MOTION SENSOR, PROXIMITY SENSOR AND TOUCH PAD USING ARDUINO

CLUSTER INNOVATION CENTRE
UNIVERSITY OF DELHI

B.TECH IV.4.1

LOGIC AND FUNCTIONS THROUGH DIGITAL ELECTRONICS

GUIDE

Dr. Sunita Negi, Assistant Professor

SUBMITTED BY

Sumit Yadav (521)
Rahul Yadav (1387)

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our teacher Dr. Sunita Negi who gave the golden opportunity to do this wonderful project in electronics on motion sensor, proximity sensor and touch pad using Arduino which also helped us in doing a lot of research and we came to know about so many new things. Special thanks to Dr. Pankaj Tyagi who helped us with the physics of Proximity Sensor. We would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

INDEX

INTRODUCTION

Electronics deals with electrical circuits that involve active electrical components such as vacuum tubes, transistors, diodes and integrated circuits, and associated passive interconnection technologies. The nonlinear behavior of active components and their ability to control electron flows makes amplification of weak signals possible and electronics is widely used in information processing, telecommunications, and signal processing. The ability of electronic devices to act as switches makes digital information processing possible. Interconnection technologies such as circuit boards, electronics packaging technology, and other varied forms of communication infrastructure complete circuit functionality and transform the mixed components into a regular working system.

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP). In this electronics project, we have used Arduino platform to design a Motion Sensor, Proximity Sensor and Touch Pad.

MOTION SENSOR USING ARDUINO

A motion detector is a device that detects moving objects, particularly people. A motion detector is often integrated as a component of a system that automatically performs a task or alerts a user of motion in an area. Motion detectors form a vital component of security, automated lighting control, home control, energy efficiency, and other useful systems.

In this project we have used proximity sensor values (A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact) detected with arduino to design our motion sensor.

Three proximity sensors are connected in a horizontal row. When there is any kind of motion along this horizontal row, the arduino reads the values of the three sensors. These values are taken as input values for our code (See Appendix 1). The program written in arduino detects if there is any motion depending upon the change in values in these proximity sensors. The program also tells us the direction of the motion.

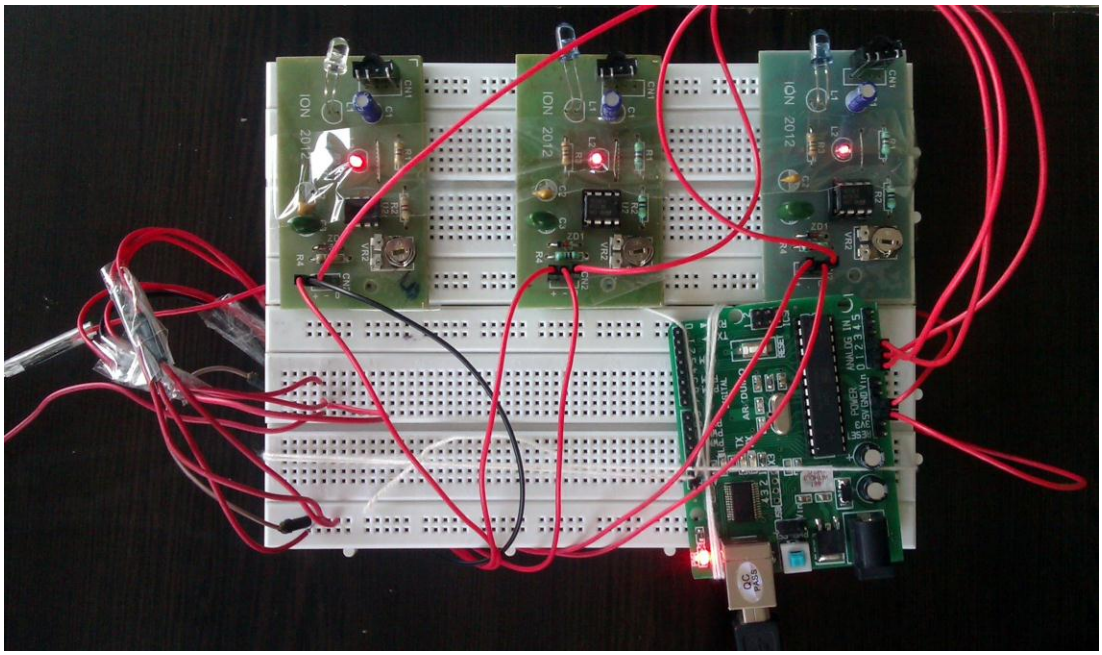

Image 1: Motion Sensor

PROXIMITY SENSOR USING ARDUINO

A proximity sensor is a sensor capable to detect the presence of nearby objects without any physical contact. A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the field or return signal. The object being sensed is often referred to as the proximity sensor's target. The maximum distance that this sensor can detect is defined "nominal range". Some sensors have adjustments of the nominal range or means to report a graduated detection distance.

Here we have designed a capacitive proximity sensor using arduino. In electrical engineering, capacitive sensing is a technology based on capacitive coupling that takes human body capacitance as input. Capacitive sensors detect anything that is conductive or has a dielectric different from that of air. Human interface devices based on capacitive sensing, such as trackpads, can replace the computer mouse. Digital audio players, mobile phones, and tablet computers use capacitive sensing touchscreens as input devices.

We have used the technology called surface capacitance. Here, one side of the insulator is coated with conductive material. A small voltage is applied to this layer, resulting in a uniform electrostatic field. When a conductor, such as a human finger, touches the uncoated surface, a capacitor is dynamically formed. Because of the sheet resistance of the surface, each corner is measured to have a different effective capacitance. The sensor's controller can determine the location of the touch indirectly from the change in the capacitance as measured from the four corners of the panel: the larger the change in capacitance, the closer the touch is to that corner. With no moving parts, it is moderately

durable, but has low resolution, is prone to false signals from parasitic capacitive coupling, and needs calibration when we use it.

Having these concepts in mind, we designed a proximity sensor based on this technology. We have used arduino to detect the different values as input as we go nearer or further away from the metal plate. Our code (see appendix 2) detects the different values and process them to detect of there is any object in front of it or not.

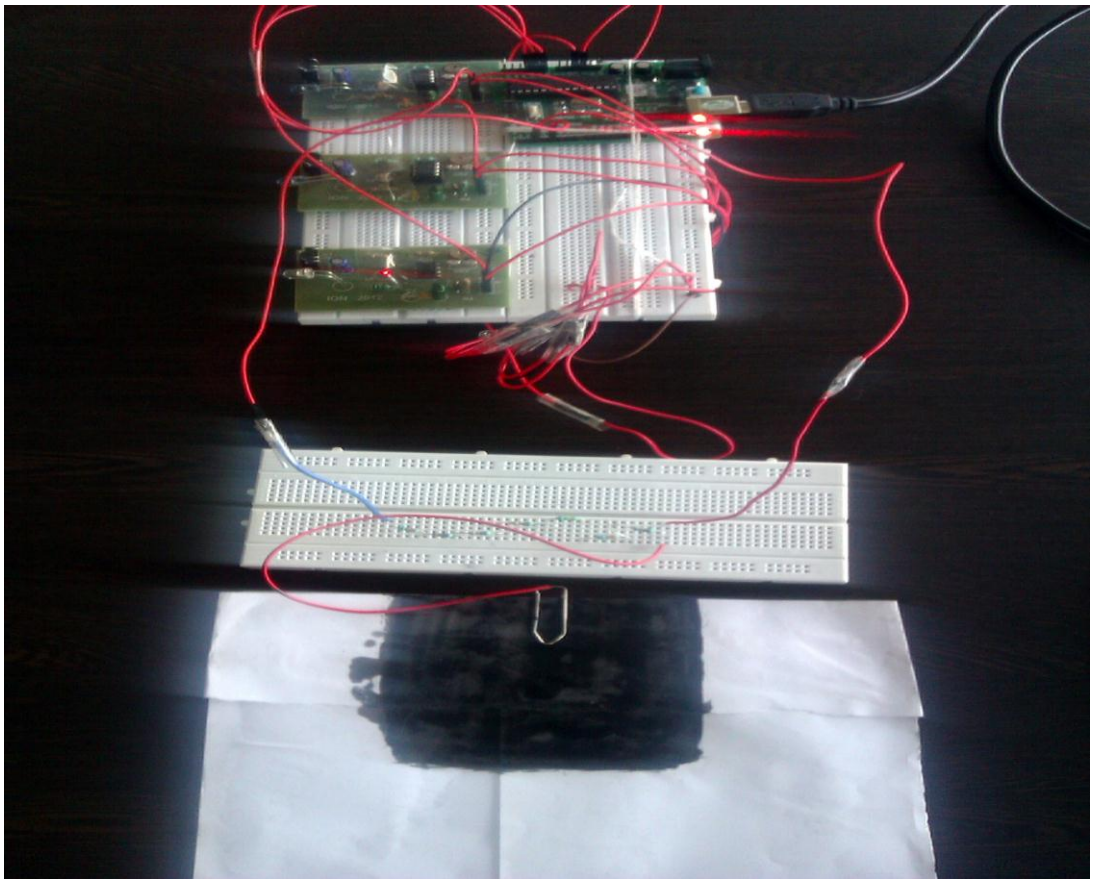

Image 2: Proximity Sensor

TOUCHPAD USING ARDUINO

A touchpad is a pointing device featuring a tactile sensor (sensitive to touch), a specialized surface that can translate the motion and position of a user's fingers to a relative position on screen. Touchpads are a common feature of laptop computers, and are also used as a substitute for a mouse where desk space is scarce. Touchpads operate in one of several ways, including capacitive sensing and conductance sensing. The most common technology entails sensing the capacitive virtual ground effect of a finger, or the capacitance between sensors.

In this activity, we have used a metal plate as one conducting material. The arduino code (see appendix 3) takes as input the different values when we touch the plate at different places. Higher the capacitance, the nearer is our object which is touching the plate.

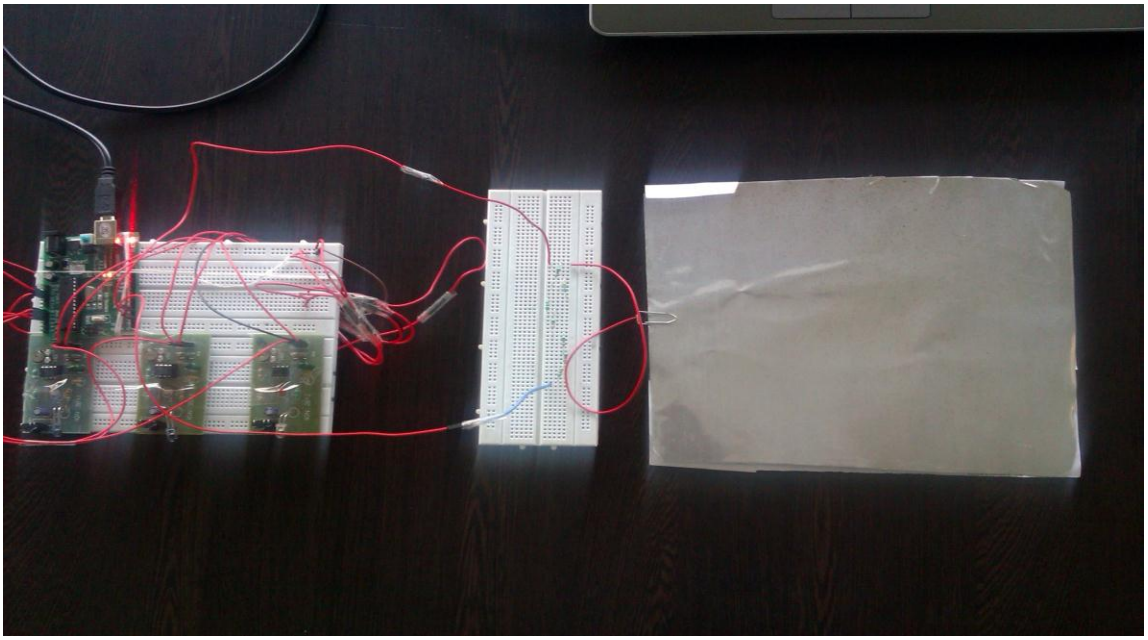

Image 3: Touchpad

CONCLUSIONS

- Motion Sensor, Proximity Sensor and Touchpad can be designed using arduino.
- A lot of physics is used in designing any kind of circuit design.
- Many types of sensors uses capacitive sensing including sensors to detect and measure proximity, position or displacement, fluid level, acceleration etc.
- We need to calibrate the sensors carefully to get the correct desired results.

REFERENCES

- http://www.arduino.cc/
- http://en.wikipedia.org/wiki/Motion_detector
- https://en.wikipedia.org/wiki/Capacitive_sensing
- https://en.wikipedia.org/wiki/Capacitive_displacement_sensor
- http://en.wikipedia.org/wiki/Touchpad
- http://en.wikipedia.org/wiki/Tactile_sensor
- http://vimeo.com/29831907#

APPENDIX 1

```
//CODE FOR MOTION SENSOR
int pin1=0;
int pin2=1;
int pin3=2;
int led=13;
int a1=0;   // first
int a2=0;   // second
int a3=0;   // third analog pin of arduino
int b1=0;   //previous value of first pin
int b2=0;   //previous value of second pin
int b3=0;   //previous value of third pin
int c1=0;   // difference of old and new value of first pin
int c2=0;   // difference of old and new value of second pin
int c3=0;   // difference of old and new value of third pin
void setup()
{
  pinMode(pin1,INPUT);
  pinMode(pin2,INPUT);
  pinMode(pin3,INPUT);
  pinMode(led,OUTPUT);
  Serial.begin(9600);
  b1=1024;
  b2=1024;
  b3=1024;
}

void loop()
{
  a1=analogRead(pin1);
  a2=analogRead(pin2);
  a3=analogRead(pin3);

  if(abs(b1-a1)>800 & abs(b2-a2)>800 & abs(b3-a3)>800)
  {
    Serial.println("Somebody is coming from the front..");
    //delay(500);
  }
  else if((c1>700 & abs(b2-a2)>800) || (c2>700 & abs(b3-a3)>800))
  {
    Serial.println("MOTION has been detected ! MOTION has been
detected !");
    Serial.println("Direction: Forward Motion");
   //delay(500);
  }
  else if((c3>700 & abs(b2-a2)>800) || (c2>700 & abs(b1-a1)>800))
  {
    Serial.println("MOTION has been detected ! MOTION has been
detected !");
    Serial.println("Direction: Reverse Motion");
```

```
 //delay(500);
  }
  else if(abs(b1-a1)>800 || abs(b2-a2)>800 || abs(b3-a3)>800)
  {
    Serial.println("MOTION has been detected ! MOTION has been
detected !");
    Serial.println("Direction: UnKnown...");
  //delay(100);
  }
   delay(500);
   c1=abs(a1-b1);
   c2=abs(a2-b2);
   c3=abs(a3-b3);
   b1=a1;
   b2=a2;
   b3=a3;

 /* Serial.println("value of pin1 is ");
  Serial.println(a1);
   Serial.println("value of pin2 is ");
  Serial.println(a2);
   Serial.println("value of pin3 is ");
  Serial.println(a3);*/
  //Serial.println("value of pin1 is "+a1);
  //Serial.println("value of pin1 is "+a1);
  //delay(5000);
  /*if( abs(a2-a1) > 800 )
  {
    Serial.println("motion has been detected ! motion has been
detected !");
    digitalWrite(led,HIGH);
    delay(100);
  }
  else
  {
    delay(500);
    Serial.println("Sensing Motion");

    digitalWrite(led,LOW);
  }
  a2=a1;
  */
}
```

APPENDIX 2

```
//CODE FOR CAPACITIVE PROXIMITY SENSOR
#include <CapacitiveSensor.h>

CapacitiveSensor   cs_4_2 = CapacitiveSensor(4,2);        // 10M
resistor between pins 4 & 2, pin 2 is sensor pin, add a wire and or
foil if desired
int temp=0;
void setup()
{
   cs_4_2.set_CS_AutocaL_Millis(0xFFFFFFFF);     // turn off
autocalibrate on channel 1 - just as an example
   Serial.begin(9600);
}

void loop()
{
    long start = millis();
    long total =  cs_4_2.capacitiveSensor(30);
    Serial.println(total);                  // print sensor output 1
    Serial.print("\t");
    if(abs(temp-total)>50)
    {
      Serial.println("object has been detected..");
    digitalWrite(13,HIGH);
    delay(200);    // LED Will Blink For 0.2 seconds
    digitalWrite(13,LOW);
    }
    else
  {
    Serial.println("trying to detect the object");
  }
    temp=total;
    delay(500);                             // arbitrary delay to
limit data to serial port
}
```

APPENDIX 3

```
#include <CapacitiveSensor.h>

CapacitiveSensor   cs_4_2 = CapacitiveSensor(4,2);        // 10M
resistor between pins 4 & 2, pin 2 is sensor pin, add a wire and or
foil if desired
int temp=0;
void setup()
{
   cs_4_2.set_CS_AutocaL_Millis(0xFFFFFFFF);     // turn off
autocalibrate on channel 1 - just as an example
   Serial.begin(9600);
}

void loop()
{
    long start = millis();
    long total =  cs_4_2.capacitiveSensor(30);
    Serial.println(total);                  // print sensor output 1
    Serial.print("\t");
    if(abs(temp-total)>50)
    {
      Serial.println("touch has been detected");
      Serial.println("keep your hand away from the plate");
    digitalWrite(13,HIGH);
    delay(200);    // LED Will Blink For 0.2 seconds
    digitalWrite(13,LOW);
    }
    else
  {
    Serial.println("detecting the touch input");
  }
    temp=total;
    delay(500);                             // arbitrary delay to
limit data to serial port
}
```