EP-315 PROJECT REPORT

ROBOTIC HAND

GUIDANCE:

PROF.PRADEEP SARIN

DEPT. OF PHYSICS

IIT-BOMBAY.

BY

SATISH KOKATE (08026002)

RAHUL DABHI (08026011)

SRIKAR KANURI (08D26007)

❖ Description:

We have tried to make a robotic hand which can imitate the movement of the human hand by sensing its motion with the sensors attached to the real hand. The sensors that are going to be used are accelerometer to sense position in x-axis and y-axis and a potentiometer to sense movement of thumb with respect to the rest of the fingers. Main application of Arduino ATmega168 in this project is that it provides as an interface between analog inputs from accelerometer and potentiometer and required digital output, by doing ADC conversion. The corresponding PWM output that we get from arduino after feeding inputs from accelerometer and potentiometer is given to the three servos:

(1) One for **wrist roll** = movement of human hand in **Y-axis**.

(2) Another for **wrist pitch** = movement of human hand in **X-axis**.

(3) And remaining one for finger opening and closing = change in position of finger with respect to thumb.

Servos which are known as the ones difficult to control can easily be controlled and calibrated for the motion of hand with help of this microprocessor.

❖ REQUIRED COMPONENTS:

1. 3 Servo motors.
2. An accelerometer.
3. Potentiometer.
4. Arduino ATmega168 board

❖ SUB-BLOCKS ASSIGNED TO THREE GROUP MEMBER :

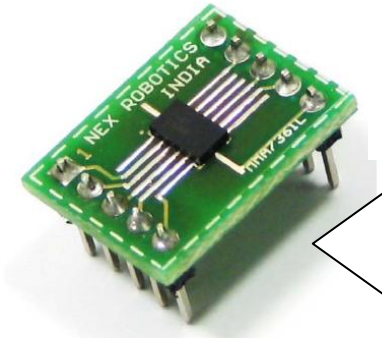
(1) Coding of motion sensing through accelerometer and producing corresponding outputs, and test it: **Rahul Dabhi, Satish Kokate**

(2) Coding of motion sensing through potentiometer and producing corresponding outputs, and test it: **Srikar Kanuri**

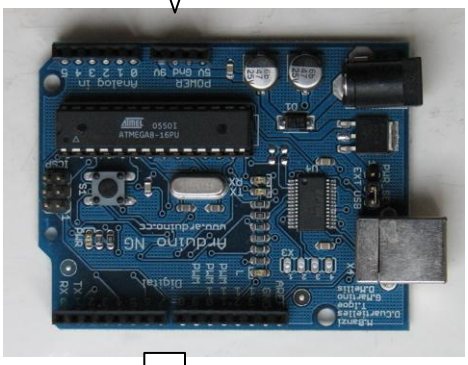
(3) Mechanical part[Giving outputs from microprocessor to servos and assembling servos so as to get net motion of hand]: **Satish Kokate, Rahul Dabhi, Srikar Kanuri**

We were inspired for doing this project from the *James Cameron*'s movie *'Avatar'*, in which he has used **AMP-suit** [***Amplified Mobility Platform suit***],which is one massive version of battle robot. Its driver can drive it by wearing motion sensing gloves and it is programmed to amplify that motion [hence the name AMP]. Our project is a smaller prototype of it i.e. it imitates motion of only one hand about the wrist and motion of index finger with respect to thumb. This project provides the newer way to control robots besides the traditional remote controls.

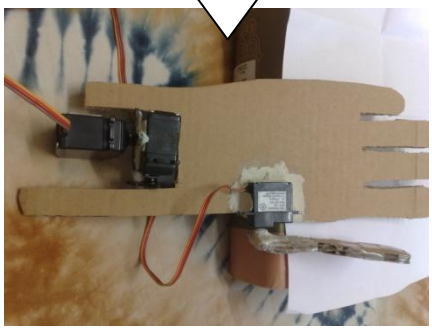
❖ BLOCK DIAGRAM:



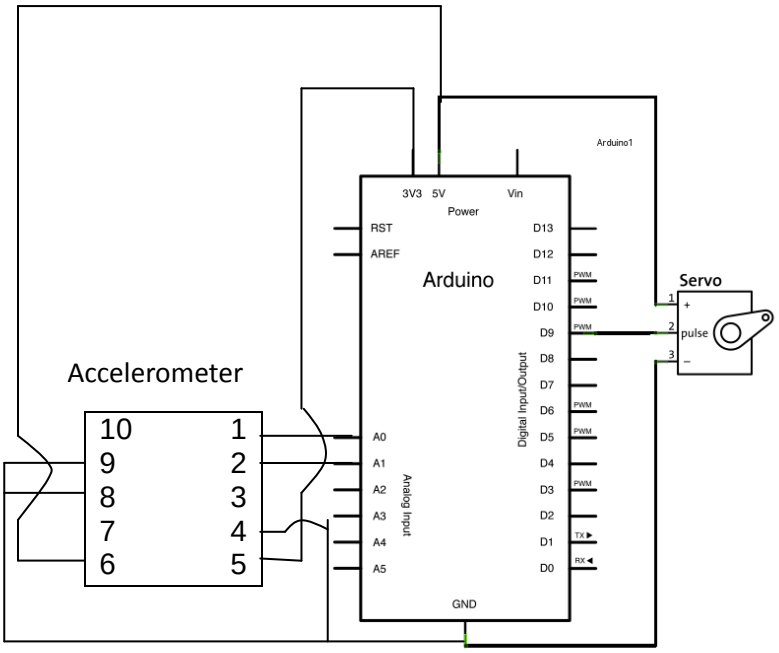
Analog pins of arduino take inputs from accelerometer and potentiometer.



PWM o/p from arduino is then fed to the combination of servo motors, which makes up the robotic hand.



❖ ACTUAL CIRCUIT DIAGRAM:



Accelerometer datasheet can be found here.

❖ ARDUINO CODE:

```
#include <Servo.h>                    // Includes the Servo Lib

Servo myservo1, myservo2, myservo3;
int val1, val2, val3;

void setup()
{
  Serial.begin(9600);                 // initialize the serial port at 9600bps

  myservo1.attach(9);                 // Attach the Servo variable to a pin
  myservo2.attach(10);
  myservo3.attach(11);
}

void loop()                           // loop over the sensors
{
    // read each sensor
    int sensorReading1 = analogRead(0); // accelerometer x-axis reading
    Serial.print(sensorReading1);
    Serial.print(",");                // if this isn't the last sensor to read,
                                      // then print a comma after it.

    int sensorReading2 = analogRead(1);  // accelerometer y-axis reading
    Serial.print(sensorReading2);
    Serial.print(",");

    int sensorReading3 = analogRead(3); // pot reading for fingers
    Serial.print(sensorReading3);

    Serial.println();

  val1= map(sensorReading2, 150, 520, 0, 179);
  myservo1.write(val1);
  val2= map(sensorReading1, 150, 5200, 0, 179);
  myservo2.write(val2);
  val3= map(sensorReading3, 0, 1023, 0, 179);
  myservo3.write(val3);
 }
```

❖ Commands used:

(1) map(value, fromLow, fromHigh, toLow, toHigh)

Re-maps a number from one range to another. That is, a **value** of **fromLow** would get mapped to **toLow**, a value of **fromHigh** to **toHigh**, values in-between to values in-between, etc. The map() function uses integer math so will not generate fractions, when the math might indicate that it should do so. Fractional remainders are truncated, and are not rounded or averaged.

Parameters

value: the number to map

fromLow: the lower bound of the value's current range

fromHigh: the upper bound of the value's current range

toLow: the lower bound of the value's target range

toHigh: the upper bound of the value's target range

Returns

The mapped value.

(2) *servo*.write(angle)

Description

Writes a value to the servo, controlling the shaft accordingly. On a standard servo, this will set the angle of the shaft (in degrees), moving the shaft to that orientation. On a continuous rotation servo, this will set the speed of the servo (with 0 being full-speed in one direction, 180 being full speed in the other, and a value near 90 being no movement).

Parameters

servo: a variable of type Servo

angle: the value to write to the servo, from 0 to 180