard model, the resulting H would need to be substituted into equation 3 ($H = mc \div \Delta t$). From the earlier derived equation 3, the values of c and m are required. Each model has a different material that conducts its current flow as well as a different mass due to the different densities of the materials and different wire lengths.

A PCB, and the ideal medium of current flow and voltage flow in my investigation, has a standard construction of copper conductive traces, form the pathways for current flow. The solder mask, typically applied over the copper layer, prevents unintended contact and corrosion. The PCB model that I usually use for the ESP family and Arduino Uno and is of industry standard incorporate approximately 5.5Oz copper for reliable electrical contacts (Elecaas, 2023). The length of the copper traces is given to its respective scale on the construction sheets of each microcontroller.

Definition 2.32 (Δl_c)**.** Let ΔT_{Aw} be the total temperature change column of the ESP-8266 resistor circuit modeled without a breadboard, measured in degrees Celsius.

Definition 2.33 (ΔT_{Ab})**.** Let ΔT_{Ab} be the total temperature change column of the ESP-8266 resistor circuit modeled with a breadboard, measured in degrees Celsius.

Similarly, both the alligator clips and pre-cut jumper wires I used for the non-breadboard model and connections for the breadboard model respectively used a standard 22-AWG copper. The pre-cut wire jumpers I have on hand and use regularly were all of varying sizes and each had to be measured (see figure 7b) while the crocodile clips I have on hand are all $0.40m$ in length (Botland, 2023), on the other hand, the breadboard wire strips I used all had uniform strips of 8% phosphor bronze (BPS, 2015). Breadboards have internal connections of a set size for the distribution strips at $0.071m$ and circuit connections was at $0.01m$ (BPS, 2024).



(a) Measuring jumper wires

(b) All jumper wire measurements

Figure 7: Measuring jumper wires

Based on the materials used above, self-measured data and online data, I've collated table 2 which contains the specific heat capacity (c) and the masses per cm (m) of each material.

Definition 2.34 (c_{Cu})**.** Let c_{Cu} be the specific heat capacity of 5.5Oz Copper, measured in joules per kilogram per degrees Celsius.

Definition 2.35 (c_{Br})**.** Let c_{Br} be the specific heat capacity of 8% Phosphor Bronze, measured in joules per kilogram per degrees Celsius.

Definition 2.36 (m_{Cu})**.** Let m_{Cu} be the mass per centimeter of 5.5Oz Copper, measured in kilograms.

Definition 2.37 (m_{Br})**.** Let m_{Br} be the mass per centimeter capacity of 8% Phosphor Bronze, measured in kilograms.

Material	**Specific Heat Capacity (c), J/kg/^{o}C**	**Mass per cm (m), kilogram**
5.5Oz Copper	376.812 (c_{Cu})	0.1984464 (m_{Cu}) - 22-AWG Standard
8% Phosphor Bronze	380 (c_{Br})	0.02475 (m_{Br}) - 400-Point Breadboard Standard

Table 2: Masses and specific heat capacities of materials conducting electricity, Raw - (Engineers Edge, 2015), (MatWeb, 2024), (BPS, 2015)

Definition 2.38 (ΔT_A)**.** Let ΔT_A be the ideal (on a PCB) total temperature change column of the ESP-8266 resistor circuit containing H_X and H_Y, measured in degrees Celsius.

Definition 2.39 (ΔT_{Aw})**.** Let ΔT_{Aw} be the total temperature change column of the ESP-8266 resistor circuit modeled without a breadboard, measured in degrees Celsius.

Definition 2.40 (ΔT_{Ab})**.** Let ΔT_{Ab} be the total temperature change column of the ESP-8266 resistor circuit modeled with a breadboard, measured in degrees Celsius.

As both the specific heat capacity and the masses of each material have been calculated, the change in temperature for H_A, H_{Aw} and H_{Ab} can be found.

$$\Delta T_A = H_A \div (m_{Cu} \times c_{Cu})$$

The total length of conductive trace on the ESP-8266's resistor network is 16.8cm (see Table 1)

$$\Delta T_A = \begin{bmatrix} 314.1056571 \\ 316.1056571 \end{bmatrix} \div ((16.8 \times m_{Cu}) \times c_{Cu})$$

$$\Delta T_A = \begin{bmatrix} 0.250033699 \\ 0.2516257346 \end{bmatrix}^o Celsius \quad \text{(Using GDC)}$$

The final ΔT matrix is rounded to 2 significant figures because the UNI-T UT33C+'s maximum resolution is to 2 s.f..

$$\Delta T_A \approx \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}^o Celsius$$

Finding ΔT_{Aw} would require finding the length of all the non-battery wires used. From figure 5b, it can be seen that 7 separate wires were used meaning that the total circuit length of the non-breadboard model is 280 cm.

$$\Delta T_{Aw} \approx \begin{bmatrix} 0.0050 \\ 0.0030 \end{bmatrix}^o Celsius \quad \text{(using GDC)}$$

Definition 2.41 (l_x)**.** Let l_x be the length of the jumper wire conductive trace for the breadboard model in centimeters

Definition 2.42 (l_y)**.** Let l_y be the length of the breadboard conductive trace for the breadboard model in centimeters

Finding ΔT_{Ab} would require finding the length of all the jumper wires used plus the total length of the circuit connections and distribution strips. From figure 7b, it can be seen that 2×3.8, 2×2 and 1×2.8 jumper wires were used, meaning that the wires in the circuit x totals to 9.18 centimeters (l_x) and 4 circuit connections and 4 distribution strips y equaling to 32.4 centimeters (l_y) of the breadboard connections being used. This means two equations must be created with respect to c_{Cu}, m_{Cu} and c_{Br}, m_{Br}.

Definition 2.43 (A_x)**.** Let A_x be the matrix of the change in temperature in the jumper wire for the breadboard model

Definition 2.44 (A_y)**.** Let A_y be the matrix of the change in temperature in the breadboard conductive traces for the breadboard model

$$A_x \leftarrow \begin{bmatrix} 322.812 \\ 421.632 \end{bmatrix} \div ((l_x \times m_{Cu}) \times c_{Cu})$$

$$A_y \leftarrow \begin{bmatrix} 322.812 \\ 421.632 \end{bmatrix} \div ((l_y \times m_{Br}) \times c_{Br})$$

$$A_x = \begin{bmatrix} 0.4702610829 \\ 0.6142185572 \end{bmatrix}, A_y = \begin{bmatrix} 1.059365586 \\ 1.383661173 \end{bmatrix}$$

$$\Delta T_{Ab} = A_x + A_y = \begin{bmatrix} 1.529626669 \\ 1.99787973 \end{bmatrix} \quad \text{using GDC}$$

$$\Delta T_{Ab} \approx \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}^o Celsius$$

2.7 Reflection for the ESP-8266

From the two models' matrices in temperature change ΔT_{Aw} and ΔT_{Ab}, after 1 minute, the change in temperature in loop X of the non-breadboard model was 0.0050^oC and was 0.0030^oC in loop Y. Meanwhile, the change in temperature in loop Y of the non-breadboard model was 1.50^oC and was 2.00^oC for loop Y. From face value, we can see that the non-breadboard model's change in temperature was significantly less than that of the breadboard model where loop X, which experienced 12200Ω of resistance like loop Y, but had $3.3V$ induced in the loop, has a change in temperature that is greater than Loop Y's by 0.0020^oC in the non-breadboard model but 0.5^oC lesser than Loop Y's in the breadboard model. If we compared it to the ideal breadboard model's temperature change, ΔT_A, we see that the ideal change in temperature for loop X and loop Y was 0.25^oC, implying that two models' behaviour in temperature change seems to be less related in the resistor network construction and more towards modelling method. So far, this suggests that each individual loop's temperature change may be dependent on resistors in the loop and circuit construction.

We can attempt to analyze this phenomena by using the total amount of heat energy dissipated in the conductive traces of the circuits. The heat energy dissipated after 1 minute can be used to attempt to deduce the relationship between the current and the change in temperature with respect to the modelling method of the microcontrollers. In the non-breadboard model, the total heat energy dissipation in the non-breadboard model (H_{Aw}) was 533.628 $Joules$ for loop X and 615.612 $Joules$ for loop Y while it was 322.812 $Joules$ for loop X and 421.632 $Joules$ for loop Y in the breadboard model. If total sum of the heat energy matrices was taken to determine if the change in temperature per loop is independent, the resulting value for H_{Aw} would be 1149.24 $Joules$ while the resulting value for H_{Ab} is 744.444 $Joules$. We take the sum of energy change because this signifies the total heat energy dissipated of the circuit without regarding any physical component's mass or specific heat capacity (as outlined in equation 3 ($\Delta T = H \div mc$)). As the matrix sum of H_{Aw} is greater than that of H_{Ab}, it can be implied that the non-breadboard model's heat energy dissipation is greater than that of the breadboard model and indirectly confirms that each individual loop's temperature change is independent because of the different lengths and masses of wire used in the breadboard model versus the uniform lengths of the jumper wires in the non-breadboard model.

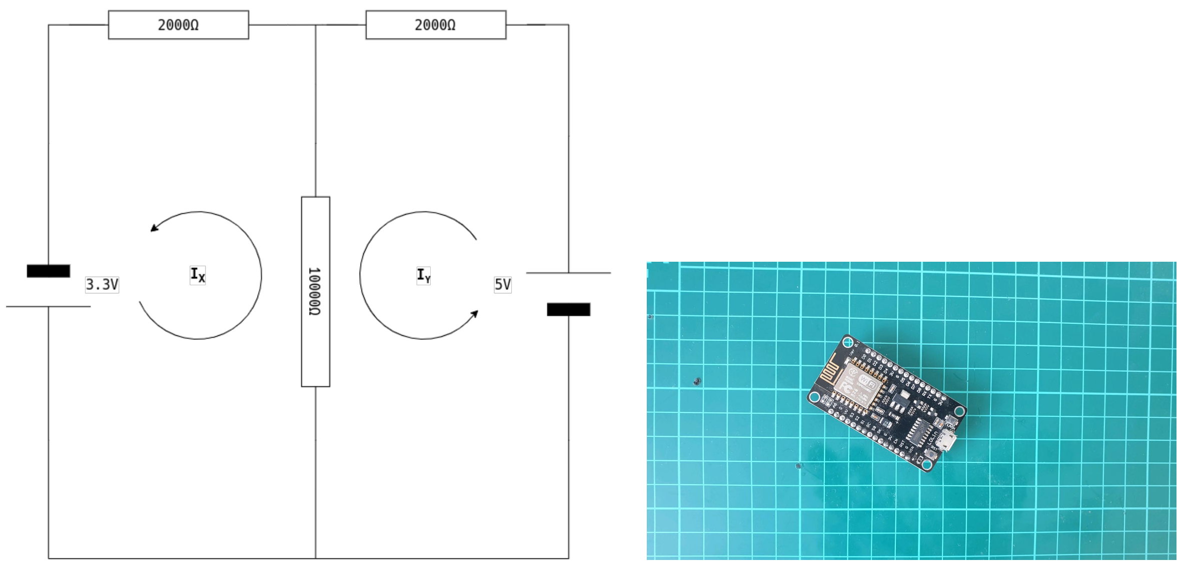
The currents measured is used to determine which model provides the best circuit integrity and can also still be used to analyze the veracity of the claim that the non-breadboard model's change in temperature is solely due to the construction of the models. Using the UNI-T UT33C+ in loop X and loop Y in I_{Aw} and I_{Ab} where the change in current in both loops in the model without a breadboard was 0.0063 $Amps$ in loop X and 0.029 $Amps$ in loop Y while the change in current in both loops with a breadboard was measured to be 0.021 $Amps$ in loop X and 0.024 $Amps$ in loop Y. Loop X for both models' current seem to be less than that of loop Y. In the ideal PCB current, the change for loop X and loop Y was both 0.021 $Amps$. Loop Y's measurement follows the observation that loop Y's current is greater than loop X but **only** at an unrounded value. This rejects my initial hypothesis that loop temperature change is independent of each other and shows that current does not affect the change in temperature of a resistor network due to the varying m and c in equation 3 ($\Delta T = H \div mc$) and how heat energy dissipated due to current outlined in equation 2 ($H = I^2Rt$) is wholly dependent on the construction resistor network (represented by R) and its loops' current (I). This implies that the two models' behaviour in circuit integrity is more related in the resistor network construction and less related to modelling methods - which is opposite to that of the suggestion raised in terms of temperature change.

Overall, this indicates that a non-breadboard model is the best modelling option for thermal management due to lower changes in temperature in its loops and a breadboard model is the best modelling option for circuit integrity due to its current difference being lower than that of a non-breadboard model. This will be further explored with the remaining 2 models.

3 Resistor Network Analysis and Data Collection

3.1 ESP-32

I'll be repeating the method outlined in section 2 for the ESP-32 to the remaining 2 microcontrollers.



(a) Illustration of the resistor network in the ESP-32 (b) The ESP-32

Figure 8: The ESP-32 and its resistor network

Definition 3.1 ($\Sigma_{i=1}^{3} B'$)**.** Let $\Sigma_{i=1}^{3} B'$ be the maximum-possible-routes matrix for the ESP-32.

Definition 3.2 (I_B)**.** Let I_B be the ideal current column of the ESP-32 resistor circuit, measured in amps.

Definition 3.3 (δI_{Bw})**.** Let δI_{Bw} be the current difference column of the ideal ESP-32 resistor circuit and non-breadboard model, measured in amps.

Definition 3.4 (δI_{Bb})**.** Let δI_{Bb} be the current difference column of the ideal ESP-32 resistor circuit and breadboard model, measured in amps.

Definition 3.5 (ΔT_B)**.** Let ΔT_B be the ideal (on a PCB) total temperature change column of the ESP-32 resistor circuit, measured in degrees Celsius.

Definition 3.6 (ΔT_{Bw})**.** Let ΔT_{Bw} be the total temperature change column of the ESP-32 resistor circuit modeled without a breadboard, measured in degrees Celsius.

Definition 3.7 (ΔT_{Bb})**.** Let ΔT_{Bb} be the total temperature change column of the ESP-32 resistor circuit modeled with a breadboard, measured in degrees Celsius.

$$I_B \approx \begin{bmatrix} 0.0020 \\ 0.0021 \end{bmatrix} Amps$$

$$\delta I_{Bw}{}^{12} = \begin{bmatrix} 0.00056 \\ 0.00069 \end{bmatrix} Amps$$

$$\delta I_{Bb}{}^{12} = \begin{bmatrix} 0.00016 \\ 0.00029 \end{bmatrix} Amps$$

$$\Delta T_B = \begin{bmatrix} 0.0013 \\ 0.0014 \end{bmatrix}^o Celsius$$

$$\Delta T_{Bw} = \begin{bmatrix} 0.00023 \\ 0.00027 \end{bmatrix}^o Celsius$$

$$\Delta T_{Bb} = \begin{bmatrix} 0.017 \\ 0.020 \end{bmatrix}^o Celsius$$

$$R_B = \begin{bmatrix} 12000 & 0 \\ 0 & 12000 \end{bmatrix} Ohms \qquad \Sigma_{i=1}^{3} B' = \begin{bmatrix} \mathbf{1} & 1 & 2 & 1 \\ 2 & 3 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 2 & 1 & \mathbf{1} \end{bmatrix} \therefore \text{Valid}$$

3.1.1 Reflection for the ESP-32

Like the ESP-8266, the change in temperature for the non-breadboard model (ΔT_{Bw}) was less than that of the breadboard model (ΔT_{Bb}) and the difference in current for the breadboard model (I_{Bb}) was less than that of the non-breadboard model (I_{Bw}). This pattern holds up in the ESP-32.

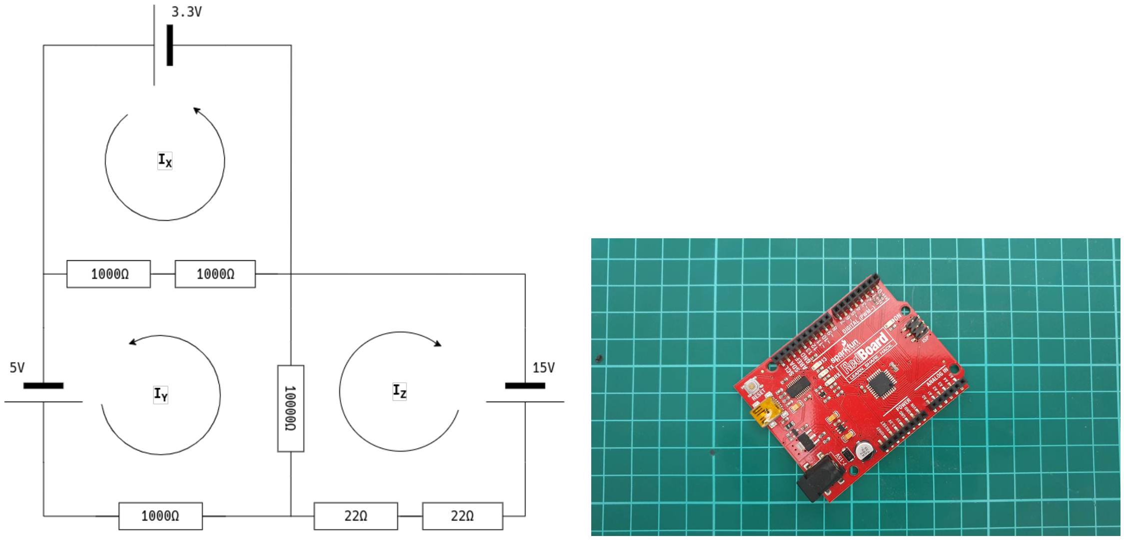
The ESP-32 is touted as a computationally powerful and computationally precise (the ESP-8266 using L106 cache standard while ESP-32 using the LX6 cache standard to allow more instructions at a lower current according to Embedic (2022)) alternative of its predecessor, the ESP-8266. This is reflected in the values of the ideal current matrix for the ESP-32 I_B where $I_B < I_A$ for both loop X and loop Y by a tenth of a decimal place. However, analyzing within the scope of the ESP-32's values shows that the ideal current matrix I_B has loop X's ideal current to be 0.0020 $Amps$ while loop Y's ideal current was 0.0021 $Amps$. The smaller current value compared to the ESP-8266 not only supports the fact that the ESP-32 is more computationally precise than the ESP-8266 but as the ESP-32 has the same form factor and follows the same model construction and voltage input with a different resistor network model implies that the ideal PCB resistor network will have a lower heat energy dissipation value and, by extension, a lower change in temperature versus the ESP-8266. This is corroborated by the ΔT_B where the ESP-32's change in temperature in loop X was 0.0013^o $Celsius$ and was 0.0014^o $Celsius$ in loop Y. Due to the similar construction with the ESP-8266, the pattern where the ideal thermal dissipation in Loop Y is greater than the thermal dissipation in Loop X for the model holds in the ESP-32 as well. This suggests that for the ESP family of microcontrollers, components handled by loop X of the resistor network are handled so due to their need of more precise data transfer and experience lesser thermal dissipation while components handled by loop Y of the resistor network are handled so due to their need for faster processing and experience greater thermal dissipation than loop X.

In terms of the modelling option, the ESP-32's breadboard model and non-breadboard model follow the pattern in the ESP-8266 where $T_{Bw} < T_{Bb}$ and $I_{Bw} < I_{Bb}$ for both loop X and loop Y. For the ESP-32, the ideal total heat energy dissipated was calculated to be 6.3 Joules (see appendix B.2). When modelled on a non-breadboard model, the total heat energy dissipated was 11 joules and when modelled on the breadboard model, the total heat energy dissipated was 7.6 joules. Like the ESP-8266, the element sum of $H_{Bw} > H_{Bb}$ while $T_{Bw} < T_{Bb}$. This holds like the ESP-8266 due to the same construction required, just with a difference in resistor network construction, shown in the resistance matrix r_B. This means that when calculating the change in temperature, the same lengths for the ESP-8266 were used (l_x and l_y), which lead to a lesser current than ESP-8266 being passed through a lesser resistance matrix per loop (diagonal values of R_B) compared to the ESP-8266's diagonal values of R_A thus leading to lesser heat energy dissipated according to equation 2 ($H = I^2Rt$) and with the same masses per centimeter and same specific heat capacities to lead to even lesser changes in temperature than the ESP-8266 due to equation 3 ($\Delta T = H \div mc$). This means that the phenomena outlined in section 2.7 in terms of the breadboard and non-breadboard model also applies to the ESP-32.

Overall, this confirms that for the ESP family of microcontrollers, the breadboard model is the best for circuit integrity while the non-breadboard model is the best for thermal management.

[1] - The UNI-T UT33C+ was set to resolution 2000μ where result is multiplied by $\times 10^{-7}$A
[2] - Calculated using UT33C+ Multimeter

3.2 Arduino Uno



(a) Illustration of the resistor network in the Arduino Uno (b) Adjacency Graph of the Arduino Uno

Figure 9: The Arduino Uno and its resistor network

Definition 3.8 ($\Sigma_{i=1}^{4} C'$)**.** Let $\Sigma_{i=1}^{4} C'$ be the maximum-possible-routes matrix for the Arduino Uno.

Definition 3.9 (I_C)**.** Let I_{Cw} be the ideal current column of the Arduino Uno resistor circuit, measured in amps.

Definition 3.10 (δI_{Cw})**.** Let δI_{Aw} be the current difference column of the ideal Arduino Uno resistor circuit and non-breadboard model (I_{Cw}), measured in amps.

Definition 3.11 (δI_{Cb})**.** Let δI_{Ab} be the current difference column of the ideal Arduino Uno resistor circuit and breadboard model (I_{Cb}), measured in amps.

Definition 3.12 (ΔT_C)**.** Let ΔT_C be the ideal (on a PCB) total temperature change column of the Arduino Uno resistor circuit, measured in degrees Celsius.

Definition 3.13 (ΔT_{Cw})**.** Let ΔT_{Cw} be the total temperature change column of the Arduino Uno resistor circuit modeled without a breadboard, measured in degrees Celsius.

Definition 3.14 (ΔT_{Cb})**.** Let ΔT_{Cb} be the total temperature change column of the Arduino Uno resistor circuit modeled with a breadboard, measured in degrees Celsius.

$$I_C \approx \begin{bmatrix} 0.0047 \\ 0.0064 \\ 0.0078 \end{bmatrix} Amps \qquad \Delta T_C = \begin{bmatrix} 0.15 \\ 1.40 \\ 1.30 \end{bmatrix}^o Celsius$$

$$\delta I_{Cw}{}^{12} = \begin{bmatrix} 0.00069 \\ 0.00034 \\ 0.00068 \end{bmatrix} Amps \qquad \Delta T_{Cw} = \begin{bmatrix} 0.024 \\ 0.19 \\ 0.19 \end{bmatrix}^o Celsius$$

$$\delta I_{Cb}{}^{12} = \begin{bmatrix} 0.00039 \\ 0.00014 \\ 0.00028 \end{bmatrix} Amps \qquad \Delta T_{Cb} = \begin{bmatrix} 5.80 \\ 48.0 \\ 46.0 \end{bmatrix}^o Celsius$$

$$R_C = \begin{bmatrix} 2000 & 0 & 0 \\ 0 & 13000 & 0 \\ 0 & 0 & 10044 \end{bmatrix} Ohms \qquad \Sigma^4 C' = \begin{bmatrix} \mathbf{1} & 2 & 1 & 2 & 3 & 2 \\ 1 & \mathbf{1} & 1 & 2 & 2 & 1 \\ 1 & 1 & \mathbf{1} & 2 & 2 & 1 \\ 3 & 2 & 3 & 4 & 4 & 3 \\ 3 & 3 & 3 & 5 & 5 & 4 \\ 2 & 2 & 2 & 3 & 3 & 2 \end{bmatrix} \therefore \text{Circuit is Valid}$$

3.2.1 Reflection for the Arduino Uno

Unlike the ESP-8266 and ESP-32, the Arduino Uno contains a 3-partite resistor network and is of a larger form factor compared to the ESP family of microcontrollers. As a consequence, previous loop-specific analysis for the ESP family of microcontrollers don't apply for the Arduino Uno. However, the Arduino Uno's ΔT_{Cw} was less than that of ΔT_{Cw} for all 3 column vector elements and the Arduino Uno's δI_{Cb} was less than that of δI_{Cw} for all 3 column vector elements, showing that the pattern where the non-breadboard model has less thermal dissipation compared to the breadboard model and the breadboard model has less difference in current flow compared to the non-breadboard model.

The Arduino Uno is a larger form factor to that of the ESP family of devices and, based on the current dissipated on the ideal PCB model of the Arduino Uno (I_C), the loop with the greatest current flow is loop Z, followed by loop Y and then loop X. Compared to the ESP family, this is less than the ESP-32 but greater than the ESP-8266 in terms of the element values. While a direct difference cannot be calculated due to the different form factor and different dimensions of I_C, I_B and I_A, an average current per loop values can be taken to prove the claim that the circuit integrity of the Arduino Uno middles that of the ESP-32 and ESP-8266.

Definition 3.15 ($\bar{I_C}$)**.** Let $\bar{I_C}$ be the average current per loop of the Arduino Uno

$$\bar{I_C} = \frac{(0.004705856543 + 0.006355856543 + 0.007821442198)}{3}$$
$$\bar{I_C} = 0.006294385094666667\ Amps$$
$$\bar{I_C} = 0.0063\ Amps$$

As seen with $\bar{I_C}$, the mean current flow per loop for the Arduino Uno's resistor network was 0.0063 $Amps$. This calculation can be repeated for the ESP-32 and ESP-8266.

Definition 3.16 ($\bar{I_A}$)**.** Let $\bar{I_A}$ be the average current per loop of the ESP-8266

Definition 3.17 ($\bar{I_B}$)**.** Let $\bar{I_B}$ be the average current per loop of the ESP-32

The value of $\bar{I_A}$ is given as 0.021 $Amps$ and $\bar{I_B}$ is 0.0021 $Amps$. This follows as $\bar{I_A} > \bar{I_C}$ and $\bar{I_C} > \bar{I_B}$, confirming the previous claim. Elaborating on this evaluation of the resistor network construction of the Arduino Uno, I'll have to outline the variations in calculation method for the Arduino Uno non-ideal models to better understand how the changes in temperature and current flow take place in the Arduino Uno model. First, the non-breadboard model of the Arduino Uno non-breadboard model used 360 cm of alligator clips to correspond to the 3-partite construction. This suggests that the dissipation between the increased length of electricity (versus the 280 cm of the ESP models) allows the heat energy dissipated to be dissipated in a circuit of greater length compared to the breadboard model - because for the mass of the wire to be calculated for the non-breadboard model, the length then multiplied with the constant m_{Cu}. So while the heat energy dissipated (H) increases for the Arduino Uno is high due to the low difference between ideal PCB current I_A, δI_{Cw} (see Appendix B.1 for values of I_{Cw}) values and higher R_C values (elaborated on in the next paragraph), it's divided by a greater amount when being operated on with equation 3 ($\Delta T = H \div mc$).

As mentioned in the previous paragraph, H will be greater if I was less and the values of R was lower or equal to a comparable value. This is because equation 2 ($H = I^2 Rt$) can be shown as a directly proportional equation.

$$H = I^2Rt$$

$$Let\ t = k \quad \because \text{t} = 60 \text{ seconds}$$

$$H = kI^2Rt$$

$$\textbf{Equation 5} \implies H \propto I^2R$$

The resulting relationship is called Joule's Law of Heating and suggests that heat energy dissipated in an electrical circuit (H) is directly proportional to the square of the current (I^2) and its resistance (R), where a lower measured I and lower values in a resistance matrix R leads to a greater H (Rutherford, 2023). Contextually, this can be used to determine how the construction of the Arduino Uno resistor network differs from the ESP family of microcontrollers' resistor network. Along with the aforementioned lower $\bar{I_C}$ compared to $\bar{I_B}$, the comparison can be made for R_C and R_B where the average sum of non-zero elements of R_C can be compared with that of R_B.

Definition 3.18 ($\bar{R_B}$)**.** Let $\bar{R_B}$ be the average resistance per loop of the ESP-32

Definition 3.19 ($\bar{R_C}$)**.** Let $\bar{R_C}$ be the average resistance per loop of the Arduino Uno

$$\bar{R_C} = \frac{(2000 + 13000 + 10044)}{3}$$

$$\bar{R_C} = 8348\ Ohms$$

The resulting value of R_C is 8348 *Ohms* while the resulting value of R_B is 12000 *Ohms*. This shows that the average resistance per loop of the Arduino Uno is less than that of the ESP-32 and thus, the ESP-32 will proportionally have less heat energy dissipated due to its greater average resistance per loop and $\approx$ 0.0042 *Amp* difference ($\bar{I_C} - \bar{I_B}$). This confirms how the resistor construction will vary per resistor network model if R remains constant and the values of I change and explains the differences in heat energy dissipated in a loop. Overall, the amount of heat energy dissipated for the models of the Arduino Uno, based on equation 2 ($H = I^2Rt$), shows how Joule's law of heating can scale in this investigation even if there's differences in the amount of loops in a model.

Thus, we can deduce that the Arduino Uno is not as computationally precise as the ESP-32 yet has a larger value for its resistor network along with its greater form factor. Following the findings for the ESP family of microcontrollers, the breadboard model will experience the greatest change in temperature. This combination of previous findings allows us to conclude that the breadboard model of the Arduino Uno will experience a far greater increase in temperature. That being said, ΔT_{Cb} experienced a temperature change of 5.80^o *Celsius* for loop X, 48.0^o *Celsius* for loop Y and 46.0^o *Celsius* for loop Z. These are abnormally high temperatures yet both confirm the validity of the equation methods for finding change in temperature via Ohm's law (equation 1) and heat energy dissipated (equation 2) and confirm the claim instated previously in section 2 and 3.2 that the breadboard model dissipates heat energy in a circuit worse than the non-breadboard model (which experienced a change in temperature of 0.024^o *Celsius*, 0.19^o *Celsius* and 0.19^o *Celsius* respectively).

In context, the resulting values of ΔT_{Cb} and ΔT_{Cw} suggest that the non-breadboard model is the best option for modelling the Arduino Uno in terms of thermal management. Outside the scope of this investigation, this suggests that modelling loops that have a low average resistance per loop should be done using non-breadboard models due to the burn risk they pose. In terms of circuit integrity, the breadboard model still holds as the best option due to the lower current difference for all 3 loops in B_{Cb} however due to the heat risks this poses, the non-breadboard model is a viable option for modelling the Arduino Uno. Furthermore, the ideal PCB model of the Arduino Uno suggests that loops Y and Z acts as the loop Y for the ESP family of microcontrollers where processes requiring faster processing and experience greater thermal dissipation are handled using loops Y and Z compared to loop X of the Arduino Uno resistor network which suggests it handles components requiring more precise data transfers with reduced thermal dissipation.

[1] - The UNI-T UT33C+ was set to resolution 2000μ where result is multiplied by $\times 10^{-7}$A
[2] - Calculated using UT33C+ Multimeter

4 Evaluation

4.1 Validating the ideal circuit values using technology

The current calculations of the breadboard and non-breadboard models are calculated using technology with the UNI-T UT33C+. However, the ideal PCB models' current calculations are done using Ohm's law. To check the validity of the ideal current column matrices I_A, I_B and I_C, I'll use digital software to calculate the currents. One of these softwares is Qucs, my software of choice and is industry standard (MIT, 2003). Initially, we can model using Qucs for the ESP-8266's resistor network circuit based on figure 3a and we can simulate the circuit using Qucs. See appendix C for the settings outline of Qucs.

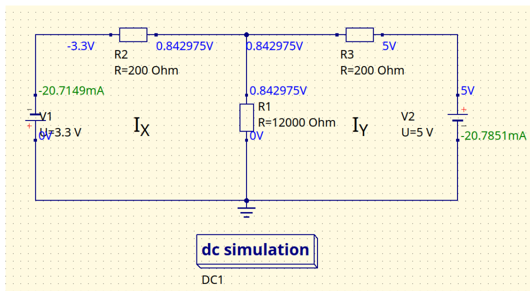


Figure 10: The Qucs simulation for the ESP-8266

The resulting values of the ESP-8266's Qucs simulation are the green values given in milliamps but when converted into amps, ≈ 0.027149 *Amps* for loop X and is ≈ 0.027851 *Amps* and the unrounded current matrix I_A for loop X is equal to 0.02071487603 *Amps* (≈ 0.0207149 *Amps*) and is 0.02078512397 *Amps* (≈ 0.0207851 *Amps*) for loop Y, meaning that the ideal PCB current calculations for the ESP-8266 in section 2.3 is accurate to 6 significant figures. This process can be repeated for the ESP-32 and the Arduino Uno. Note that the absolute value of the simulation values of the ESP-8266 (and further simulation values) are taken because current is scalar.

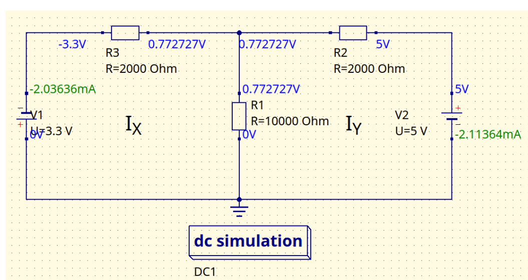


Figure 11: The Qucs simulation for the ESP-32

The resulting values of the ESP-32's Qucs simulation are ≈ 0.00203636 *Amps* for loop X and is ≈ 0.00211364 *Amps* for loop Y. This is equal to the unrounded current matrix I_B where loop X is $0.0020\overline{36}$ *Amps* (≈ 0.00211364 *Amps*) and loop Y is $0.00211\overline{36}$ *Amps* (≈ 0.00211364 *Amps*). Meaning that the ideal PCB current calculations for the ESP-32 in section 3.1 is accurate to 6 significant figures.

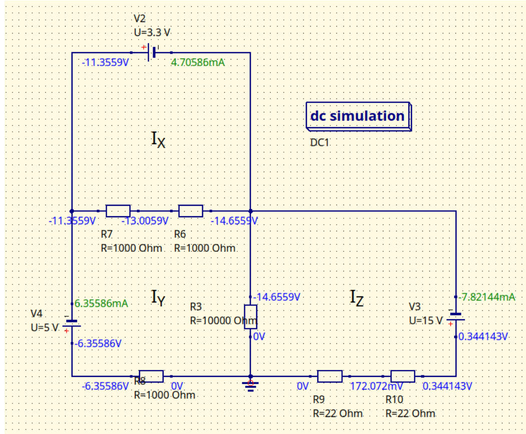


Figure 12: The Qucs simulation for the Arduino Uno

The resulting values of the Arduino Uno's Qucs simulation are ≈ 0.00470586 *Amps* for loop X, ≈ 0.00635586 *Amps* for loop Y and ≈ 0.00782144 *Amps* for loop Z. This is equal to the unrounded current matrix I_B where loop X is 0.004705856543 *Amps* (≈ 0.004705886 *Amps*), loop Y is 0.006355856543 *Amps* (≈ 0.00635586 *Amps*) and loop Z is 0.007821442198 *Amps* (≈ 0.00782144 *Amps*). Meaning that the ideal PCB current calculations for the Arduino Uno in section 3.2 is accurate to 6 significant figures.

Therefore, the calculations outlined and operated for the ideal PCB current in section 2.3 and in appendix B is accurate to 6 significant figures and thus my ideal PCB current calculations in my investigation which use are valid and accurate.

4.2 Further evaluation of circuit integrity of the models with Eigenvalues

While the method outlined in section 2.2 determines whether a short circuit occurs in the resistor networks for the models, which verifies one part of a model's circuit integrity, there needs to be a calculation to determine the long-term integrity of the resistor network models using the eigenvalues of the resistors per loop. This is because eigenvalues represent the stability of of the model's resistances and circuit over time as the components in the circuit, in this case the eigenvalues of the resistors, represent how the resisitivity of the circuit components react to electrical perturbations (Liu et. al., 2017). If resistivity increases over time, temperature increases over time as any given material that a current flows through experiences a decrease in conductivity over time (Heaney, 2003). The greater the resulting eigenvalue, the greater the temperature changes over time and the less stable the circuit current is over time. To analyze this behaviour, the eigenvalues for the resistor network with respect to any adjacent loop resistances can be used to calculate the eigenvalues of the model.

We can first model the ESP-8266 resistor network's eigenvalues to evaluate the ESP-8266 models' circuit integrity.

Definition 4.1 (λ_A)**.** Let λ_A be the eigenvalues of the ESP-8266 resistor network.

Definition 4.2 (I)**.** Let I be the identity matrix.

Using the characteristic equation:

$$\det(r_A - \lambda_A I) = 0$$

$$\det(\begin{bmatrix} 12200 & -12000 \\ -12000 & 12200 \end{bmatrix} - \begin{bmatrix} \lambda_A & 0 \\ 0 & \lambda_A \end{bmatrix}) = 0$$

$$\det(\begin{bmatrix} 12200 - \lambda_A & -12000 \\ -12000 & 12200 - \lambda_A \end{bmatrix}) = 0$$

$$(12200 - \lambda_A)(12200 - \lambda_A) - (-12000 \times -12000) = 0$$

$$(12200 - \lambda_A)^2 - (-12000)^2 = 0 \quad \text{(Using GDC)}$$
$$\lambda_A = 200, 24200$$

The resulting eigenvalues for the ESP-8266 results in two positive eigenvalues for loop X and loop Y. Primarily, this confirms that an n-partite graph with n loops will have n eigenvalues. However, this overall shows that the circuit integrity of the ESP-8266's model is put into question as the eigenvalues are both positive, meaning that the resistor network model is unstable for both loops X and Y. This implies that over time, the resistor stability, and by extension current, will increase away from the origin over time because the resisitivity of the circuit components (λ_A) increases over time, which means that the resistor network has an unstable circuit integrity. I expect this pattern to be present in the ESP-32 and the Arduino Uno because of how the resistor networks lack inductors that store and release energy in response to changes in current, thus mitigating the effects of increasing resistivity over time (Das & Yang, 2017).

Definition 4.3 (λ_B)**.** Let λ_B be the eigenvalues of the ESP-32 resistor network.

Definition 4.4 (λ_C)**.** Let λ_C be the eigenvalues of the Arduino Uno's resistor network.

$$\lambda_B = 2000,\ 22000$$

Similar to the ESP-8266, the ESP-32 resistor networks' eigenvalues are that of positive integers, implying that the ESP-32's resistor network is unstable in the long-term. Similar to the ESP-8266, the eigenvalues are whole, positive integers which also imply that the ESP-32 is unstable in the long-term.

$$\lambda_C = 324.373,\ 2972.409,\ 21747.218$$

The only 3-partite graph in this investigation, the Arduino Uno also has three positive non-integer eigenvalues which correspond to the three loops in the resistor network. This suggests that there are smaller perturbations the Arduino Uno's resistor model experiences compared to the integer eigenvalues of the ESP-8266 and the ESP-32.

From the resulting values, it can be suggested that the resistor models that have been constructed are wholly unstable in the long-term, corroborated by the equation 2 and 5 where a non-constant time will increase heat energy dissipated. This investigation measured the changes in temperature with for 60 seconds and the evaluation carried out eigenvalues indicate that in the long-term, all models' temperature will increase at a greater rate for all models in the long-term and the current, and by extension data transfer stability, is unstable in the long term. This suggests that the resistor networks models for the breadboard, non-breadboard and ideal PCB model is unstable and while no short circuits are present, as outlined in section 2.2, the reliability of the built models should be improved by adding inductors in the resistor networks (Johnson & Graham, 1993).

5 Conclusion

In conclusion, the best model to use when modelling microcontroller resistor networks in terms of thermal management is the non-breadboard model while the best model to use when modelling PCBs in terms of circuit integrity is the breadboard model. This is because the non-breadboard models experience lesser heat energy dissipation in the environment, signalled by its lesser change in temperature compared to the breadboard models. However, the breadboard models excel in circuit integrity as the breadboard models have a lesser difference in current flow, meaning a smaller difference in data transfer rates, against the ideal PCB current flow compared to the difference in current flow of the non-breadboard model and the ideal PCB current flow.

A Bibliography

Johnson, H. W., & Graham, M. (1993). High-speed digital design : a handbook of black magic. Prentice Hall, Cop.

3M. (1994). Solderless Breadboards.

DeCross, M., O'Brien, T., Menon, A., V, V. S., Ross, E., & C, S. (2018). Heat Dissipated by Resistors — Brilliant Math & Science Wiki. Brilliant.org. https://brilliant.org/wiki/heat-dissipated-by-resistors/

Guenin, B. (2002, January 8). Simplified Transient Model for IC Packages.

Engineers Edge. (2015, January 11). Specific Heat Capacity of Metals Table Chart. Engineersedge.com. https://www.engineersedge.com/materials/specific_heat_capacity_of_metals_13259.htm

BPF. (2019). Plastics in Electrical and Electronic Applications. Bpf.co.uk. https://www.bpf.co.uk/innovation/plastics_in_electrical_and_electronic_applications.aspx

MatWeb. (2024). MatWeb - The Online Materials Information Resource. Www.matweb.com. https://www.matweb.com/search/datasheet.aspx?matguid=7d2b1b0c3c3b4ccf96a5a0986e327fa8&ckck=1

Wassem, U. (2023, November 13). What are Circuit Boards Made Of? A Comprehensive Guide to PCB Materials and Manufacturing Processes. Wevolver. https://www.wevolver.com/article/what-are-circuit-boards-made-of-a-comprehensive-guide-to-pcb-materials-and-manufacturing-processes

ABL. (2020, October 15). What are printed circuit boards made of? ABL Circuits. https://www.ablcircuits.co.uk/blog/what-are-printed-circuit-boards-made-of/

UNI-T. (2022). UT33A+/B+/C+/D+ Palm Size Multimeter User Manual.

Elecaas. (2023, April). Printed Circuit Board Services. Elecaas. https://elecaas.com/services/

Botland. (2023). Small Alligator Clip 22AWG cables- 40cm - 12pcs Botland - Robotic Shop. Botland.store. https://botland.store/crocodile-cables/13806-small-alligator-clip-22awg-cables-40cm-12pcs-5904422342197.html

Kinkar Chandra Das, & Yang, Y. (2017). Eigenvalues of the resistance-distance matrix of complete multipartite graphs. Journal of Inequalities and Applications, 2017(1). https://doi.org/10.1186/s13660-017-1570-1

BPS. (2015). BPS DAT (BB400) Datasheet. In Busboard.com. https://busboard.com/documents/datasheets/BPS-DAT-(BB400)-Datasheet.pdf

Liu, X., Zhou, J., & Bu, C. (2015). Resistance distance and Kirchhoff index of R-vertex join and R-edge join of two graphs. Discrete Applied Mathematics, 187, 130–139. https://doi.org/10.1016/j.dam.2015.02.021

MIT. (2003). Qucs - Getting Started. Web.mit.edu. https://web.mit.edu/qucs_v0.0.19/docs/en/start.html

Eland. (2024). FAQ: How does data travel over a cable? — Eland Cables. Www.elandcables.com. https://www.elandcables.co

Embedic. (2022, November 17). ESP32 vs ESP8266, Which is Better and How to Choose? Embedded Technology Information EmbedIc. Www.embedic.com. https://www.embedic.com/technology/details/esp32-vs-esp8266–which-is-better-and-

Rutherford, C. (2018, December 22). 3.2: Ohm's Law, Joules Law, and Series. Workforce LibreTexts; University of Hawai'i OER. https://workforce.libretexts.org/Bookshelves/Construction/Building_Maintenance_and_Construction_-_Tools_and

B Raw Data

B.1 Current

Results table for the resistor network currents (Raw, in Amps)

	X	Y	Z
A	0.02071487603	0.02078512397	
Aw	0.027	0.029	
Ab	0.021	0.024	
B	0.00203636	0.00211363	
Bw	0.0026	0.0028	
Bb	0.0022	0.0024	
C	0.004705856543	0.006355856543	0.007821442198
Cw	0.0054	0.0067	0.0085
Cb	0.0051	0.0065	0.0081

B.2 Heat Dissipated

Table 3: Results table for the resistor network heat energies dissipated (Raw, in Joules)

	X	Y	Z
A	314.1056571	316.1056571	
Aw	533.628	615.612	
Ab	322.812	421.632	
B	2.985679338	3.286115591	
Bw	4.8672	5.6448	
Bb	3.4848	4.1472	
C	564.7027851636	4957.5681035436	4713.5139262
Cw	648	5226	5122.44
Cb	612	5070	4881.384

B.3 Change in Temperature

Table 4: Results table for the resistor network changes in temperature (Raw, in Degrees Celsius)

	X	**Y**	**Z**
A	0.25	0.25	
Aw	0.005	0.003	
Ab	1.5	2	
B	0.0013	0.0014	
Bw	0.00023	0.00027	
Bb	0.017	0.02	
C	0.15	1.4	1.3
Cw	0.024	0.19	0.19
Cb	5.8	48	46

C Determining the resistor networks of the Microcontrollers

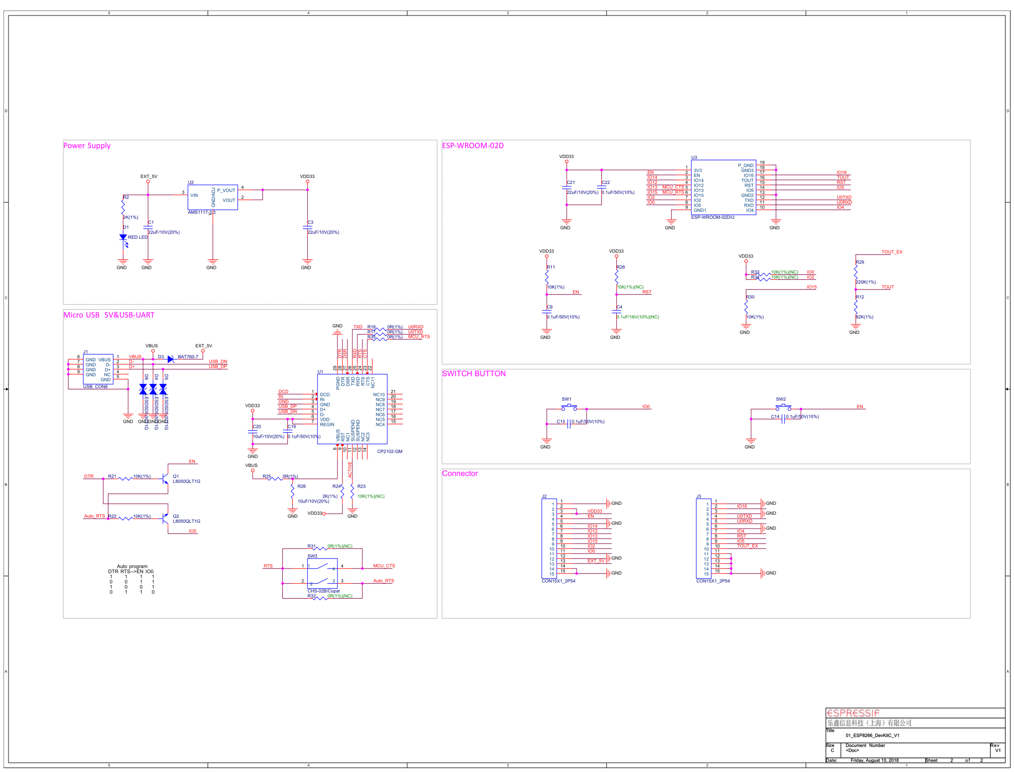


Figure 13: Schematic of ESP-8266

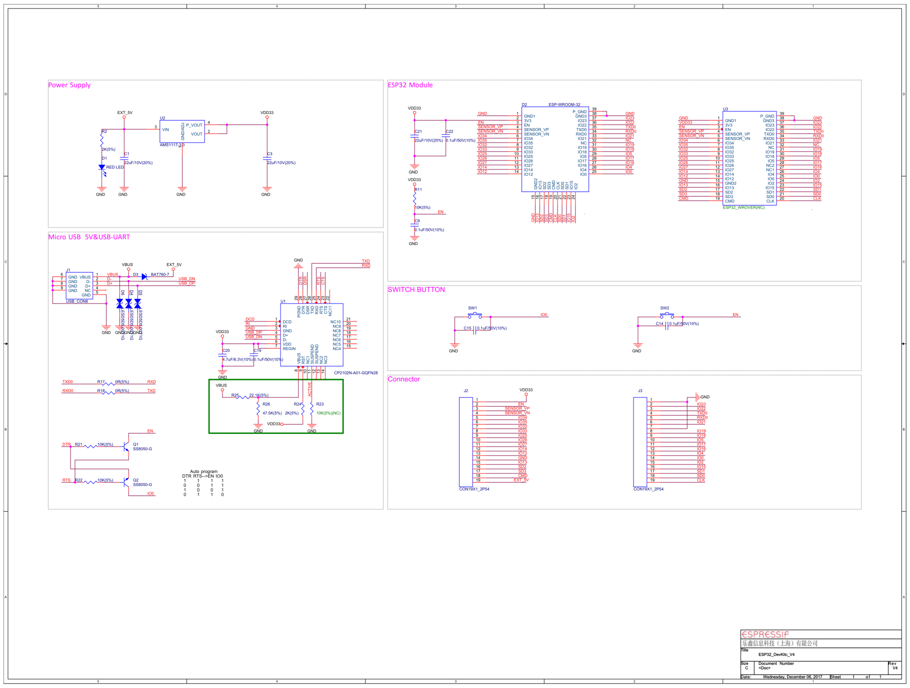


Figure 14: Schematic of ESP-32

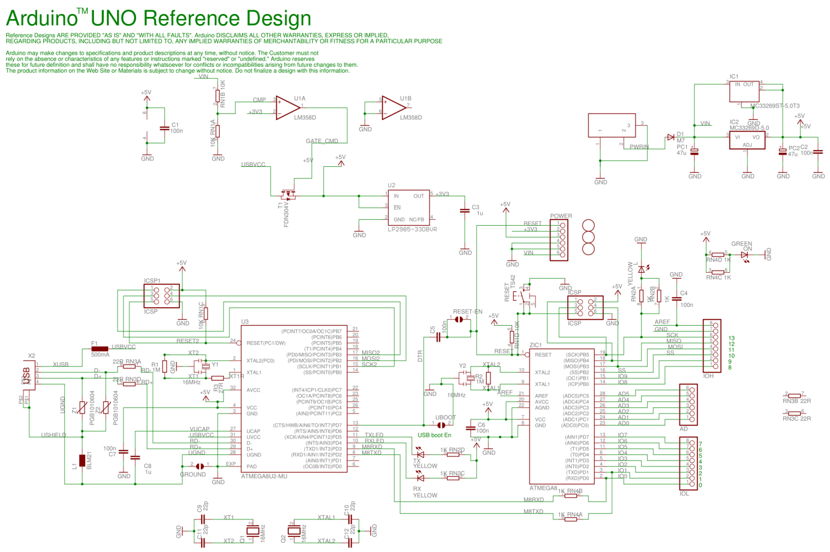


Figure 15: Schematic of Arduino Uno

D JSON File Settings for Qucs

```
[General]
AdmsXmlBinDir=/usr/bin
AscoBinDir=
Attribute=#008080
BGColor=#fffae1
Character=#ff00ff
Comment=#a0a0a4
DefaultSimulator=1
Directive=#008080
EditToolbar=true
Editor=qucs
FileToolbar=true
FileTypes=@Invalid()
GraphAntiAliasing=false
IgnoreVersion=false
Integer=#0000ff
Language=
LargeFontSize=16
NgspiceExecutable=/home/nail_/ngspice
NodeWiring=0
Nprocs=4
OctaveExecutable=octave
OpenVAFExecutable=openvaf
QucsHomeDir=/home/nail_/.qucs
Qucsator=qucsator
Real=#800080
RecentDocs=/home/nail_/tr.sch
S4Q_workdir=/home/nail_/.qucs/spice4qucs
SimParameters=
SimulateToolbar=true
SpiceOpusExecutable=spiceopus
String=#ff0000
Task=#800000
TextAntiAliasing=false
Type=#800000
ViewToolbar=true
WorkToolbar=true
XyceExecutable=/usr/local/Xyce-Release-6.8.0-OPENSOURCE/bin/Xyce
XyceParExecutable=mpirun -np %p /usr/local/Xyce-Release-6.8.0-OPENMPI-OPENSOURCE/bin/Xyce
appFont="Noto Sans,10,-1,0,400,0,0,0,0,0,0,0,0,0,0,1"
compIconsTheme=0
dx=1920
dy=1014
firstRun=false
font="Noto Sans,12,-1,0,400,0,0,0,0,0,0,0,0,0,0,1"
fullTraceName=false
maxUndo=20
panelIconsTheme=0
textFont="Hack,10,-1,7,400,0,0,0,0,0,0,0,0,0,0,1"
x=0
y=40
```