ROBOTICS ARM

by srimant Kumar nahak

Robotics arm is a mechanical based combination of different types of joints to perform a special kind of task, either it is controlled manually or automatically using programmable microcontrollers.

Degrees of freedom, in a mechanics context, are specific, defined modes in which a mechanical device or system can move. The number of degrees of freedom is equal to the total number of independent displacements or aspects of motion. A machine may operate in two or three dimensions but have more than three degrees of freedom. The term is widely used to define the motion capabilities of robots.

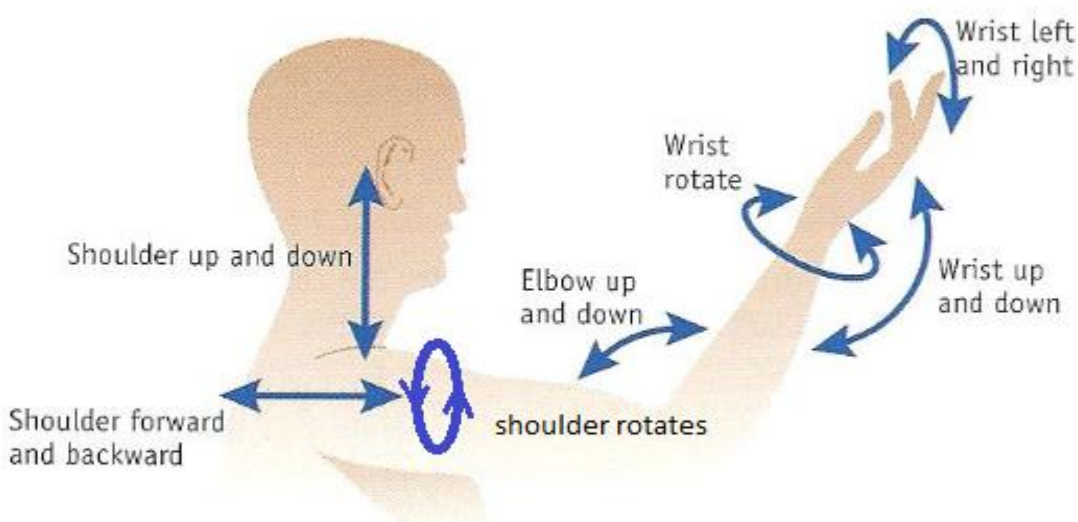
Human should basically have 7 degrees of freedom. Shoulder motion can take place as

-pitch (up and down) or yaw (left and right).

-Elbow motion can occur only as pitch.

-Wrist motion can occur as pitch or yaw.

- Rotation (roll) may also be possible for wrist and shoulder.



Benefits and uses of robotics arm

Robotics arms can be called as a proper substitute of large manpower required for industrial applications with maximum accuracy.

The three methods of movement used in these robots are:-

-pick and place: it is mainly used when we are loading or unloading materials

-point to point: it can be used while shouldering or welding and this is where the robots needs to be programmed to go in specific locations one after the other in a cycle.

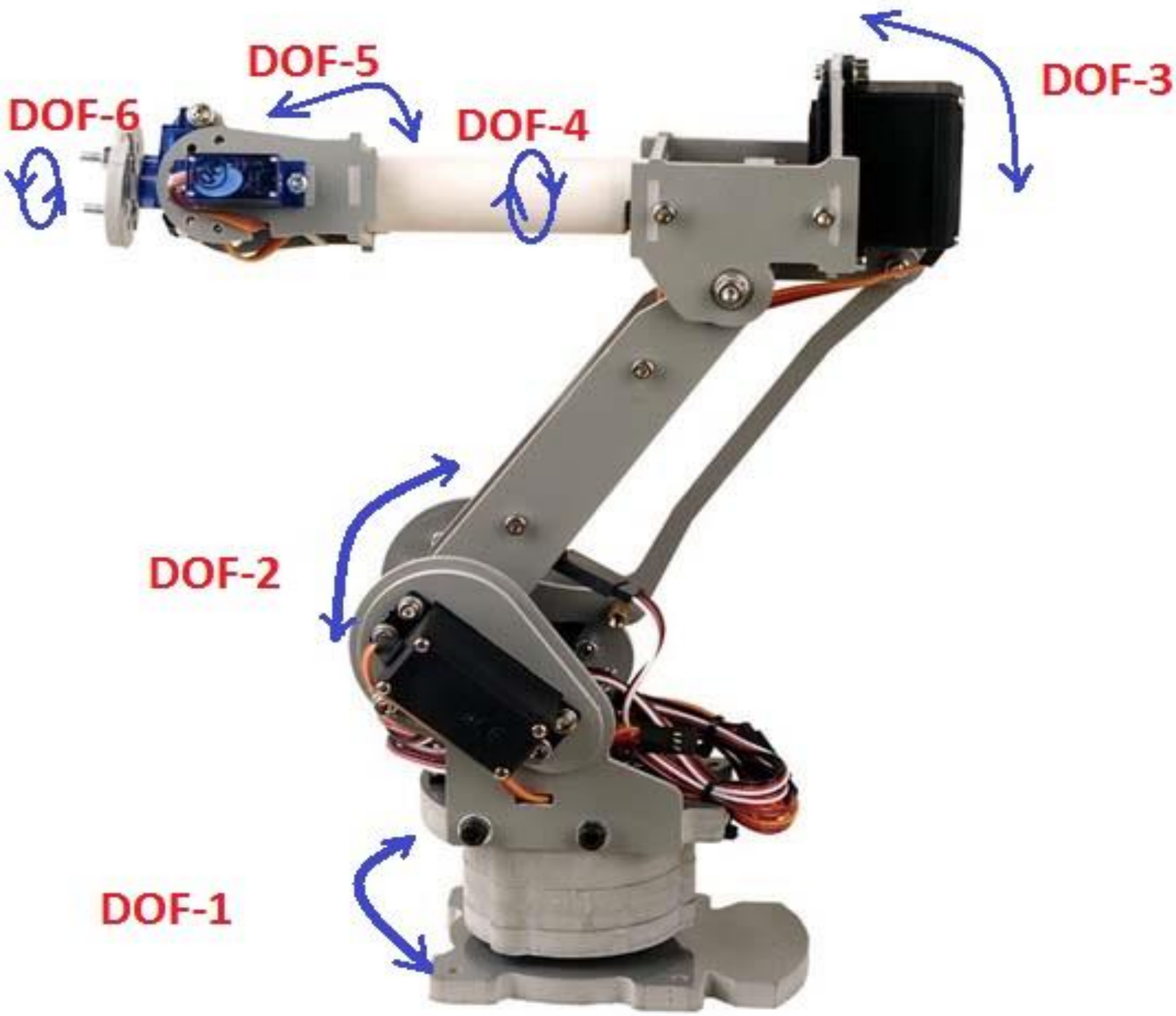
-contouring: when the robot has to be move in a fluid curve like motion and this can be used in painting purposes.

Different methods of making a robotic arm

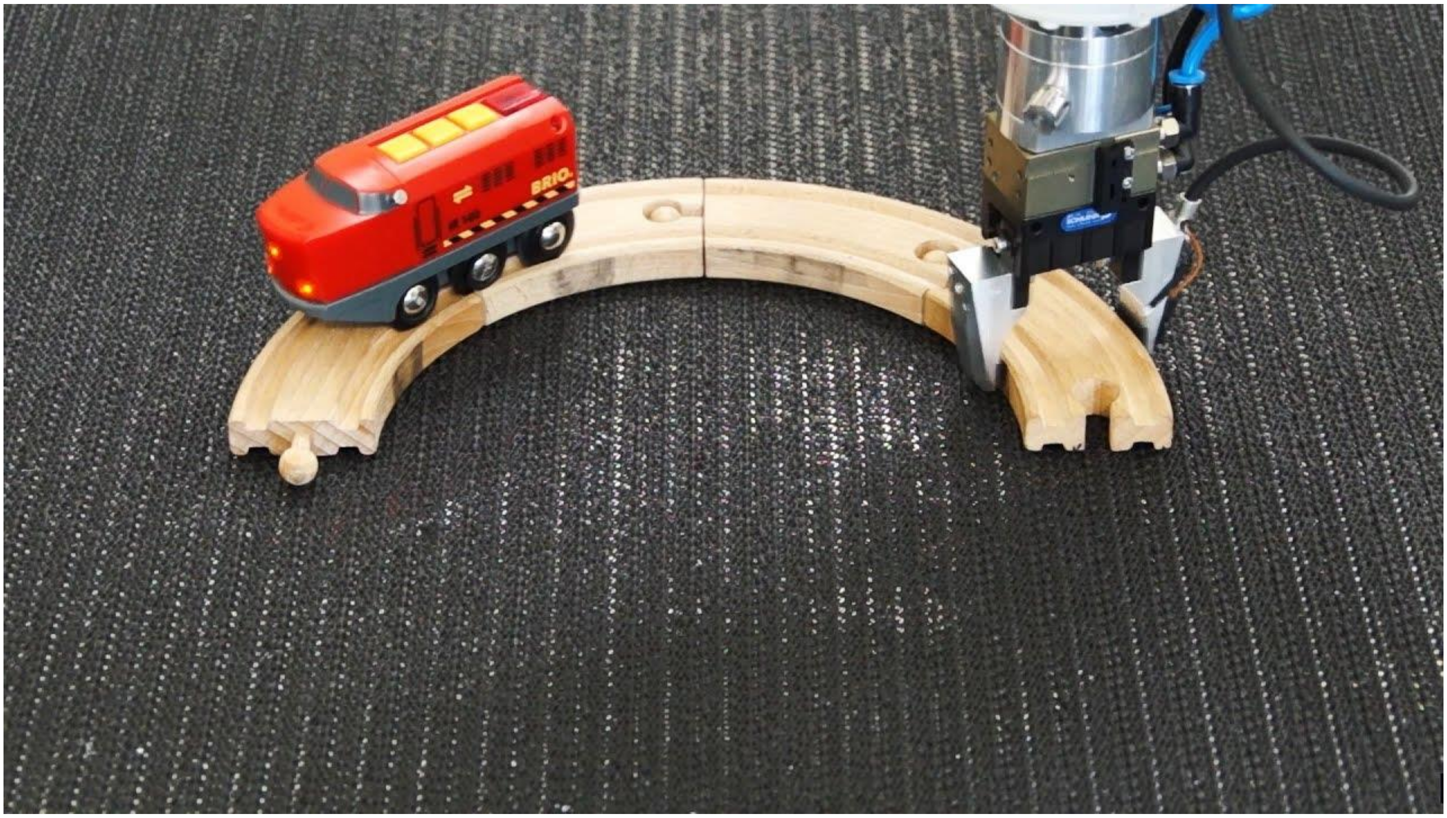
1.To connect different joint of a robotic arm either we can use **servo motors** or we can use **stepper motor** (**with metallic gears**). The basic difference between a traditional stepper and a servo-based system is the type of motor and how it is controlled. Steppers don't require encoders since they can accurately move between their many poles whereas servos, with few poles, require an encoder to keep track of their position. Moreover, stepper motors provide much more servo motor. But Still we use servos due to its low price.

2. There are many types of motors are available in today's market, but mostly Tiny pager motors, servo motors, linear motors, stepper motors and DC geared motors are used in industrial robots according to their application area.

Structure of robotic arm



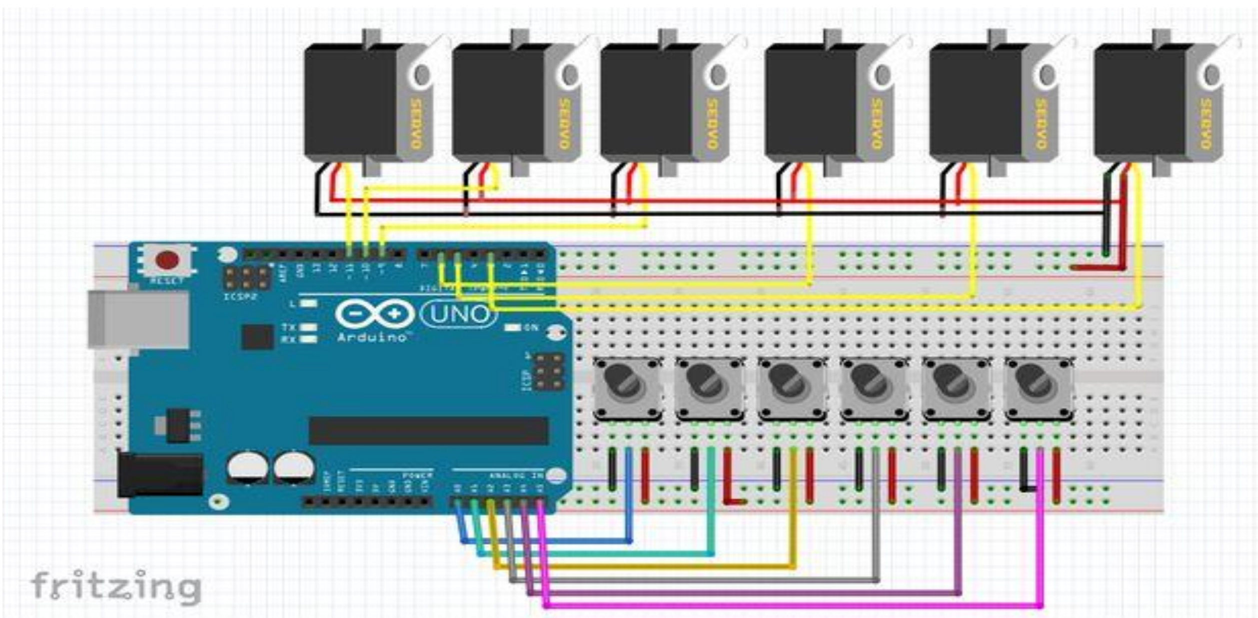
NOTE: We can also add one more motion of robot i.e., at base we add a circular rail.

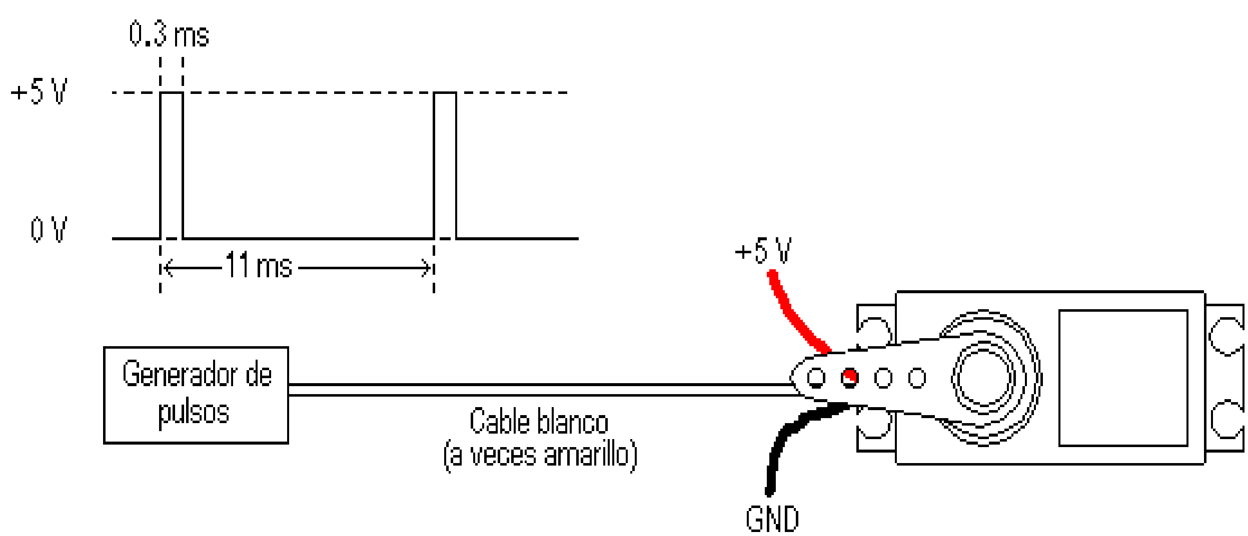


Moving on to the most important topic i.e. how to control it????

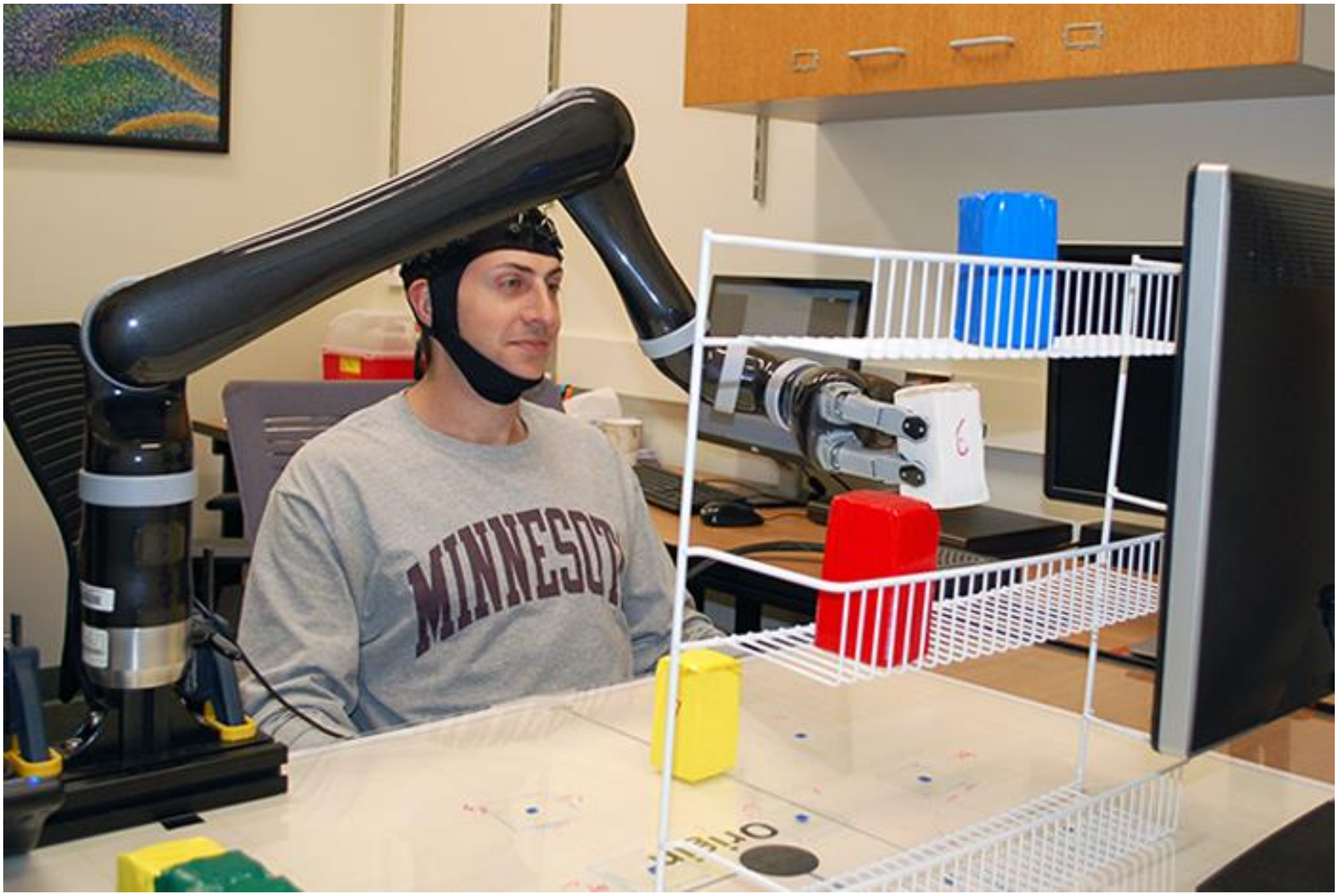
1.Robotic arm controlled manually by a potentiometer and Arduino.

The output voltage for driving the motor depends upon the two potentiometers voltage value: the driving potentiometer and the axis potentiometer. Two potentiometers are integrated into the feedback amplifier circuit. The driving potentiometer voltage works like an input voltage and it can be set by the user whereas the axis potentiometer works like feedback voltage, altering the output. The axis potentiometer voltage depends upon the position of the axis and its changes due to the rotation of the axis. When the input voltage and feedback voltage are in the same phase, then the output becomes positive and it drives the motors in a positive direction until the input voltage (driving potentiometer) and the feedback voltage (axis potentiometer) have the same voltage value. When the input voltage and feedback voltage are in the inverse phase, then the output becomes negative and it drives the motors in a negative direction until these voltages are the same. In every case, the direction of the motor is set in such a way that it rotates to change the voltage of the axis potentiometer the same as the driving potentiometer voltage and then the motor stops.





2. Mind control robotic arm



It’s a major step in the development of non-invasive brain-computer interfaces (BCIs), which build a direct communication link between the brain and an external device.

The researchers said the brain-computer interface technology works due to the geography of the motor cortex -- the area of the cerebrum that governs movement. When humans move, or think about a movement, neurons in the motor cortex produce tiny electric currents. Thinking about a different movement activates a new assortment of neurons, a phenomenon confirmed by cross-validation using functional MRI . Sorting out these assortments using advanced signal processing laid the groundwork for the brain-computer interface.

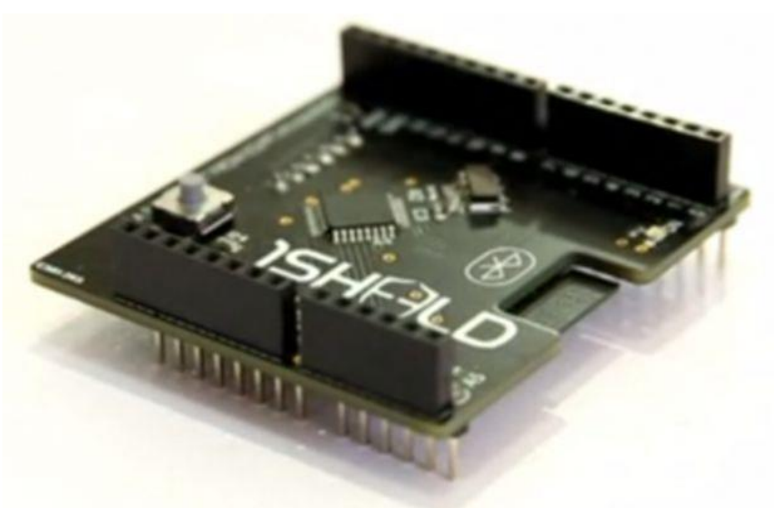
3.Robotic arm control using android phone and Arduino

send the mobile sensors data via Bluetooth to Arduino using 1Sheeld and then controlling the motion of the servos.

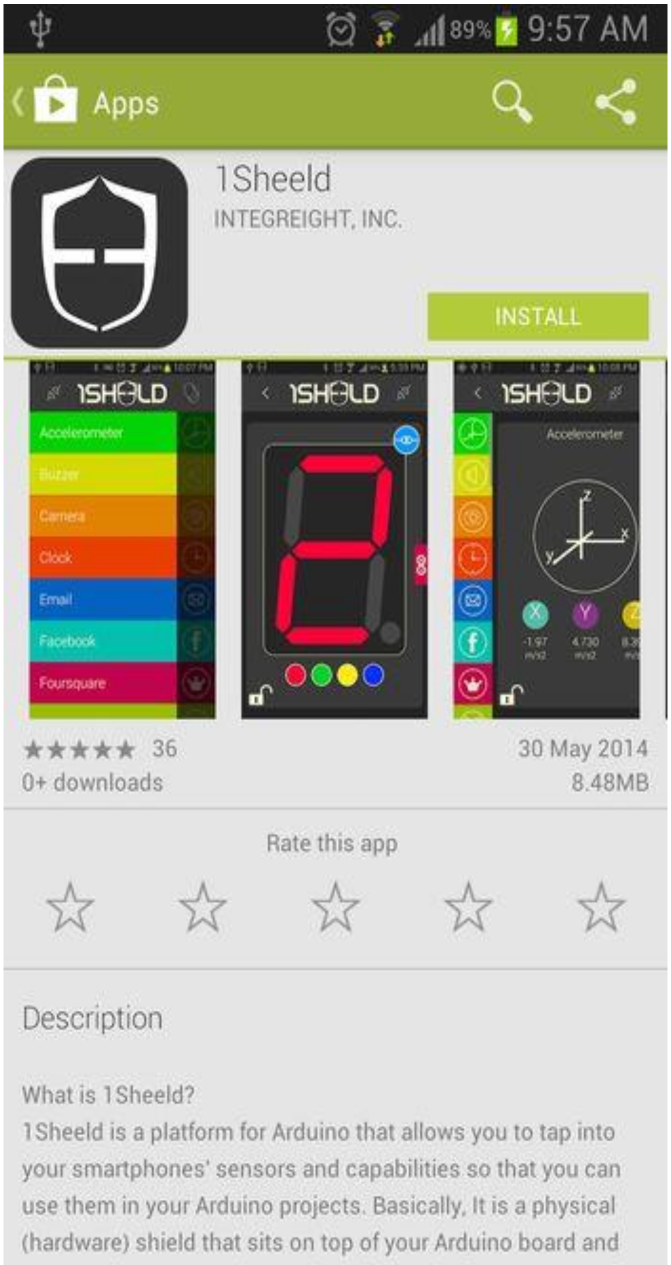
Components:

*Arduino

*1Sheeld



*Android phone with 1Sheeld mobile application



All we have to declare the pins (PWM)for each servo and then map the Value of the sensor from (0 to 180) which are the whole region the servo can rotate.

Here I’m providing a sample Arduino sketch

```
#include <OneSheeld.h>

#include <Servo.h>


Servo myservo1;

Servo myservo2;

Servo myservo3;

Servo myservo4;


void setup()

{

  OneSheeld.begin();

  myservo1.attach(3);

  myservo1.write(90);

  myservo2.attach(5);

  myservo2.write(0);

  myservo3.attach(6);

  myservo3.write(180);

  myservo4.attach(9);

  myservo4.write(15);

}


void loop() {


if (ToggleButton.getStatus ())

    myservo2.write(120);

  else

    myservo2.write(0);


if(OrientationSensor.getZ()>0 && abs(OrientationSensor.getZ())<= 83 && GravitySensor.getZ() > 0)

    myservo1.write(map(OrientationSensor.getZ(),0,90,90,0));
```

```
if(OrientationSensor.getZ()<0&&  abs(OrientationSensor.getZ()) <= 83&& GravitySensor.getZ() > 0)

    myservo1.write(map(abs(OrientationSensor.getZ()),90,0,180,90));


 if(OrientationSensor.getY() < 0 && OrientationSensor.getY() >= -90 &&  abs(GravitySensor.getZ())
>= 2.5)

    myservo4.write(map(abs(OrientationSensor.getY()),0,90,15,105));


if(OrientationSensor.getY() > 0 && OrientationSensor.getY() <= 50 &&  abs(GravitySensor.getZ()) >=
2.5)

    myservo4.write(map(abs(OrientationSensor.getY()),0,50,15,0));


if(OrientationSensor.getX() > 0 && OrientationSensor.getX() <= 180 &&  abs(GravitySensor.getZ()) >
= 2.5)

    myservo3.write(map(abs(OrientationSensor.getX()),0,180,180,0));

}
```

4.Robotic arm control using joystick

The USB joystick is plugged into a computer which runs Xpadder and Processing.
Xpadder converts each movement of the joystick into letters (it acts like when someone hits a key on a keyboard).
The processing code detects if any key is pressed and sends a specific number to the Arduino via serial communication according to this key.

The Arduino assigns to each number a movement of the motor (polarity included) and writes a 3V voltage on the corresponding pins.
As the DC motors require more power, it is necessary to use an external power supply (a lab power bench in this case, but it can also be a battery)
The motors worked with 3V with the robot arm's batteries, but after a few tests, it seemed to work better with ~5-7V

The L293D chips make a bridge between the Arduino and the "high" voltage of the motors (which may damage the Arduino board).
When the 3V voltage comes in a specific pin of the chip, it lets the 6V voltage go to the motors with a certain polarity.
A single L293D chip can control 2 DC motors.

5.Gyro-Accelerometer based control of a robotic Arm

*A gesture recognition based 6DOF robotic arm controller using gyro-meter with accelerometer to improve the stability and to detect the rotational gesture of human arm.

*The whole system divided into two sections. One is data transmitting section and another is data receiving section.

*These two systems are interfaced with **xbee protocol**.

* We can use handmade artificial robotic arm which contains

180degree rotation angle.

* We also use MPU-6050 IMU board. The MPU-6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die together with an onboard **Digital Motion Processor (DMP)** capable of

processing complex 9-axis Motion Fusion algorithms. The

parts feature a user-programmable gyro full-scale range of

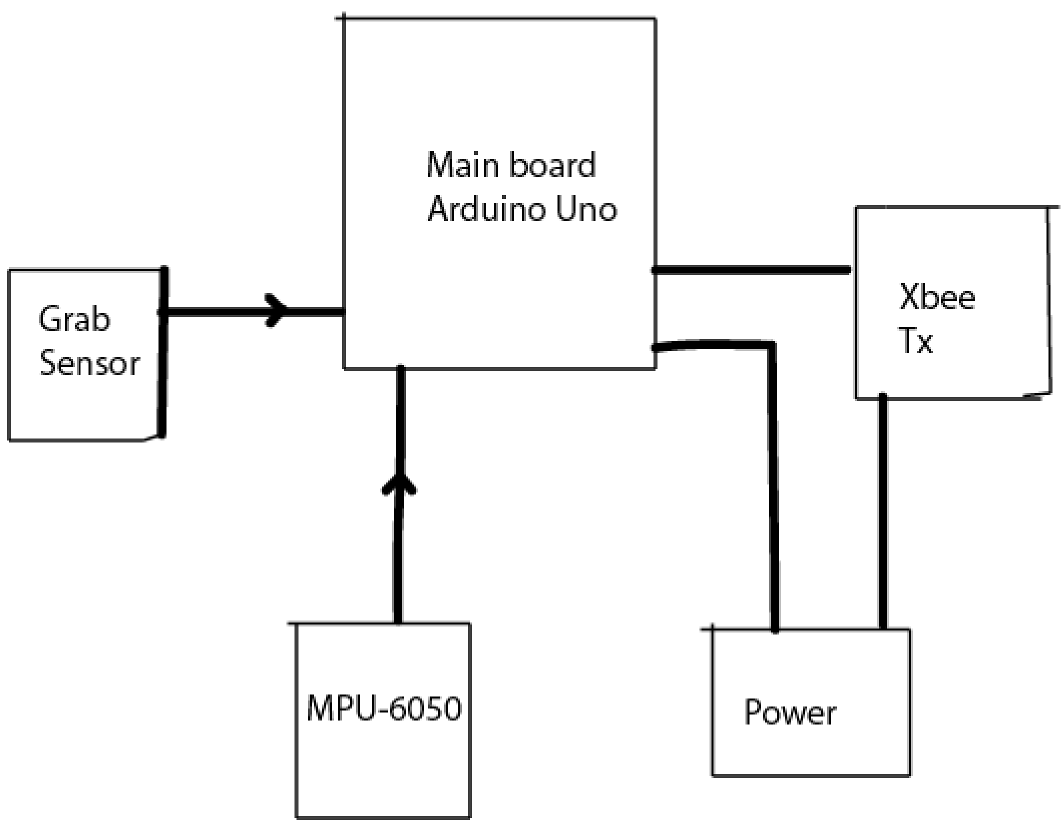
±250, ±500, ±1000, and ±2000°/sec (dps) and a user programmable

accelerometer full-scale range of ±2*g*, ±4*g*, ±8*g*,

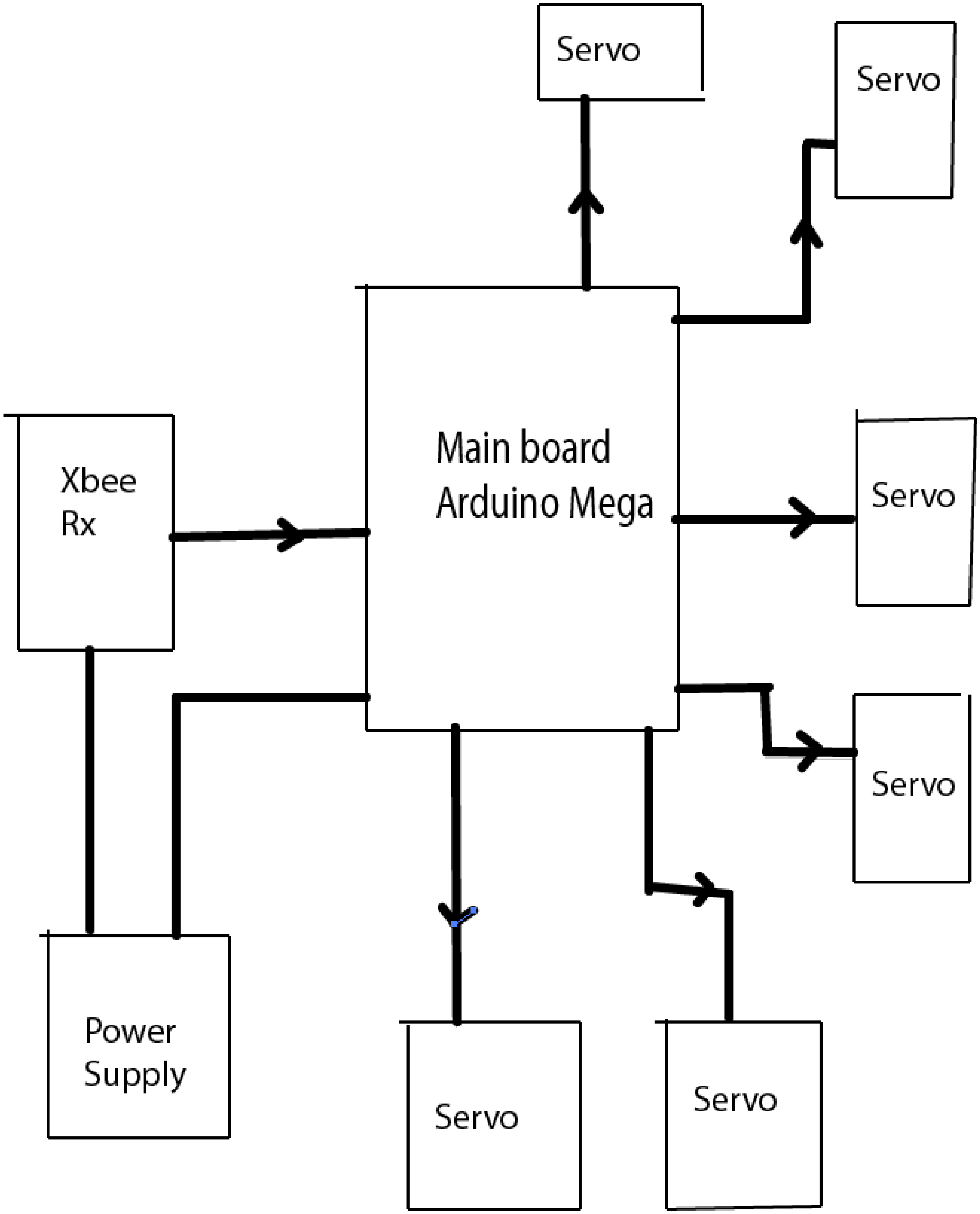
and ±16*g*. It’s detects hand gesture and send data to main

board for processing. With the help of complex geometry, we calculate the proper angle of movement of hand gesture in main board. Then these data send to receiving section by xbee module. Transmitted data received by receiving xbee module and processed by receiving sections main board. Then servo moves with main board instruction which is received from transmitter section. The main board for processing unit used Arduino Uno (contains ATmega microcontroller).

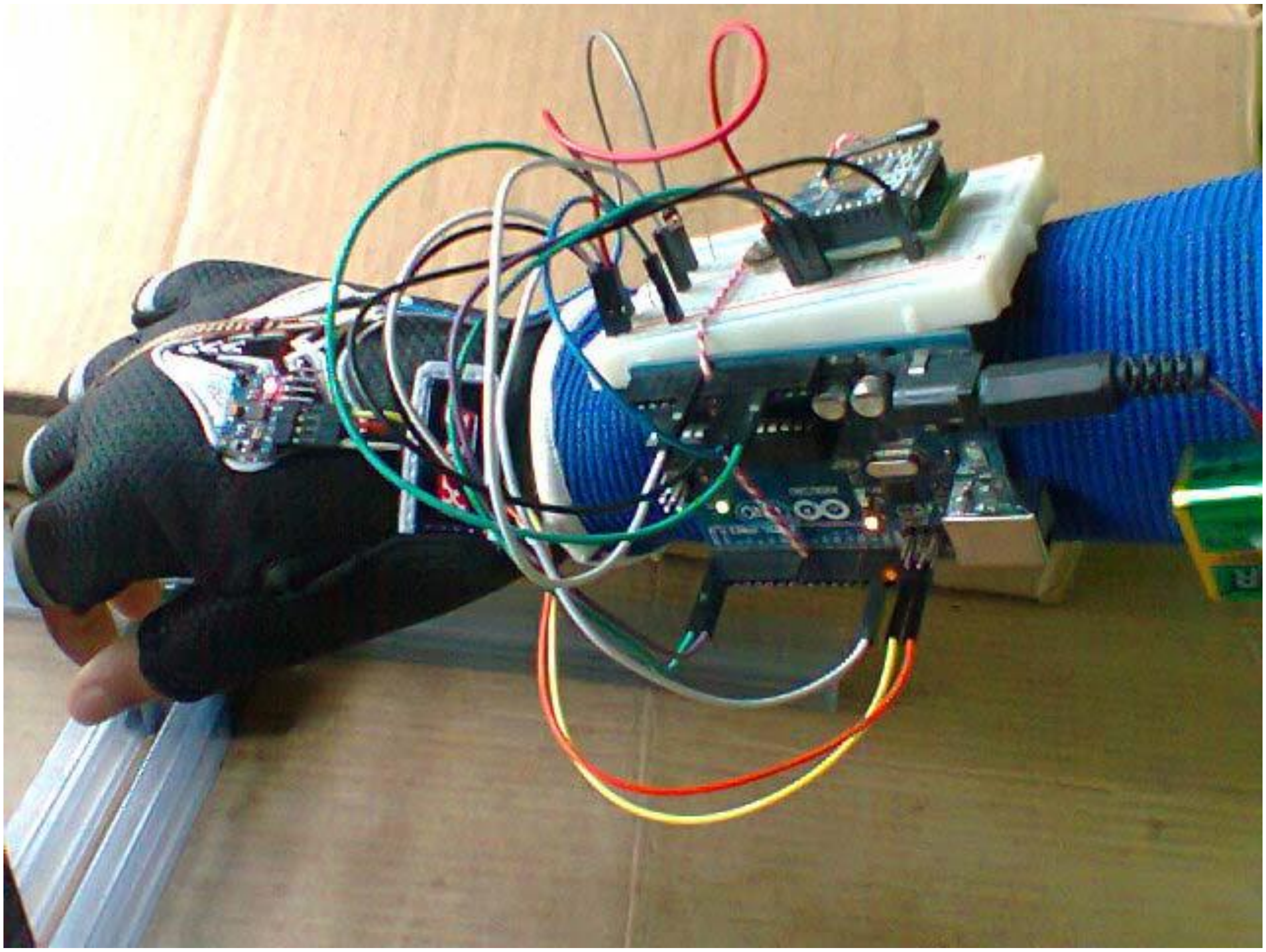
BLOCK DIAGRAM



Block Diagram of hand gesture transmitting section.



Block Diagram of hand gesture data receiving section. This system also control the robotic arm.



Wireless data transmitting section with Xbee module. Figure shows MPU5060 IMU board. Total system is able to transmit hand gesture data wirelessly.