PROJECT REPORT

**SHADOW ROBOT**

GROUP: ROBOCREED

**GROUP MEMBERS:**

FERDINAND THOMAS

JACOB SABU

ABHIJITH S MENON

AKASH K A

GODLy THOMAS

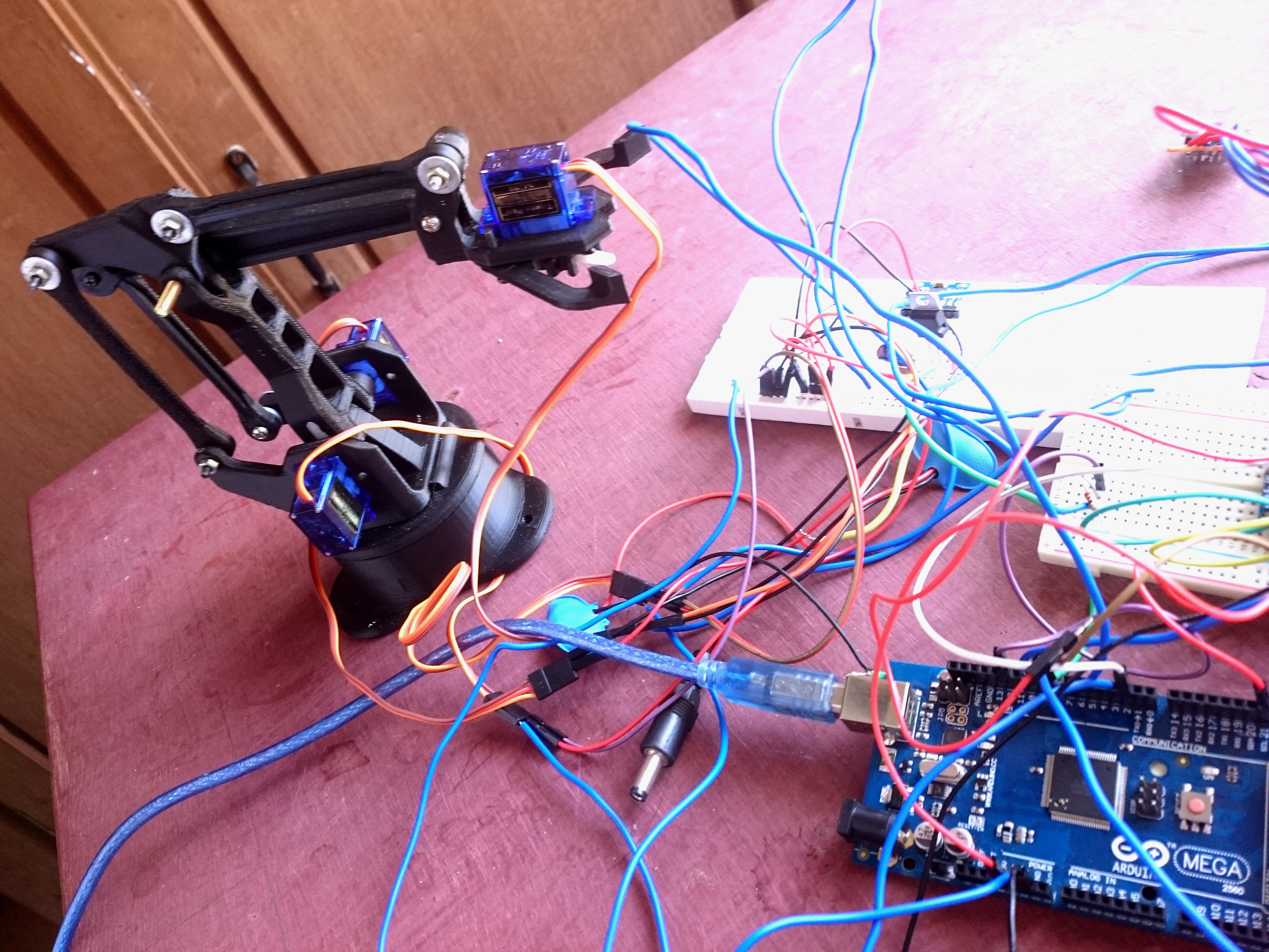
The project is a robotic arm which mimics the motion of the arm of the user

COMPONENTS REQUIRED:

* Robotic arm (1)
* Arduino mega 2560(1)
* MPU 6050 (2)
* Servo micro SG 90(4)
* Breadboard (2)
* 9 V power source (1)

WORKING:

Shadow Robot is a robotic arm which mimics the motion of the user’s arm .A gyroscope sensor(MPU6050) is used to get motion of the arm .MPU6050 reads the variation of angles of the arm of the user and the readings are then send to the arduino through jumper wires and these are then converted to angle in 0 to 180 degree in arduino program and the angle fed to the servos (micro SG 90) on the Robotic arm ,which in turn make the robotic arm move the same way in which the users arm is moved.



APPLICATION:

**REMOTE SURGERY**

Remote surgery (also known as telesurgery) is the ability for a doctor to perform surgery on a patient even though they are not physically in the same location. It is a form of telepresence. A robot surgical system generally consists of one or more arms (controlled by the surgeon), a master controller (console), and a sensory system giving feedback to the user. Remote surgery combines elements of robotics, cutting edge communication technology such as high-speed data connections and elements of management information systems. While the field of robotic surgery is fairly well established, most of these robots are controlled by surgeons at the location of the surgery. Remote surgery is essentially advanced telecommuting for surgeons, where the physical distance between the surgeon and the patient is immaterial. It promises to allow the expertise of specialized surgeons to be available to patients worldwide, without the need for patients to travel beyond their local hospital.

BOMB DEFUSE BOT

This system is very beneficial in areas where there is high risk for humans to enter .this system makes use of robotic arm as well as robotic vehicle which helps not only to enter an area involving high risk but also to pick whatever object it wants to . the system also includes night vision camera which will not only allow viewing whatever will be recorded in day time but also during night .the arm will be controlled by the bomb diffuser at a distant using our shadow robotic arm .thus enabling him to do delicate works like bomb diffusing.

Arduino code:

#include "Wire.h"

#include "I2Cdev.h"

#include "MPU6050.h"

#include<Servo.h>

MPU6050 accelgyroIC1(0x68);

MPU6050 accelgyroIC2(0x69);

Servo one,two,three,grip;

int16\_t ax1, ay1, az1;

int16\_t gx1, gy1, gz1;

int16\_t ax2, ay2, az2;

int16\_t gx2, gy2, gz2;

double accXangle1;

double accYangle1;

double accZangle1;

double accXangle2;

double accYangle2;

double accZangle2;

double gyroXangle1 = 90;

double gyroYangle1 = 90;

double gyroZangle1 = 90;

double gyroXangle2 = 90;

double gyroYangle2 = 90;

double gyroZangle2 = 90;

#define LED\_PIN 13

bool blinkState = false;

bool bstate = false;

int button = 12;

void setup() {

Wire.begin();

one.attach(9);

two.attach(10);

three.attach(3);

grip.attach(5);

Serial.begin(38400);

Serial.println("Initializing I2C devices...");

accelgyroIC1.initialize();

Serial.println(accelgyroIC1.testConnection() ? "MPU6050 #1 connection successful" : "MPU6050 connection failed");

pinMode(8,OUTPUT);

digitalWrite(8,HIGH);

accelgyroIC2.initialize();

// verify connection

Serial.println("Testing device connections...");

Serial.println(accelgyroIC2.testConnection() ? "MPU6050 #2 connection successful" : "MPU6050 connection failed");

// configure Arduino LED for

pinMode(LED\_PIN, OUTPUT);

pinMode(button ,INPUT);

}

void loop() {

uint32\_t timer;

timer=micros();

// read raw accel/gyro measurements from device

accelgyroIC1.getMotion6(&ax1, &ay1, &az1, &gx1, &gy1, &gz1);

delay(10);

digitalWrite(8,HIGH);

accelgyroIC2.getMotion6(&ax2, &ay2, &az2, &gx2, &gy2, &gz2);

delay(10);

// these methods (and a few others) are also available

//accelgyro.getAcceleration(&ax, &ay, &az);

//accelgyro.getRotation(&gx, &gy, &gz);

accXangle1 = (atan2(ax1,az1) + PI) \* RAD\_TO\_DEG;

accYangle1 = (atan2(ay1,az1) + PI) \* RAD\_TO\_DEG;

accZangle1 = (atan2(ax1,ay1) + PI) \* RAD\_TO\_DEG;

accXangle2 = (atan2(ax2,az2) + PI) \* RAD\_TO\_DEG;

accYangle2 = (atan2(ay2,az2) + PI) \* RAD\_TO\_DEG;

accZangle2 = (atan2(ax2,ay2) + PI) \* RAD\_TO\_DEG;

double gyroXrate1 = ((double)gx1 / 131.0);

double gyroYrate1 = ((double)gy1 / 131.0);

double gyroZrate1 = ((double)gz1 / 131.0);

double gyroXrate2 = ((double)gx2 / 131.0);

double gyroYrate2 = ((double)gy2 / 131.0);

double gyroZrate2 = ((double)gz2 / 131.0);

gyroXangle1 += gyroXrate1\*((double)(micros()-timer)/1000000);

gyroYangle1 += gyroYrate1\*((double)(micros()-timer)/1000000);

gyroZangle1 += gyroZrate1\*((double)(micros()-timer)/1000000);

gyroXangle2 += gyroXrate2\*((double)(micros()-timer)/1000000);

gyroYangle2 += gyroYrate2\*((double)(micros()-timer)/1000000);

gyroZangle2 += gyroZrate2\*((double)(micros()-timer)/1000000);

gyroXangle1 += gyroXrate1\*((double)(micros()-timer)/1000000);

gyroYangle1 += gyroYrate1\*((double)(micros()-timer)/1000000);

gyroZangle1 += gyroZrate1\*((double)(micros()-timer)/1000000);

gyroXangle2 += gyroXrate2\*((double)(micros()-timer)/1000000);

gyroYangle2 += gyroYrate2\*((double)(micros()-timer)/1000000);

gyroZangle2 += gyroZrate2\*((double)(micros()-timer)/1000000);

constrain(gyroXangle1,0,179);

constrain(gyroYangle1,0,179);

constrain(gyroZangle1,0,179);

constrain(gyroXangle2,0,179);

constrain(gyroYangle2,0,179);

constrain(gyroZangle2,0,179);

one.write(gyroZangle1);

two.write(180-gyroYangle1);

three.write(gyroYangle2);

// display tab-separated accel/gyro x/y/z values

Serial.print("MPU1:\t");

Serial.print(gyroXangle1); Serial.print("\t");

Serial.print(gyroYangle1); Serial.print("\t");

Serial.print(gyroZangle1); Serial.print("\t");

// display tab-separated accel/gyro x/y/z values

Serial.print("MPU2:\t");

Serial.print(gyroXangle2); Serial.print("\t");

Serial.print(gyroYangle2); Serial.print("\t");

Serial.println(gyroZangle2);

delay(20);

// blink LED to indicate activity

blinkState = !blinkState;

digitalWrite(LED\_PIN, blinkState);

bstate = digitalRead(button);

delay(10);

if(bstate == 0)

{

grip.write(90);

}

else

{

grip.write(40);

}

Serial.println(bstate);

}