

Inside a Computer:

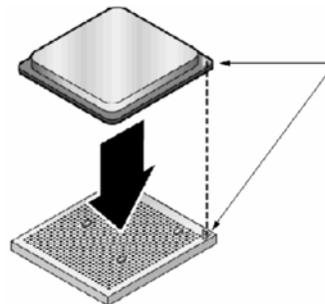
Motherboard: The Motherboard connects the components inside and out of the computer to allow them to communicate with one another.

This is what we will attach all the main components to. Lay the motherboard in front of you.



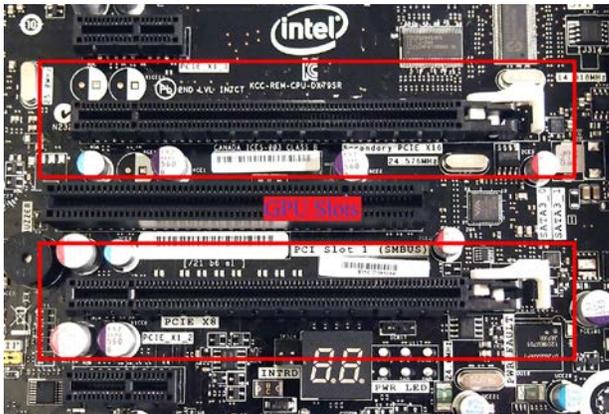
CPU: The CPU (Central Processing Unit) is the brain of the Computer. All of the computational thinking of how to run computer programs and surf the web is done here.

Find the square of either the pins or the holes on the Motherboard. This is the CPU Socket. Carefully grab a CPU, line up the triangles located on the side of the CPU and Socket, then drop it in. It should sit in easily. If not, check its alignment.



GPU: The GPU (Graphical Processing Unit) is a processor specifically designed to handle graphics. Most CPU's and/or Motherboards have this already integrated, which is enough to manage simple non-graphical intensive programs. This component is added for gaming or running simulations. This is what the monitor is plugged into if used.

Insert the GPU into the Video Card Slot with its outer connections next to the Motherboards outer connections. This is shown below.

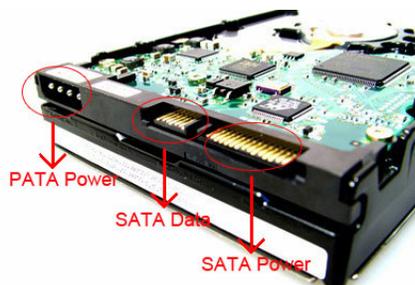


Hard Drive/SSD: The Hard Drive and/or SSD (Solid State Memory) is where all of the programs and files are stored. The Hard Drive connects to the motherboard via a SATA Cable while SSD uses either a SATA Cable or plugs straight into the motherboard via a M.2 PCI-Express port.

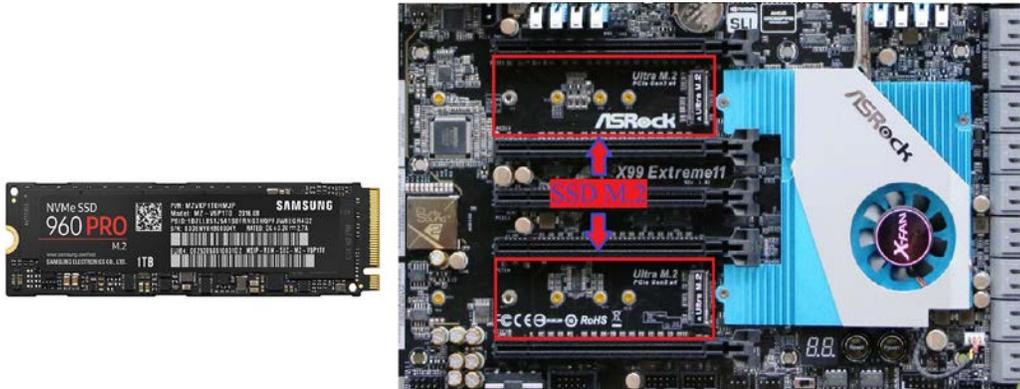
Grab a SATA Cable if needed:



Connect one end to the Hard Drive or SSD and the other to the Motherboard

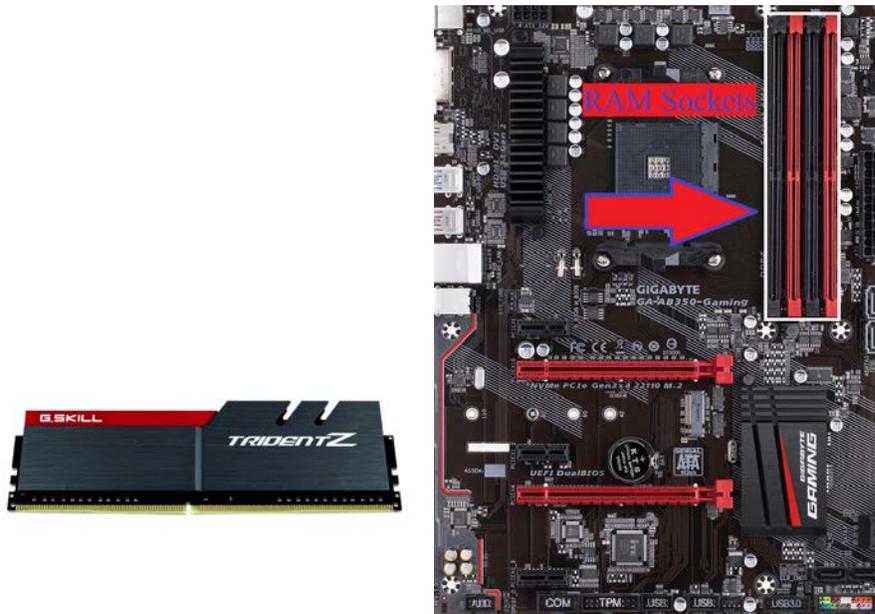


If the SSD connects into the M.2 port:



RAM: The RAM (Random Access Memory) is another form of memory which acts as the computer's short-term memory. It stores data that the computer needs to access quickly in order to help run more programs faster.

Insert the RAM into the long memory sockets by lining up the two sides of the stick (one shorter and one longer). It should sit in easily, then snap the side clamp to secure it.

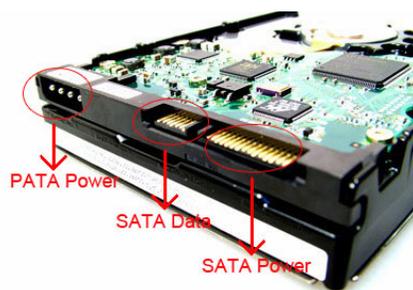


Power Supply: The Power Supply provides power to all of the main components of the computer. These come with different rating, both for power output and efficiency.

The Power Supply has many connectors; it is important to read the instructions that come with the main components to determine which connections connect to which component and where.



Find the connectors which plug into the Hard Drive (if needed), GPU (if needed) and the Motherboard, then connect them.

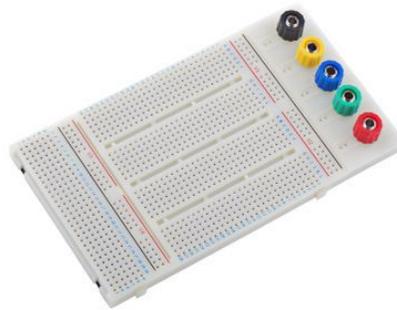


The Motherboard is powered which then sends the necessary power to the CPU and RAM.

It's Lit:

Don't Get Board: The breadboard is a piece of equipment where circuits are often prototyped. Elements can be easily added and removed, and even complex circuits can be modeled on the breadboard. Depending on where on the board the holes are located, they are either connected horizontally or vertically. The terminals at the top of the board are where the power supply and the ground are connected.

This is where the rest of the circuit will be built before it is connected to the Arduino Board.



Lighting the Way: Also known as a light-emitting diode, an LED is an extremely common element. They are most often used as a visible indicator of whether something is "true," when the LED is on, and "false," when the LED is off. They come in many different colors, though the most common is red.

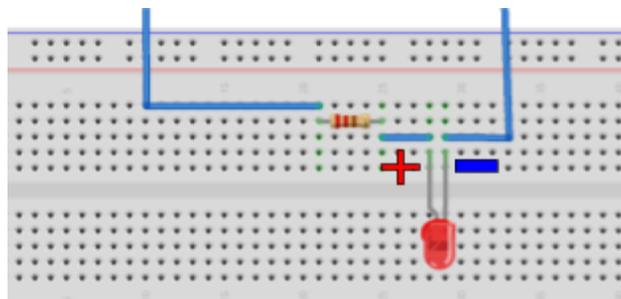


This is what we will use to see if the circuit operates correctly. Attach the LED somewhere in the horizontal batch of holes. Location doesn't matter, but the shorter wire is the negative side, and the longer is the positive. If put in the wrong direction, the LED won't light!

Ohm Sweet Ohm The resistor is another very common, and very critical, element of circuits. They act as a barrier to the flow of electricity, and prevent overloading an element with too much electricity, which can cause damage. The color bands indicate the resistance value, measured in Ohms. The higher the value, the more the resistor limits the flow.

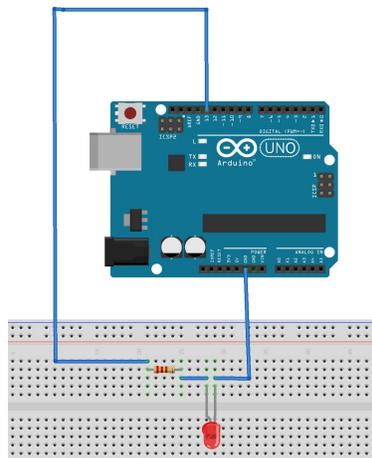


These elements are not directional, unlike the LEDs. Connect a resistor in series with the LED, so that it looks like the figure below. The other lines are wires. You can connect them now as they will come in handy when you connect the circuit to the Arduino.



Arduino: Arduino is a hardware and software company known as open-source. This means the source code, the basic code used to develop the software, is freely available for redistribution and modification. Arduino manufactures microcontrollers, which is a self-contained system with everything you need to drive a circuit, except the power supply. The board used in this activity is known as the Arduino Uno Rev3.

Now the wires from the previous figure come into play. Connect the wires if you haven't already and connect them to the Arduino board. Once you have completed this, connect the Arduino board to the the computer using the micro USB cable.



C: Now we get to the coding. For this board, the UNO uses something known as Arduino C, a simplified version of the popular C++. The code is typed into the Arduino Integrated Development Environment (IDE) open on the computer, then loaded onto the board. There is a basic code already typed into the IDE. Currently, it is set to turn the LED on. Load the program, and watch the LED turn on.

Once you have completed this, you can begin to experiment. Instead of turning the LED on, you can set the LED to flash with just two more lines of code. Add a delay after the line that turns it on, keeping in mind that values are in milliseconds which means 1000 milliseconds is 1 second:

```
delay(1000) //Wait for 1000 milliseconds (1 second)
```

(The `//` indicates a comment; anything that follows after it the IDE will ignore.)

Then add a line to turn the LED off. This is as simple as copying a line from earlier in the code and changing just one word. Once this is done, load the program again. The board will continue to run the program from the beginning to the end with no further input, so the LED will flash. Increasing the delay will decrease the frequency of the blinks, and decreasing the delay will do just the opposite.