**PROJECT REPORT**

**ON**

**Claytronic and Future of Self Assembling Robots**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**



**Submitted to:**

**Er. Prabhjot Kaur**

**ASSOSIATE PROFESSOR**

**CSE-DEPARTMENT**

**Submitted By:**

**GUNEET SINGH LAMBA**

**13BCS1077**

**HARMANDEEP SINGH**

**13BCS1091**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

**May 2017**

**ACKNOWLEDGEMENT**

We sincerely express our indebtedness to our esteemed faculty members of Computer Science Without their kind patronage, this project wouldn’t have taken shape. We also extend our gratitude our project mentor for their valuable information on the software and hardware part of our project and their advices have been an encouragement to us.

**NAME**

GUNEET

SINGH

**ABSTRACT**

Claytronics is an abstract future concept that combines nanoscale robotics and computer science to create individual nanometer-scale computers called claytronic atoms, or catoms. These catoms are like atoms we studied in our science books. Atoms are present in every matter and are responsible for the shape of a particular matter. Similarly catoms can be transform into such a manner so that they can act like atoms and can transform into any shape. Catoms are programmable gadgets which can be programmed to work in specific manners, that they communicate with other catoms to form 3D structures. Now the size may vary from nanoscale cube to big helium balloon. The very first catoms were developed by researchers in year 2005, at Carnegie Mellon University in Pittsburgh, Pennsylvania.

The previous claytrons were made of big electromagnets and power consumption was very high. The processing speed to join with other catoms was very slow. The catoms made in year 2005 by researchers at Carnegie Mellon University in Pittsburgh, Pennsylvania, were bigger in size and shape. In present, the availability of objects when required and to form shapes according to that is not possible by present robots. Robot creation is done with a single anatomy which is not at all flexible and dedicated to a single architecture only. Clone formation has never been done before by any robot either by big robots or nano robots. The algorithms for claytrons work are localization algorithm and shape sculpting algorithms are not well defined by previous processes.

**Requirements:-**

Hardware:-

AVR Microcontroller [Arduino]

Dc motors

NEODYNIUM MAGNETS

cubical structures.

Software:-

Arduino environment.

Proteus.

**Description :-**

**Hardware:-**

## Arduino:-

[Arduino](http://arduino.cc/) is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a [microcontroller](http://en.wikipedia.org/wiki/Microcontroller)) and a piece of [software](http://arduino.cc/en/Main/Software), or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

**What’s on the board:-**

### **Power (USB / Barrel Jack)**

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply ([like this](https://www.sparkfun.com/products/8269)) that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our [Installing and Programming Arduino](https://learn.sparkfun.com/tutorials/installing-arduino-ide) tutorial.

### **Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)**

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a [breadboard](https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/) and some [wire](https://learn.sparkfun.com/tutorials/working-with-wire). They usually have black plastic ‘headers’ that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

* **GND (3)**: Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
* **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
* **Analog (6)**: The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a [temperature sensor](https://www.sparkfun.com/products/10988)) and convert it into a digital value that we can read.
* **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
* **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have [a tutorial on PWM](https://learn.sparkfun.com/tutorials/pulse-width-modulation), but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
* **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

### **Reset Button**

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn’t usually fix any problems.

### **Power LED Indicator**

Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED next to the word ‘ON’ **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time to re-check your circuit!

### **TX RX LEDs**

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for [serial communication](https://learn.sparkfun.com/tutorials/serial-communication). In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we’re loading a new program onto the board).

### **Main IC**

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC’s from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC’s, reading the datasheets is often a good idea.

### **Voltage Regulator**

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it’s for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don’t hook up your Arduino to anything greater than 20 volts.

**Software:-**

Proteus Design Suite ,Arduino C

**Table of Contents :**

|  |  |
| --- | --- |
| Introduction about project | Introduction about project and history |
| Abstract knowledge of project | Full abstract knowledge about project |
| SRS | SRS document of project |
| Architecture Diagram | It consits of DFD models |
| Project methodology | Includes working requirements of hardware, software, conclusion and  Future scope |
| Reference | Reference took from books and websites to build this project |

**Introduction**

Claytronics is an abstract future concept that combines nanoscale robotics and computer science to create individual nanometer-scale computers called claytronic atoms, or catoms. These catoms are like atoms we studied in our science books. Atoms are present in every matter and are responsible for the shape of a particular matter. Similarly catoms can be transform into such a manner so that they can act like atoms and can transform into any shape. Catoms are programmable gadgets which can be programmed to work in specific manners, that they communicate with other catoms to form 3D structures. Now the size may vary from nanoscale cube to big helium balloon. The very first catoms were developed by researchers in year 2005, at Carnegie Mellon University in Pittsburgh, Pennsylvania. First of all we made two algorithms which we derived from the definition of basic claytronics algorithm that were localization and shape sculpting algorithms. The first algorithm we made is Centre point algorithm for forming shapes from a centre point. This comes under shape sculpting algorithm. The second algorithm is master-slave algorithm for localization of catoms and to arrange them in proper order. For allocating resources we are using Master-Slave Algorithm for scheduling and allocating resources to catoms according to their priorities. The formation of 3D shapes can be done easily and effectively by our technology. Nano batteries are used to power single claytron and inductive charging techniques such as Faraday's law of induction and Tesla tower for wireless charging is used for charging the claytrons. Electromagnets are powerful and can be magnetize and demagnetize easily, so small and powerful electromagnets are used for joining the two catoms. Video calling is being made real from virtual by creating a 3D formation of faces with the help of claytronic kit. Claytrons also contains sensors like touch sensor, motion sensor so that one can feel the real simulation of other person on either end doing video call. Also these sensors can help in giving any command to claytron to get into any shape or send signals. Microcontroller of 32-bit is being used in every claytron for fast processing. Receiver and transmitter is installed to send and receive signals so that claytrons can communicate with each other easily.

**SRS DOCUMENT**

**Introduction**

Claytronics is an abstract future concept that combines nanoscale robotics and computer science to create individual nanometer-scale computers called claytronic atoms, or catoms. These catoms are like atoms we studied in our science books. Atoms are present in every matter and are responsible for the shape of a particular matter. Similarly catoms can be transform into such a manner so that they can act like atoms and can transform into any shape. Catoms are programmable gadgets which can be programmed to work in specific manners, that they communicate with other catoms to form 3D structures. Now the size may vary from nanoscale cube to big helium balloon. The very first catoms were developed by researchers in year 2005, at Carnegie Mellon University in Pittsburgh, Pennsylvania. The previous claytrons were made of big electromagnets and power consumption was very high. The processing speed to join with other catoms was very slow. The catoms made in year 2005 by researchers at Carnegie Mellon University in Pittsburgh, Pennsylvania, were bigger in size and shape. In present, the availability of objects when required and to form shapes according to that is not possible by present robots. Robot creation is done with a single anatomy which is not at all flexible and dedicated to a single architecture only. Clone formation has never been done before by any robot either by big robots or nano robots. The algorithms for claytrons work are localization algorithm and shape sculpting algorithms are not well defined by previous processes.

**Structure of SRS document:**

**Requirements:-**

Hardware:-

AVR Microcontroller [Arduino]

Dc motors

Solenoids

metal cubical structures.

Software:-

Arduino environment.

Protius Design Suite

**Hardware:-**

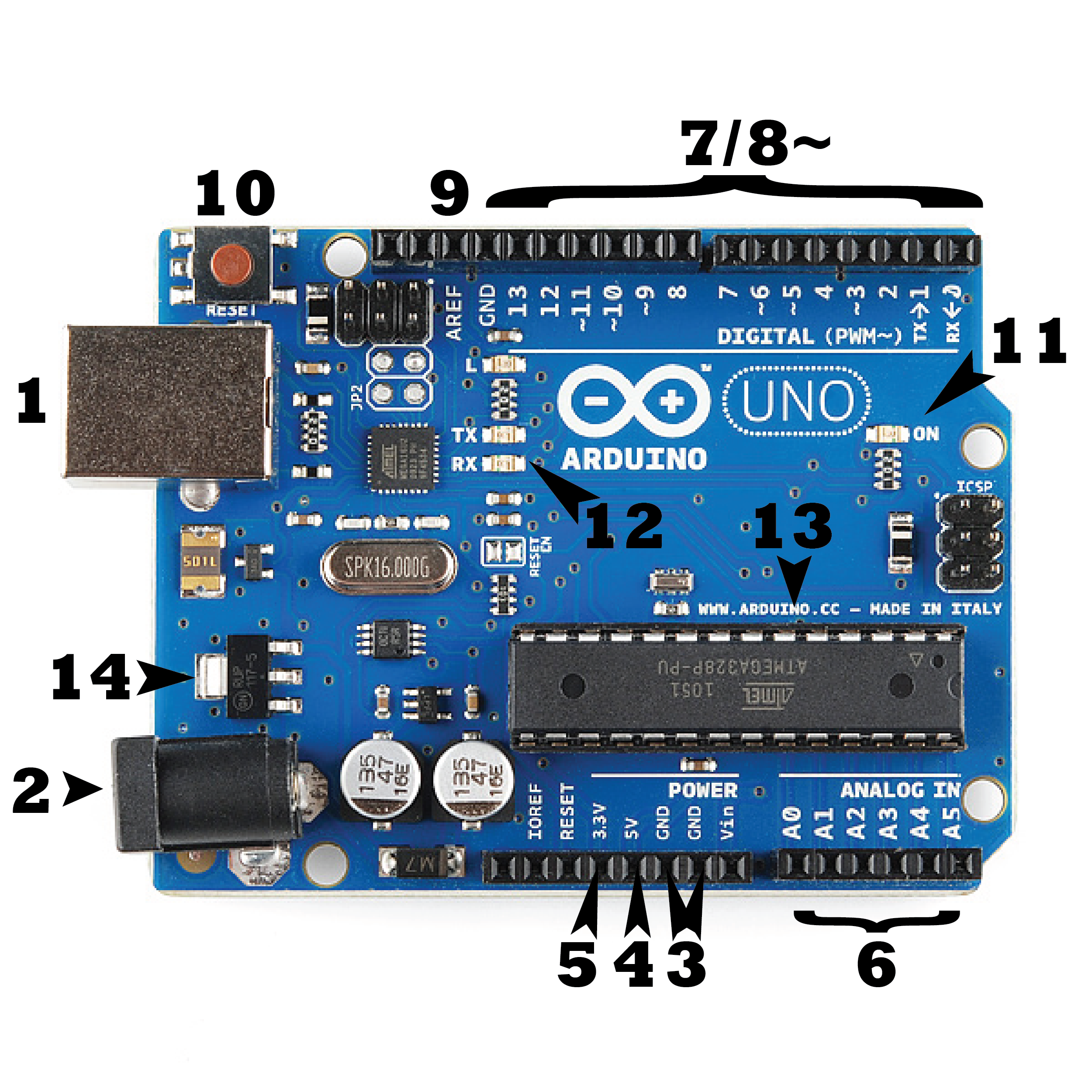
Arduino:-

[Arduino](http://arduino.cc/) is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a [microcontroller](http://en.wikipedia.org/wiki/Microcontroller)) and a piece of [software](http://arduino.cc/en/Main/Software), or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

****

Arduino Uno Board



### Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply ([like this](https://www.sparkfun.com/products/8269)) that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our [Installing and Programming Arduino](https://learn.sparkfun.com/tutorials/installing-arduino-ide) tutorial.

### Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a [breadboard](https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/) and some [wire](https://learn.sparkfun.com/tutorials/working-with-wire). They usually have black plastic ‘headers’ that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

* **GND (3)**: Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
* **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
* **Analog (6)**: The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a [temperature sensor](https://www.sparkfun.com/products/10988)) and convert it into a digital value that we can read.
* **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
* **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have [a tutorial on PWM](https://learn.sparkfun.com/tutorials/pulse-width-modulation), but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
* **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

### Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn’t usually fix any problems.

### Power LED Indicator

Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED next to the word ‘ON’ **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time to re-check your circuit!

### TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we’re loading a new program onto the board).

### Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC’s from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC’s, reading the datasheets is often a good idea.

### Voltage Regulator

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it’s for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don’t hook up your Arduino to anything greater than 20 volts.

Software:-

### Protius , Arduino C

The Proteus Design Suite is an [Electronic Design Automation](https://en.wikipedia.org/wiki/Electronic_Design_Automation) (EDA) tool including schematic capture, simulation and PCB Layout modules. It is developed in [Yorkshire](https://en.wikipedia.org/wiki/Yorkshire), England by Labcenter Electronics Ltd with offices in North America and several overseas sales channels. The software runs on the [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) operating system and is available in English, French, Spanish and Chinese languages..

**Design :**

The hardware for our design is made of light metal with the mixture of silicon. There is a microprocessor along with receiver and transmitter to receive and send signals which are in coded form which will only be executed by microprocessor. Microprocessor is designed in such a manner that it transmits and processes these signals. Further we used artificial magnets which are activated by us when to get magnetize or demagnetize according to our program requirement. The shape and structure of our catoms is shown below.

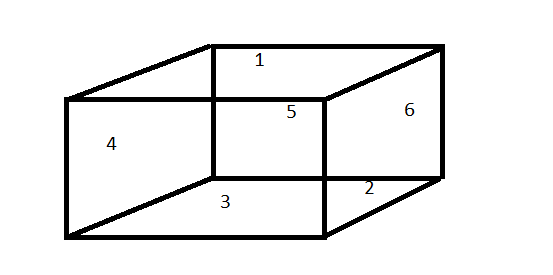
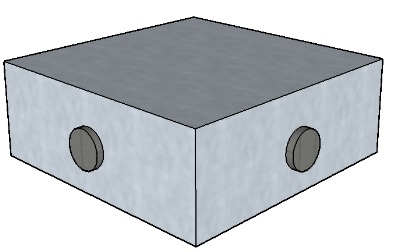
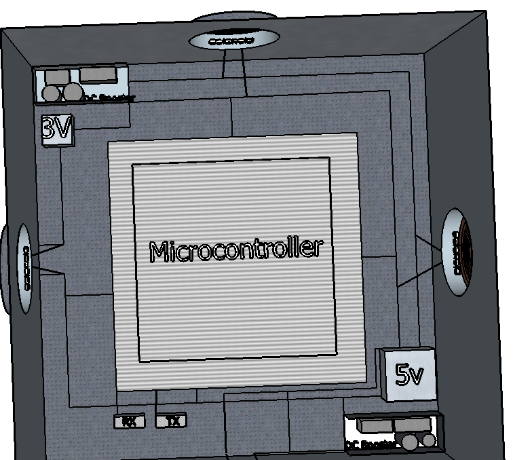


Figure 1 Figure 2

**figure 3**

**CHARACTERISTICS:**

1. We are working at big scale so more catoms are being generated.

2. Our algorithms are based on the basics of claytronics, which can be implemented easily.

3. Our algorithm provides useful method to form 3D shapes and clone formation.

4. Each claytron or catom has its own computer so intelligence is at high level.

6. It is based on basic science.

7. In future advanced version of claytronics can be used in creating claytronic clones of astronaut to send them in to space and whatever astronauts do on earth the clone performs the same in the space and research becomes easier.

8. Communication is be better with claytronics, as by video calling one can be able to see others claytronic clone in front of them

9. Shapes of buildings or design easier to make with our algorithms for claytronics as one can see actual 3D structures of designs.

**DFD**

**Introduction**

Claytronics is an abstract future concept that combines robotics and computer science to create individual computers called claytronic atoms, or catoms. These catoms are like atoms we studied in our science books. Atoms are present in every matter and are responsible for the shape of a particular matter. Similarly catoms can be transform into such a manner so that they can act like atoms and can transform into any shape. Catoms are programmable gadgets which can be programmed to work in specific manners, that they communicate with other catoms to form 3D structures. Now the size may vary from nanoscale cube to big helium balloon. The very first catoms were developed by researchers in year 2005, at Carnegie Mellon University in Pittsburgh, Pennsylvania. The previous claytrons were made of big electromagnets and power consumption was very high. The processing speed to join with other catoms was very slow. The catoms made in year 2005 by researchers at Carnegie Mellon University in Pittsburgh, Pennsylvania, were bigger in size and shape. In present, the availability of objects when required and to form shapes according to that is not possible by present robots. Robot creation is done with a single anatomy which is not at all flexible and dedicated to a single architecture only. Clone formation has never been done before by any robot either by big robots or nano robots. The algorithms for claytrons work are localization algorithm and shape sculpting algorithms are not well defined by previous processes.

**Design :**

The hardware for our design is made of light metal with the mixture of silicon. There is a microprocessor along with receiver and transmitter to receive and send signals which are in coded form which will only be executed by microprocessor. Microprocessor is designed in such a manner that it transmits and processes these signals. Further we used artificial magnets which are activated by us when to get magnetize or demagnetize according to our program requirement. The shape and structure of our catoms is shown below.

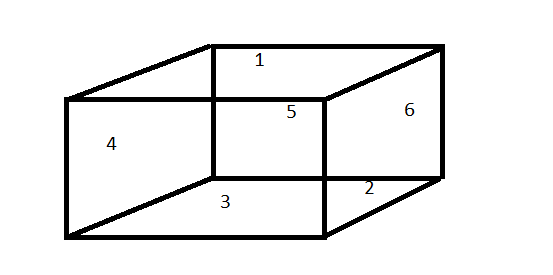
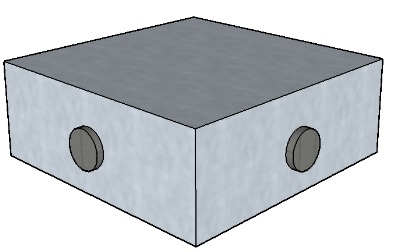
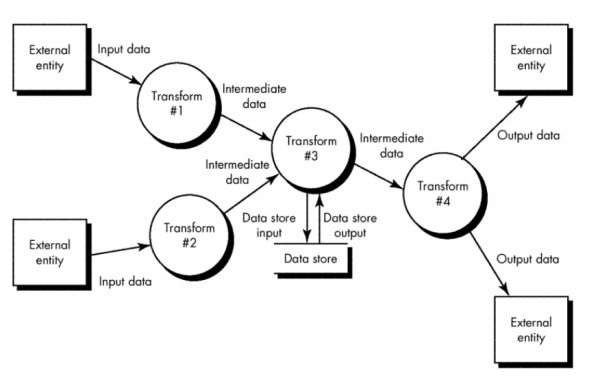
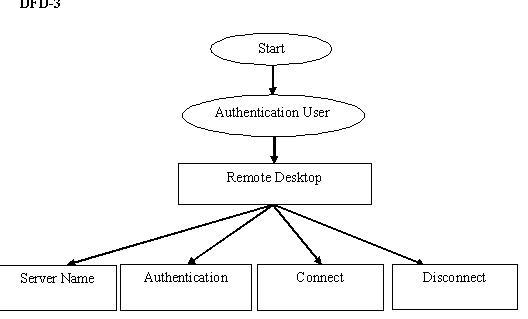


Figure 1 Figure 2

**DFD LEVEL 2:**



**DFD LEVEL 3:**

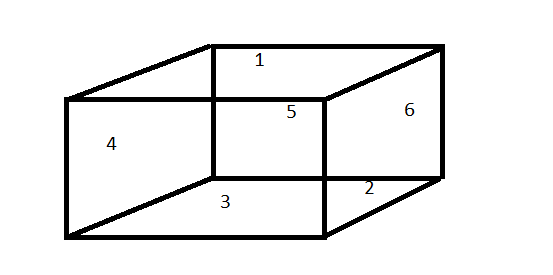
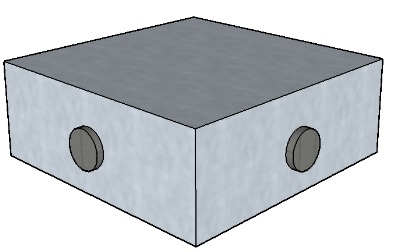


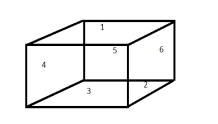
**Description:**

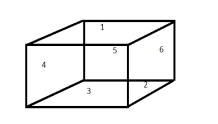
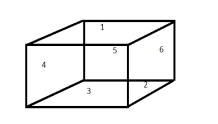
This includes working of our project and further the problems we faced and encountered during making it.

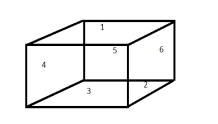
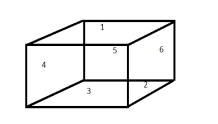
**Working:**

There are 'n' number of catoms. We took a simple case of forming a square base out of it. We already discussed two algorithms center point algorithm and master-slave algorithm. We used them to make structure. The catoms are in cubical form and each side of cube contains an artificial magnet with number assigned AM1-AM6. These magnets can be magnetize and demagnetize and polarity can be assigned according to choice and shape and we can control which part of the catom we want to magnetize or demagnetize. Below are some catoms randomly distributed:-



1 

2 3 

4  5 

The above catoms are randomly distributed. We made a straight line out of it. So for the whole working, below are the steps that we used on the basis of our algorithms:

1. First run random function generator to generate a random number to select a particular catom.

2. Catom number 2 comes out to be result of random function.

3. Now apply master-slave algorithm and catom number 2 becomes master catom to control other catoms.

4. The catom number 2 sends information to its computer to get information and generate an output.

5. The computer generates an output by activating AM-6(artificial magnet on sixth side) as South Pole of master catom number 2. As the AM-6 is South Pole and its slave has to be of different polarity to get attached. The next catom is decided by master catom for allocating resources, same used here to decide what catom be selected next. Next catom comes out to be catom number 1.

Note: The very first catom is only be selected by random function generator. Others are decided by master catom.

6. The master catom sends signal to next slave catom number 1 responding that it has activated south pole at sixth magnet and in order to get attach it must activate north pole on whatever side except 6 as it contains same polarity.

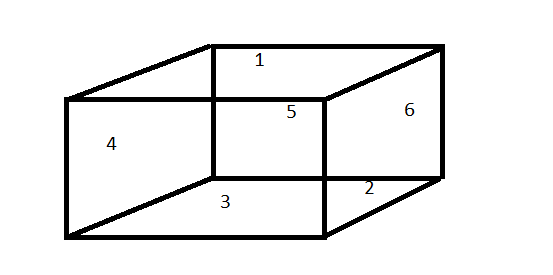
7. The computer processes the signal and activates AM-4(artificial magnet at fourth side) of catom number 1 as North Pole by changing the polarity that is changing it to North Pole and they get attracted, AM-6 of catom number 2 and AM-4 as being of opposite charges.

8. As the two catoms got attached now again master claytron keeps checking which catom should be selected next. The next catom comes out to be catom number 4. As the catom number 1 have AM-4(artificial magnet at fourth side) activated as North Pole so next catom gets attached to its sixth side so for that AM-6(artificial magnet at sixth side) of catom number 4 gets activated. For catom number 4 the master catom is catom number 1 and catom number 1 is the sub slave of main master catom number 2 and sub slave catom number 1. Due to opposite charges they get attached to each other.

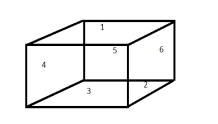
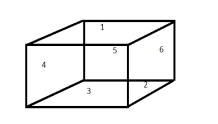
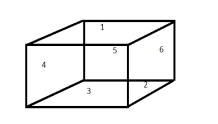
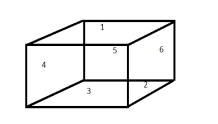
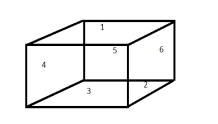
9. Similarly it works with other catoms until all the catoms not form the required structure.

10. After all this a straight line is formed with catoms. Here we took only a simple example to explain how the catoms really work and form structures. More complex structures require more masters and slaves.

NOTE: It is not fixed that AM-6 and AM-4 can be used. You can activate any magnet and set polarities according to requirements. You can also activate AM-3 and AM-5 for straight line.



A single catom assigned with AM1- AM6 on each side

21453

Catoms connecting with each other to from straight line

CATOM (2): Master slave and AM-6 is activated as South Pole.

CATOM (1): Sub slave of master 2 and AM-4 is activated as North Pole

CATOM (2) and CATOM (1) gets attached.

CATOM (1): master for CATOM (4) and AM-6 activated as South Pole.

CATOM (4): Sub slave of master CATOM (1) and AM-4 activated as North Pole.

CATOM (1) and CATOM (4) gets attached.

CATOM (5): master for CATOM (3) and AM-6 activated as South Pole.

CATOM (3): Sub slave of master CATOM (1) and AM-4 activated as North Pole.

CATOM (5) and CATOM (3) gets attached.

A straight line is formed of catoms.

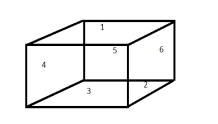
The working of catoms varies from shape to shape and from size to size. Now suppose you want to generate a vertical straight line in which each catom comes over the another catom. The whole procedure is same only the magnetic properties and charges get changed.

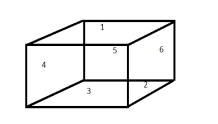
like if :

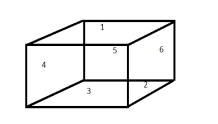
CATOM (2): Master slave and AM-1 is activated as North Pole.

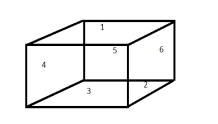
CATOM (1): Sub slave of master 2 and AM-3 is activated as North Pole.

Due to opposite charges and polarity CATOM (1) gets overlap CATOM (2) and comes over it. Similarly the process is same as of straight line the next come over the other.









overlapping of catoms to form vertical lines

These are the working steps for understanding basics of claytronics. The complex structures are made up of complex shapes and at that time more masters and slave are formed but the whole process is based upon the two algorithms we derived centre point algorithm and master-slave algorithm. For example making a hand like structure need many catoms and many master and slave catoms works to form structure. At that time master- slave algorithm take care of which catoms be placed where and for sculpting centre point algorithm is used but the complexity gets increased.

**Problems encountered:**

In present, the availability of objects when required and to form shapes according to that is not possible by present robots. Robot creation is done with a single anatomy which is not at all flexible and dedicated to a single architecture only. Clone formation has never been done before by any robot either by big robots other robots. The algorithms for claytrons work are localization algorithm and shape sculpting algorithms are not well defined by previous processes. First of all we made two algorithms which we derived from the definition of basic claytronics algorithm that were localization and shape sculpting algorithms. The first algorithm we made is Centre point algorithm for forming shapes from a centre point. This comes under shape sculpting algorithm. The second algorithm is master-slave algorithm for localization of catoms and to arrange them in proper order. For allocating resources we are using Master-Slave Algorithm for scheduling and allocating resources to catoms according to their priorities.

**CONCLUSION:**

It is more effective than any other technology because it supports shape formation of any matter and clone formation also sensors for sensing which had never been use before. It is efficient and power saving as compared to other robots it has batteries charged by techniques of inductive charging. In future advanced version of claytronics can be used in creating claytronic clones of astronaut to send them in to space and whatever astronauts do on earth the clone performs the same in the space and research becomes easier. So the conclusion that can be drawn is that our project is cheap , durable and efficient.

**Future scope:**

It is more effective than any other technology because it supports shape formation of any matter and clone formation also sensors for sensing which had never been use before. It is efficient and power saving as compared to other robots it has batteries charged by techniques of inductive charging and last up to long time and no need of wires is required. In future advanced version of claytronics can be used in creating claytronic clones of astronaut to send them in to space and whatever astronauts do on earth the clone performs the same in the space and research becomes easier. So the conclusion that can be drawn is that our project is cheap , durable and efficient.

**References:**

1. Books: Artificial learning by Peter Norvig, Embedded C The Complete Reference .

2. Sites :www.google.com,www.Youtube/Learn Proteus tutorials.