A

Mini Project Report

On

Tactile Keyboard using Micro-controller

Naman Kalkhuria (IEC2010065)

Prathak Rastogi (IEC2010093)

Prashant Mishra (IEC2010096)

B.Tech. - 7th Semester (E.C.)

Under the Guidance of:

Mr. Ashutosh Kumar Singh

Assistant Professor

I.I.I.T.-Allahabad

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this project entitled "**Tactile Keyboard using Micro-controller**", submitted in the partial fulfillment of the completion of the semester VIIth of Bachelor of Technology (B.Tech.) program, in Electronics and Communication Engineering at Indian Institute of Information Technology, Allahabad, is an authentic record of my original work carried out under the guidance of **Mr. Ashutosh Kumar Singh** due acknowledgements have been made in the text of the project to all other material used. This semester work was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place: Allahabad
Date:

Naman Kalkhuria - IEC2010065
Prathak Rastogi – IEC2010093
Prashant Mishra – IEC2010096

CERTIFICATE FROM SUPERVISOR

I/We do hereby recommend that the mini project report prepared under my/our supervision by "Naman Kalkhuria (IEC2010065), Prathak Rastogi (IEC2010093), Prashant Mishra (IEC2010096) " titled "**Tactile Keyboard using Micro-controller**" be **accepted** in the partial fulfillment of the requirements of the completion of VIIth semester of Bachelor of Technology in Electronics and Communication Engineering **for Examination.**

Date:
Place: Allahabad, IIITA

Mr. Ashutosh Kumar Singh

Committee for Evaluation of the Thesis

_____ _____

_____ _____

ACKNOWLEDGEMENTS

This project could not have been accomplished without **Mr. Ashutosh Kumar Singh** who not only served as our supervisor but also encouraged us throughout project. He guided us through the debugging process, never accepting less than our best efforts. We thank him for all his help.

Naman Kalkhuria (IEC2010065)

Prathak Rastogi (IEC2010093)

Place: Allahabad

Prashant Mishra (IEC2010096)

Date:

B.Tech. 3rd Year, IIITA

Indian Institute of Information Technology Allahabad

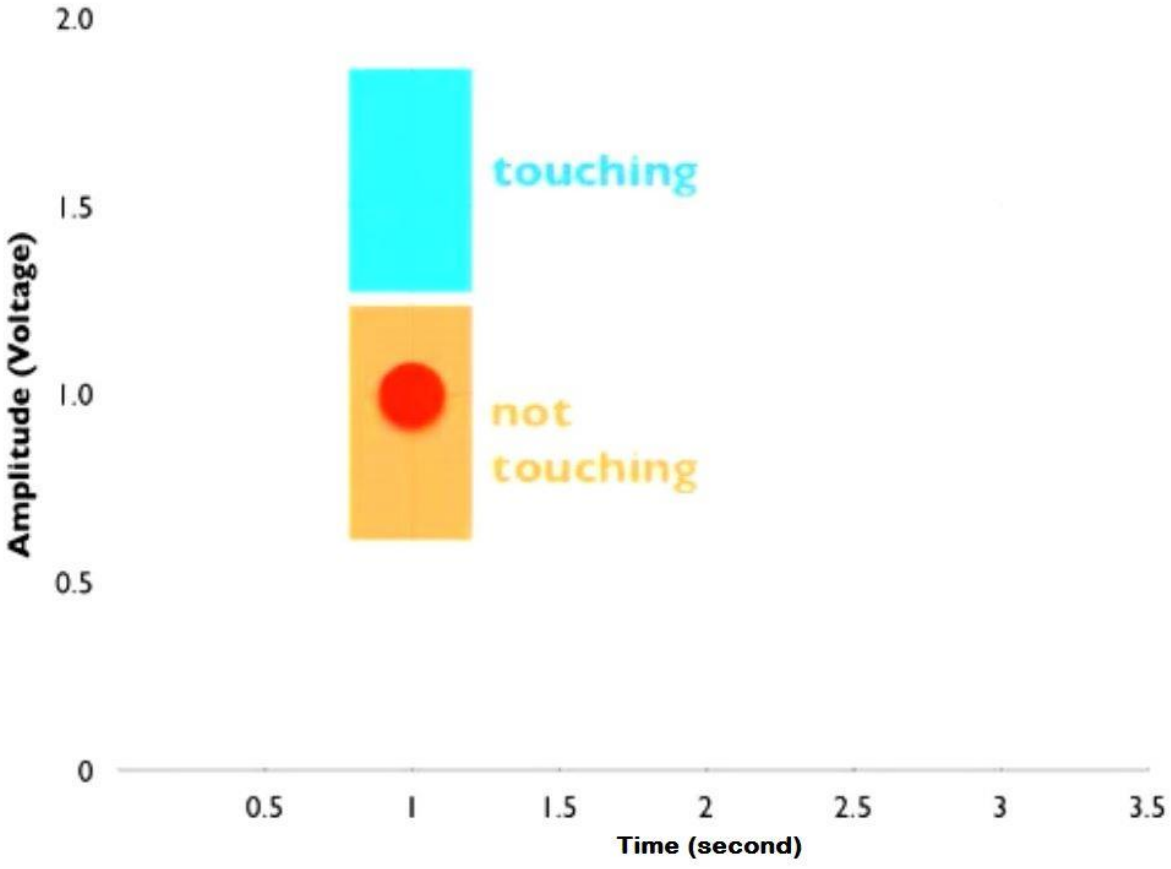
TABLE OF CONTENTS:

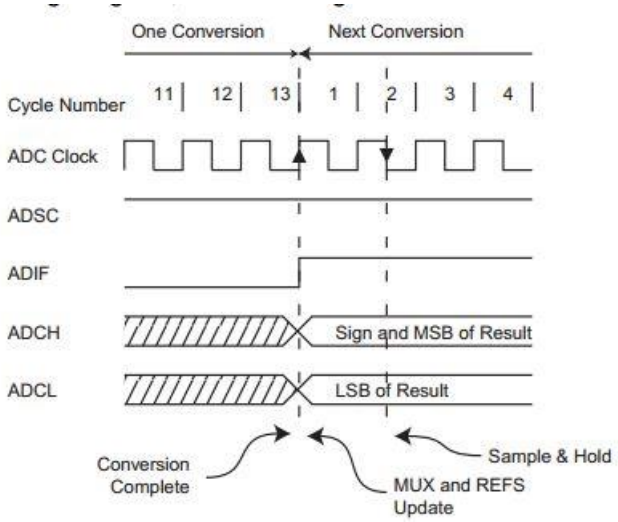
LITERATURE SURVEY

In a typical keyboard, we require a particular hardware which consist of keys making the keyboard a little bit tedious to carry on. In a keyboard we require 26 alphabetic key, 10 numeric keys with a total of 106 keys. We can make the keys using a simple day to day objects that could help in reducing the requirement of particular hardware for the keyboard.

In this we use the method of voltage drop across the resistor. We first connect the object to one end of the resistor of the circuit and then we carry the Ground wire in the hand. When touch the object the circuit gets completed thus providing the sufficient amount of voltage at the ADC pin to get detected.

This act as a key pressing. When the object is touched the circuit get completed thus providing the voltage to the ADC pin. Then this ADC channel convert the input to the digital value which is then compared to the threshold defined before.





By comparing this input voltage with the threshold we get the logic diagram as we required that is if the voltage is above threshold or not. If the voltage is above we call the instance of key pressing of the driver in the board which makes the computer to think that a key is being pressed thus making the touch to work as a keyboard key.

This method is known as *Voltage Drop Detection,* as in this method the main criteria is to detect the voltage change across the resistor and completion of the circuit.

Voltage Drop in Resistance:

Voltage drop describes how the supplied energy of a voltage source is reduced as electric current moves through the passive elements (elements that do not supply voltage) of an electrical circuit. Voltage drops across internal resistances of the source, across conductors, across contacts, and across connectors are undesired; supplied energy is lost (dissipated). Voltage drops across loads and across other active circuit elements are desired; supplied energy performs useful work. Recall that voltage represents energy per unit charge. For example, an electric space heater may have a resistance of ten ohms, and the wires which supply it may have a resistance of 0.2 ohms, about 2% of the total circuit resistance. This means that approximately 2% of the supplied voltage is lost in the wire itself. Excessive voltage drop may result in unsatisfactory operation of, and damage to, electrical and electronic equipment.

Consider a direct-current circuit with a nine-volt DC source; three resistors of 67 ohms, 100 ohms, and 470 ohms; and a light bulb—all connected in series. The DC source, the conductors (wires), the resistors, and the light bulb (the load) all have resistance; all use and dissipate supplied energy to some degree. Their physical characteristics determine how much energy. For

example, the DC resistance of a conductor depends upon the conductor's length, cross-sectional area, type of material, and temperature.

If you measure the voltage between the DC source and the first resistor (67 ohms), you will notice the voltage potential at the first resistor is slightly less than nine volts. The current passes through the conductor (wire) from the DC source to the first resistor; as this occurs, some of the supplied energy is "lost" (unavailable to the load), due to the resistance of the conductor. Voltage drop exists in both the supply and return wires of a circuit. If you measure the voltage across each resistor, you will measure a significant number. That represents the energy used by the resistor. The larger the resistor, the more energy used by that resistor, and the bigger the voltage drop across that resistor.

You can use Ohm's Law to verify voltage drop. In a DC circuit, voltage equals current multiplied by resistance. V = I R. Also, Kirchhoff's circuit laws state that in any DC circuit, the sum of the voltage drops across each component of the circuit is equal to the supply voltage.

Resistance:

The electrical resistance of an electrical conductor is the opposition to the passage of an electric current through that conductor; the inverse quantity is electrical conductance, the ease at which an electric current passes. Electrical resistance shares some conceptual parallels with the mechanical notion of friction. The SI unit of electrical resistance is the ohm (Ω), while electrical conductance is measured in siemens (S).

An object of uniform cross section has a resistance proportional to its resistivity and length and inversely proportional to its cross-sectional area. All materials show some resistance, except for superconductors, which have a resistance of zero.

Keyboard.h library of Arduino:

Sends a keystroke to a connected computer. This is similar to pressing and releasing a key on your keyboard. You can send some ASCII characters or the additional keyboard modifiers and special keys.

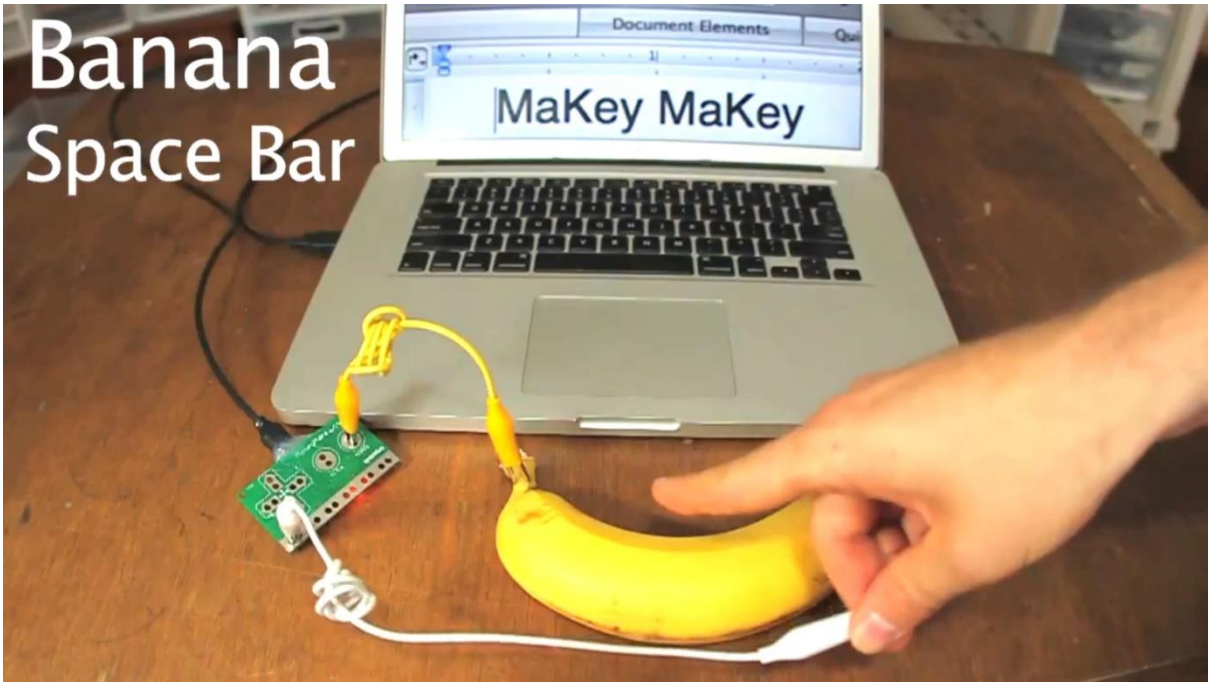
Only ASCII characters that are on the keyboard are supported. For example, ASCII 8 (backspace) would work, but ASCII 25 (Substitution) would not. When sending capital letters, Keyboard.write() sends a shift command plus the desired character, just as if typing on a keyboard. If sending a numeric type, it sends it as an ASCII character (ex. Keyboard.write(97) will send 'a').

The keyboard functions enable a Leonardo, Micro, or Due to send keystrokes to an attached computer.

Note: Not every possible ASCII character, particularly the non-printing ones, can be sent with the Keyboard library. The library supports the use of modifier keys. Modifier keys change the behavior of another key when pressed simultaneously.

MOTIVATION

In our day-to-day task we come in contact with various types of machines and we always yearn that these machines work according to our will without even telling them to do so. So to make this happen we need a type of interaction with these types of machines.



The most common type of interaction we make with objects or machines is touch. So if we can easily make an object sensitive to our touch and make them understand the way we touch; this could help in automating various tasks. By this thought this project came into our mind.

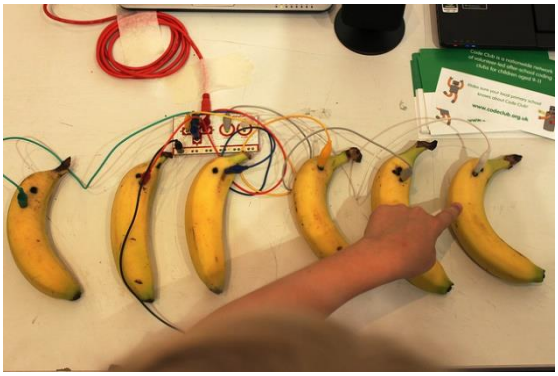
Our project can be implemented in following:

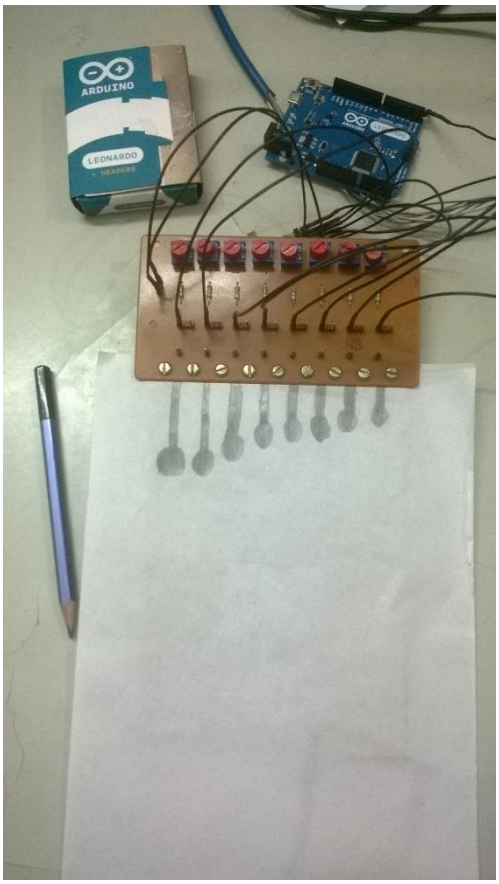
- For making day to day objects as keys of keyboard.
- For implementing keyboard where could not really expect them to be present.
- For detecting the touch of the finger on the surface of the conducting object.

PROBLEM FORMULATION

Generally the keyboard is made by use physical keys, which require lot of space make the use of conventional keyboards in day to day objects redundant.

By using the *Voltage Drop Detection*, we can easily convert the object in the keys of the keyboard.





PROPOSED SOLUTION

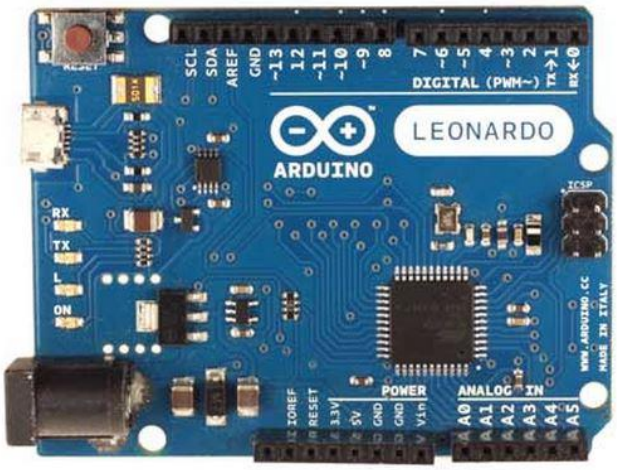
Methodology:

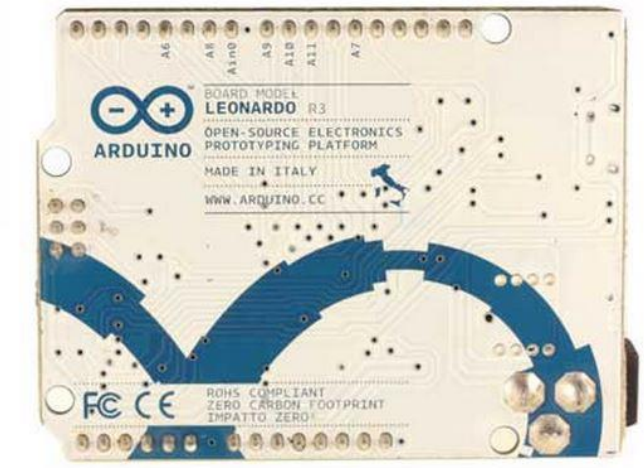
For implementing this technique of *Voltage Drop Detection*, we are taking the analog input of the sensor into the Arduino board and analyzing it using the processing language in the computer.

Various hardware and software components to be use:

1. **MICRO-CONTROLLER BOARD:-** ARDUINO Leonardo

   The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

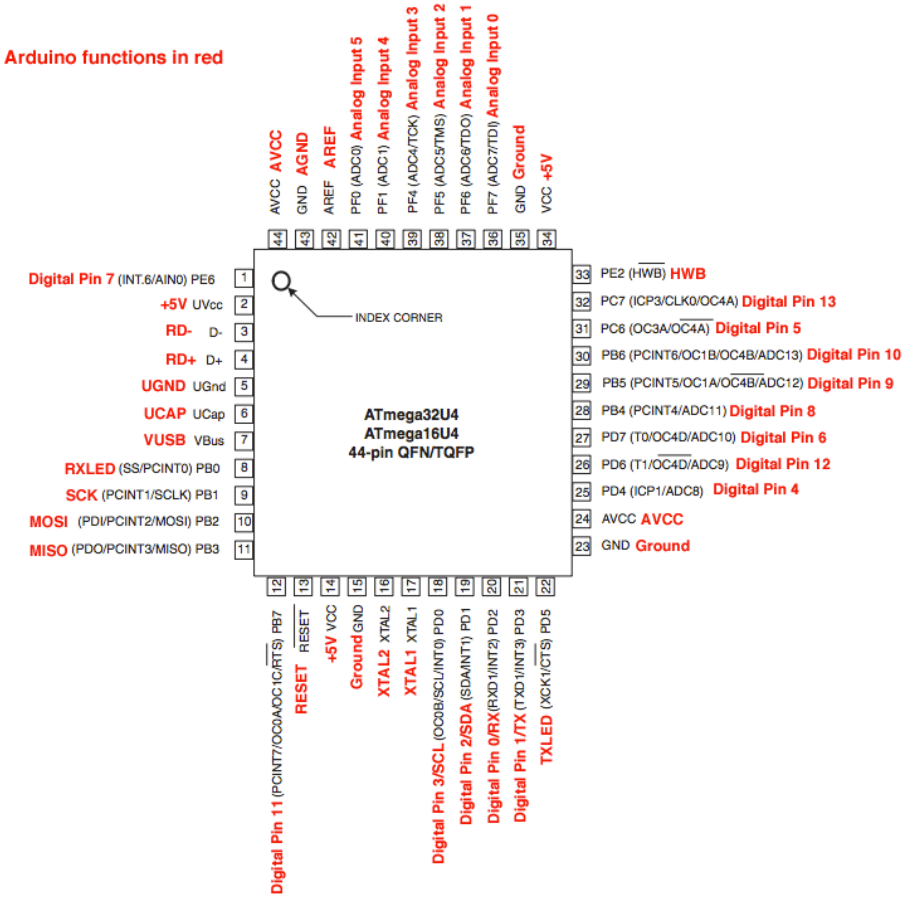




Summary

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	20

PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz



Programming

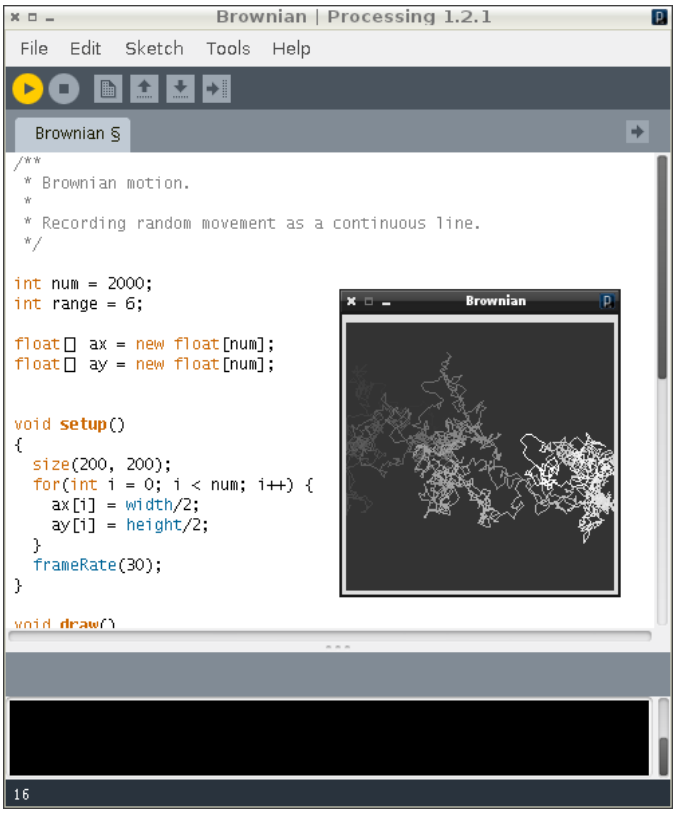
The Arduino Uno can be programmed with the Arduino software. The ATmega328 on the Arduino Uno comes pre-burned with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

2. SOFTWARE USED:

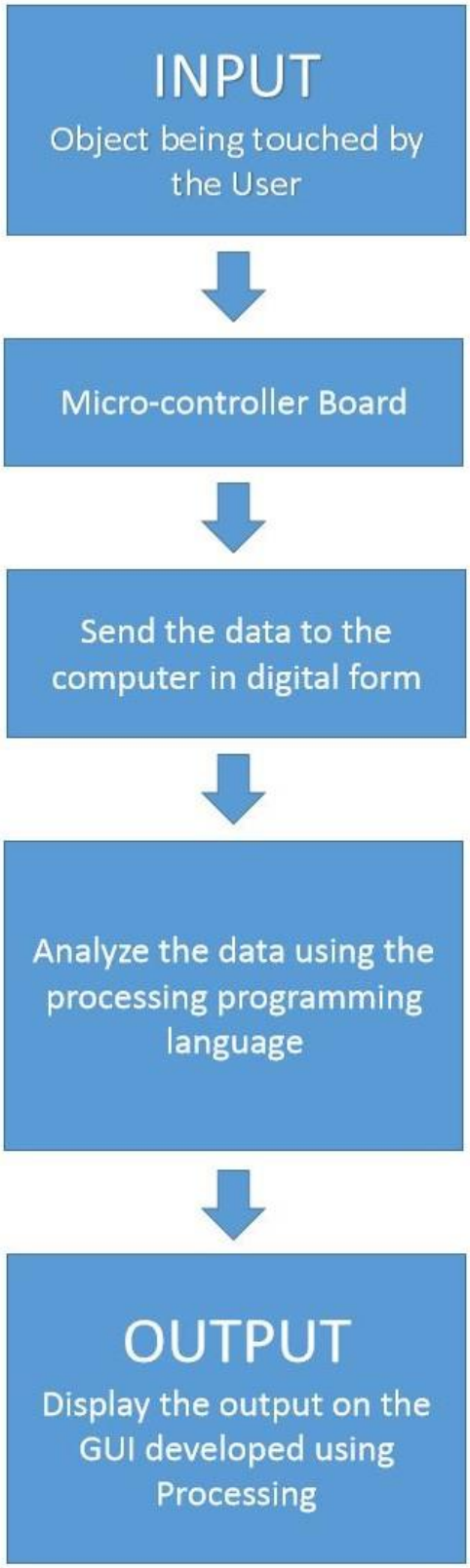
Processing Programming Language:

Processing includes a *sketchbook*, a minimal alternative to an integrated development environment (IDE) for organizing projects.

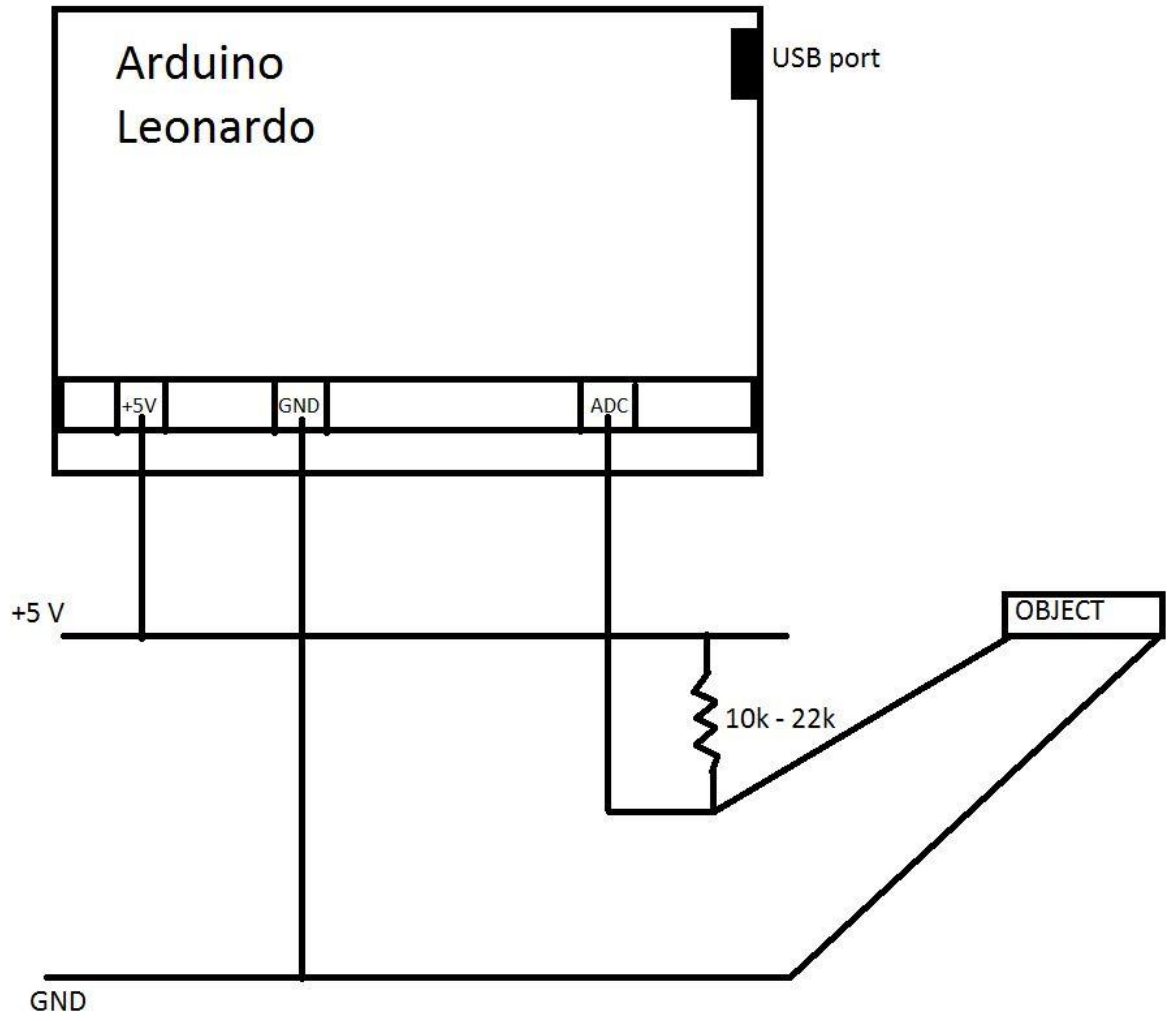
Processing also allows for users to create their own classes within the PApplet sketch. This allows for complex data types that can include any number of arguments and avoids the limitations of solely using standard data types such as: int (integer), char (character), float (real number), and color (RGB, ARGB, hex).



Block Diagram:



CIRCUIT DIAGRAM:



DETECTING THE BOARD AS A KEYBOARD BY COMPUTER:-

For getting the input from the key we have to make the board work as a keyboard which require drivers of the keyboard. Leonardo is with the capability of making the board work as a keyboard as well as mouse.

By taking the input from touching the object make the circuit complete thus making the ADC to detect some amount of voltage at its pin. This small is then compared with the threshold value of the key we have pre-determined by various experimentation. This touch is then converted to detection of the key pressing by the driver of the board.

CODE

```
boolean check0 = false;
boolean check1 = false;
boolean check2 = false;
boolean check3 = false;
boolean check4 = false;
boolean check5 = false;
boolean check6 = false;
boolean check7 = false;


void setup()
{
  Serial.begin(115200);
}


void loop()
{
  float input0 =  analogRead(A0);
  float input1 = analogRead(A1);
  float input2 =  analogRead(A2);
  float input3 = analogRead(A3);
  float input4 =  analogRead(A4);
  float input5 = analogRead(A5);
  float input6 =  analogRead(A6);
  float input7 = analogRead(A7);

  if (input0<200)
```

```
{
  if (!check0)
  {
    Keyboard.write('A');
    //Serial.println(input0);
    check0 = !check0;
  }
}

if (input0>600)
{
  if (check0)
  {
    check0 = !check0;
  }
}
//delay(15);

if (input1<200)
{
  if (!check1)
  {
    Keyboard.write('S');
    //Serial.println(input1);
    check1 = !check1;
  }
}

if (input1>600)
```

```
{
  if (check1)
  {
    check1 = !check1;
  }
}
//delay(15);

if (input2<200)
{
  if (!check2)
  {
    Keyboard.write('D');
    //Serial.println(input2);
    check2 = !check2;
  }
}

if (input2>600)
{
  if (check2)
  {
    check2 = !check2;
  }
}
//delay(15);

if (input3<200)
{
```

```
  if (!check3)
  {
    Keyboard.write('F');
    //Serial.println(input3);
    check3 = !check3;
  }
}

if (input3>600)
{
  if (check3)
  {
    check3 = !check3;
  }
}
//delay(15);

if (input4<200)
{
  if (!check4)
  {
    Keyboard.write('G');
    //Serial.println(input4);
    check4 = !check4;
  }
}

if (input4>600)
{
```

```
  if (check4)
  {
    check4 = !check4;
  }
}
//delay(15);

if (input5<200)
{
  if (!check5)
  {
    Keyboard.write('H');
    //Serial.println(input5);
    check5 = !check5;
  }
}

if (input5>600)
{
  if (check5)
  {
    check5 = !check5;
  }
}
//delay(15);

if (input6<200)
{
  if (!check6)
```

```
  {
    Keyboard.write('J');
    //Serial.println(input6);
    check6 = !check6;
  }
}

if (input6>600)
{
  if (check6)
  {
    check6 = !check6;
  }
}
//delay(15);

if (input7<200)
{
  if (!check7)
  {
    Keyboard.write('K');
    //Serial.println(input7);
    check7 = !check7;
  }
}

if (input7>600)
{
  if (check7)
```

```
    {
      check7 = !check7;
    }
  }
  //delay(15);

}
```

Details from the Reference Page:

MaKey MaKey is a printed circuit board with an ATMega32u4 microcontroller running Arduino Leonardo firmware. It uses the Human Interface Device (HID) protocol to communicate with your computer, and it can send keypresses, mouse clicks, and mouse movements. For sensing closed switches on the digital input pins, we use high resistance switching to make it so you can close a switch even through materials like your skin, leaves, and play-doh. We use a pull-up resistor of 22 mega ohms. This technique attracts noise on the input, so we use a moving window averager to lowpass the noise in software, saving money on hardware filtering. There are six inputs on the front of the board, which can be attached to via alligator clipping, soldering to the pads, or any other method you can think of. There are another 12 inputs on the back, 6 for keyboard keys, and 6 for mouse motion, which you can access with jumpers via the female headers, paper clips, or by alligator clipping creatively around the headers. If you wish to use a different set of keys, or otherwise change the behavior of your MaKey MaKey, you can simply reprogram it using the Arduino environment. By cutting a trace on the back of the board, you can disconnect the large pull-up resistors if you want to, which would be necessary in a small minority of Arduino projects.

References:

- http://blog.arduino.cc/2013/03/23/how-to-make-makey-makey/
- http://www.makeymakey.com/