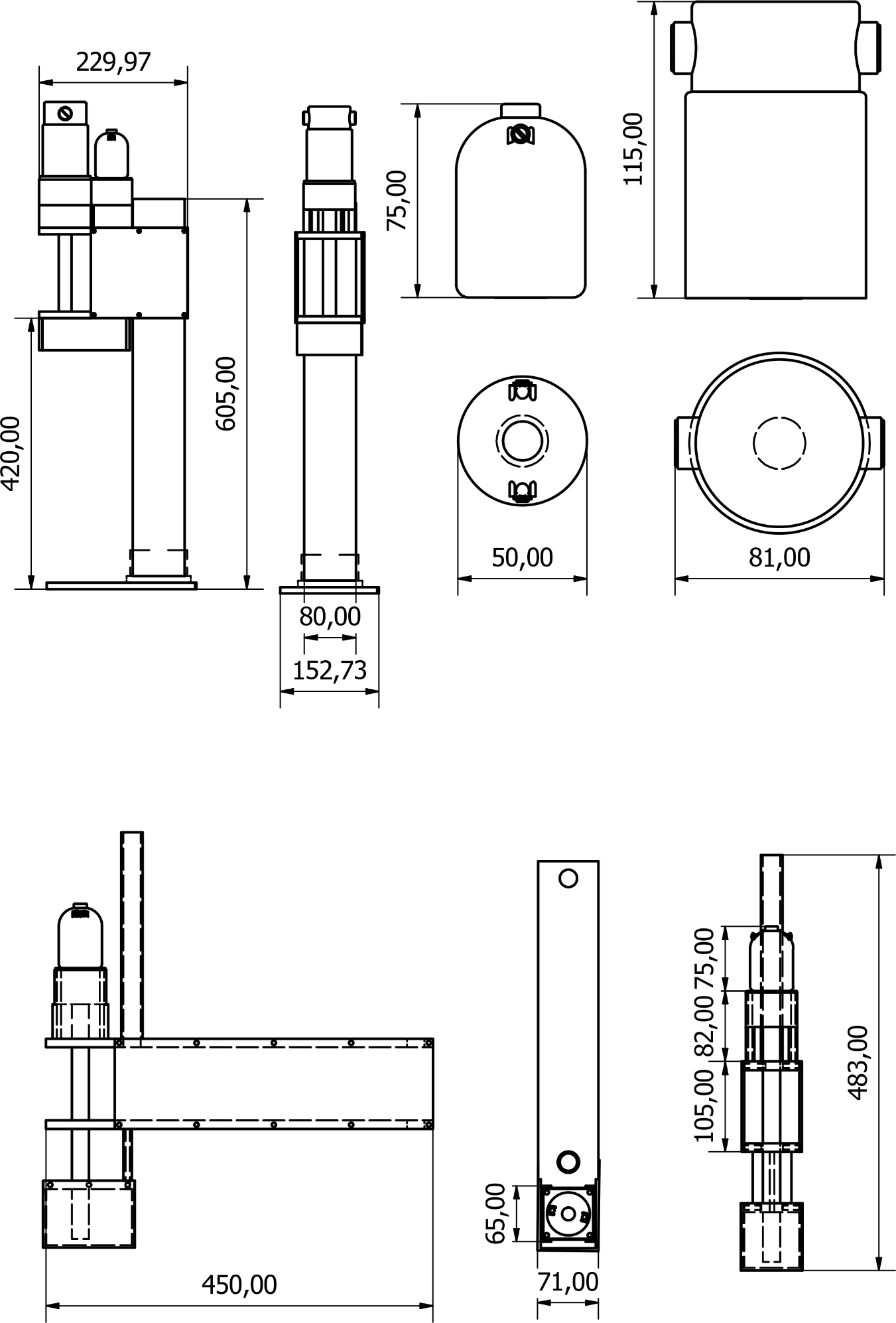
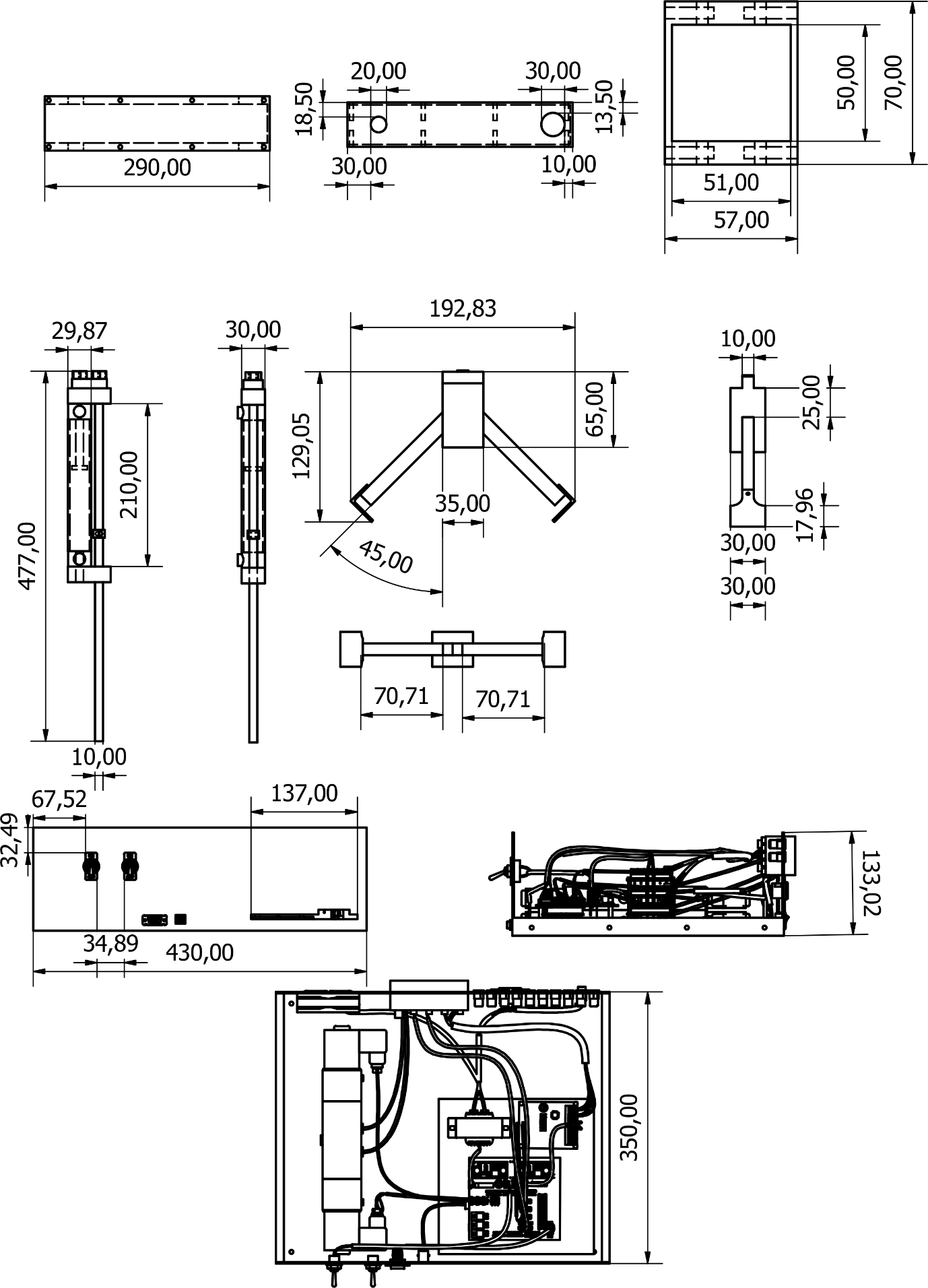
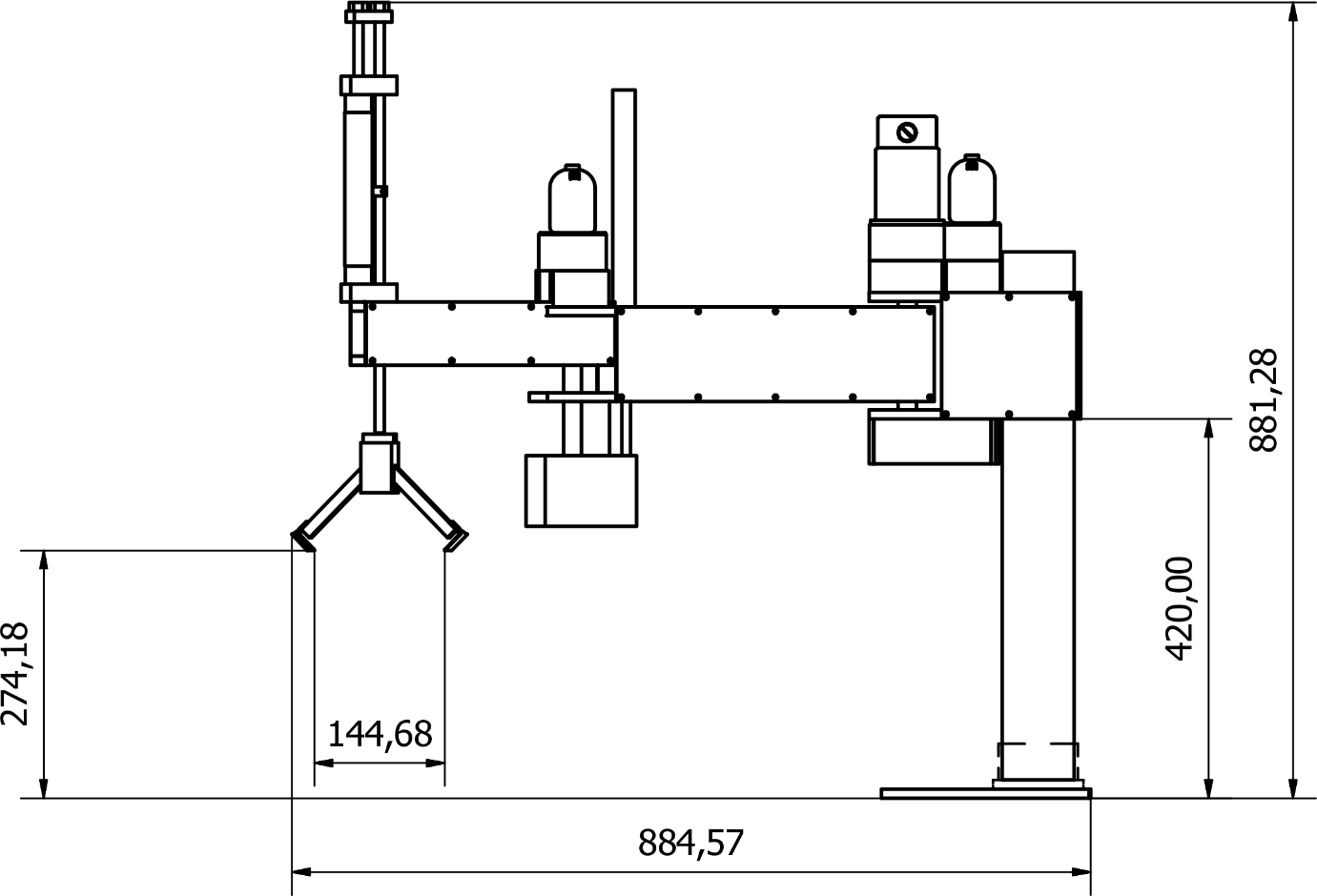
**LAMPIRAN**

**Desain 3D CAD SCARA Serpent** 





**Program Arduino IDE SCARA Serpent**

#include <MServo.h>

#include <stdio.h>

#include <math.h>

#define SpotPin A10 // Pin that connect to Potentiometer

#define SdirAPin 26 // Pin A to control the motor direction

#define SdirBPin 24 // Pin B to control the motor direction

#define SpwmPin 2 // Pin to control the motor speed

#define EpotPin A9 // Pin that connect to Potentiometer

#define EdirAPin 32 // Pin A to control the motor direction

#define EdirBPin 34 // Pin B to control the motor direction

#define EpwmPin 3 // Pin to control the motor speed

#define WpotPin A8 // Pin that connect to Potentiometer

#define WdirAPin 40 // Pin A to control the motor direction

#define WdirBPin 42 // Pin B to control the motor direction

#define WpwmPin 4 // Pin to control the motor speed

#define p1 10

#define p2 11

#define p3 12

char START\_BYTE = '\*'; //three characters used for Serial communication

char DELIMITER = ',';

char END\_BYTE = '#';

float vps, vpe, vpw, vpp1, vpp2;

int Sremote, Eremote, Wremote, P1remote, P2remote;

int minAngle = 0;

int maxAngle = 180;

int minAngleW = 0;

int maxAngleW = 360;

int val = 0;

int diff = 1;

unsigned long previousMillis = 0;

float KP, KI, KD, setkp, setki, setkd, KPremote;

float fKP = 2;

float fKI = 2;

float fKD = 2;

MServo myservoS(SpotPin, SdirAPin, SdirBPin, SpwmPin);

MServo myservoE(EpotPin, EdirAPin, EdirBPin, EpwmPin);

MServo myservoW(WpotPin, WdirAPin, WdirBPin, WpwmPin);

MServo remoteS(A15, 50, 51, 52);

MServo remoteE(A14 , 50, 51, 52);

MServo remoteW(A13, 50, 51, 52);

MServo remoteP1(A12, 50, 51, 52);

MServo remoteP2(A11, 50, 51, 52);

void setup() {

Serial.begin(115200);

pinMode(p1, OUTPUT);

pinMode(p2, OUTPUT);

pinMode(p3, OUTPUT);

mservo();

}

void loop() {

remote();

Read\_command();

KP = setkp / 100.f;

KI = setki / 100.f;

KD = setkd / 100.f;

pneumatic();

myservoS.setParam(KP, KI, KD);

myservoE.setParam(KP, KI, KD);

myservoW.setParam(KP, KI, KD);

// myservoS.setKP(KP);

// myservoE.setKP(KP);

// myservoW.setKP(KP);

myservoS.write(vps);

myservoE.write(vpe);

myservoW.write(vpw);

myservoE.update();

myservoS.update();

myservoW.update();

delay(50);

Serial.write(START\_BYTE); Serial.print(DELIMITER);

Serial.print(myservoS.getAngle()); Serial.print(DELIMITER); //1

Serial.print(myservoE.getAngle()); Serial.print(DELIMITER); //2

Serial.print(myservoW.getAngle()); Serial.print(DELIMITER); //3

Serial.print(Sremote); Serial.print(DELIMITER); //4

Serial.print(Eremote); Serial.print(DELIMITER); //5

Serial.print(Wremote); Serial.print(DELIMITER); //6

Serial.print(P1remote); Serial.print(DELIMITER); //7

Serial.print(P2remote); Serial.print(DELIMITER); //8

Serial.write(END\_BYTE); Serial.println(); //send a carriage return

}

int val = 0;

int diff = 1;

unsigned long previousMillis = 0;

float KP, KI, KD, setkp, setki, setkd, KPremote;

float fKP = 2;

float fKI = 2;

float fKD = 2;

MServo myservoS(SpotPin, SdirAPin, SdirBPin, SpwmPin);

MServo myservoE(EpotPin, EdirAPin, EdirBPin, EpwmPin);

MServo myservoW(WpotPin, WdirAPin, WdirBPin, WpwmPin);

MServo remoteS(A15, 50, 51, 52);

MServo remoteE(A14 , 50, 51, 52);

MServo remoteW(A13, 50, 51, 52);

MServo remoteP1(A12, 50, 51, 52);

MServo remoteP2(A11, 50, 51, 52);

void setup() {

Serial.begin(115200);

pinMode(p1, OUTPUT);

pinMode(p2, OUTPUT);

pinMode(p3, OUTPUT);

mservo();

}

void loop() {

remote();

Read\_command();

KP = setkp / 100.f;

KI = setki / 100.f;

KD = setkd / 100.f;

pneumatic();

myservoS.setParam(KP, KI, KD);

myservoE.setParam(KP, KI, KD);

myservoW.setParam(KP, KI, KD);

// myservoS.setKP(KP);

// myservoE.setKP(KP);

// myservoW.setKP(KP);

myservoS.write(vps);

myservoE.write(vpe);

myservoW.write(vpw);

myservoE.update();

myservoS.update();

myservoW.update();

delay(50);

Serial.write(START\_BYTE); Serial.print(DELIMITER);

Serial.print(myservoS.getAngle()); Serial.print(DELIMITER); //1

Serial.print(myservoE.getAngle()); Serial.print(DELIMITER); //2

Serial.print(myservoW.getAngle()); Serial.print(DELIMITER); //3

Serial.print(Sremote); Serial.print(DELIMITER); //4

Serial.print(Eremote); Serial.print(DELIMITER); //5

Serial.print(Wremote); Serial.print(DELIMITER); //6

Serial.print(P1remote); Serial.print(DELIMITER); //7

Serial.print(P2remote); Serial.print(DELIMITER); //8

Serial.write(END\_BYTE); Serial.println(); //send a carriage return

}

KI = setki / 100.f;

KD = setkd / 100.f;

pneumatic();

myservoS.setParam(KP, KI, KD);

myservoE.setParam(KP, KI, KD);

myservoW.setParam(KP, KI, KD);

myservoS.write(vps);

myservoE.write(vpe);

myservoW.write(vpw);

myservoE.update();

myservoS.update();

myservoW.update();

delay(50);

Serial.write(START\_BYTE); Serial.print(DELIMITER);

Serial.print(myservoS.getAngle()); Serial.print(DELIMITER); //1

Serial.print(myservoE.getAngle()); Serial.print(DELIMITER); //2

Serial.print(myservoW.getAngle()); Serial.print(DELIMITER); //3

Serial.print(Sremote); Serial.print(DELIMITER); //4

Serial.print(Eremote); Serial.print(DELIMITER); //5

Serial.print(Wremote); Serial.print(DELIMITER); //6

Serial.print(P1remote); Serial.print(DELIMITER); //7

Serial.print(P2remote); Serial.print(DELIMITER); //8

Serial.write(END\_BYTE); Serial.println(); //send a carriage return

}

import processing.serial.Serial; // serial library

import controlP5.\*; // controlP5 library

Serial serial;

ControlP5 cp5;

InverseKinematic ik1;

PFont font9, font10, font12, font14, font18, font20, font25, font30, font35;

Chart current\_chart, respon\_chart;

color yellow\_ = color(200, 200, 20), green\_ = color(46, 209, 2), red\_ = color(120, 30, 30), blue\_ = color (0, 102, 200);

color white\_ = color(255, 255, 255), black\_ = color(0, 0, 0), pink\_ = color(255, 0, 255);

//UART Variable

int serial\_conect = 0;

int commListMax;

int[] data = null;

Textlabel txtlblWhichcom;

ListBox commListbox;

ListBox portlist;

// coded by Eberhard Rensch

// Truncates a long port name for better (readable) display in the GUI

String shortifyPortName(String portName, int maxlen)

{

String shortName = portName;

if (shortName.startsWith("/dev/")) shortName = shortName.substring(5);

if (shortName.startsWith("tty.")) shortName = shortName.substring(4); // get rid off leading tty. part of device name

if (portName.length()>maxlen) shortName = shortName.substring(0, (maxlen-1)/2) + "~" +shortName.substring(shortName.length()-(maxlen-(maxlen-1)/2));

if (shortName.startsWith("cu.")) shortName = "";// only collect the corresponding tty. devices

return shortName;

}

boolean START = false;

boolean MODE = false;

boolean ik = true;

boolean P1 = false;

boolean P2 = false;

int KoordinatX, KoordinatY;

int KoordinatX\_, KoordinatY\_;

float fs, fe, fw, rs, re, rw, rp1, rp2;

int ps=380;

int pe=280;

float beta, gamma;

int Ibeta, Igamma;

int cb1, cb2;

float cb3;

int mosxe, mosye, mosex, mosey;

String textValue = "";

float KP;

float KI;

float KD;

int \_KoordinatX, \_KoordinatY;

int mosx, mosy;

int s=5;

//revision v1

int count\_click = 0;

int[] clickX = new int[100];

int[] clickY = new int[100];

int X\_rev = 50;

int Y\_rev = 120;

int statustab=1;

int[] X\_odometry = new int[5];

int[] Y\_odometry = new int[5];

int[][] color\_point = new int[6][3];

float[][] coordinat\_input= new float[6][2];

int []fxcoordinat=new int[100];

int []fycoordinat=new int[100];

void setup()

{

for (int a=0; a<100; a++) {

colorR[a] = random(255);

colorG[a] = random(255);

colorB[a] = random(255);

}

size(1374, 750, OPENGL); //ukuran window

image\_();

cp5 = new ControlP5(this);

ik1 = new InverseKinematic(ps, pe);

font9 = createFont("Arial Bold", 9, false);

font10 = createFont("Arial Bold", 10, false);

font12 = createFont("Arial Bold", 12, false);

font14 = createFont("Arial Bold", 14, false);

font18 = createFont("Arial Bold", 18, false);

font20 = createFont("Arial Bold", 20, false);

font25 = createFont("Arial Bold", 25, false);

font30 = createFont("Arial Bold", 30, false);

font35 = createFont("Arial Bold", 35, false);

GUI\_setup();

setup\_UART();

sobj();

}

float koorx=0;

float koory=90;

int cx, cy;

float r2;

int koorx1, koory1;

void mouseClicked() {

r= sqrt(pow(mouseX-680, 2)+pow(mouseY-545, 2));

r2= sqrt(pow(mouseX-540, 2)+pow(mouseY-545, 2));

if (r<600/2 && r > 351/2 && mouseY <545) {

koorx1=mouseX;

koory1=mouseY;

koorx=int(map(mouseX, 381, 979, -660, 660));

koory=int(map(mouseY, 246, 546, 660, 0));

}

if(r2<225&& r2> 115/\*&& mouseY <545\*/) {

if ((mouseX<=X\_rev+130+500-150+525) && (mouseX>=X\_rev+130-150)) {

if ((mouseY<=Y\_rev+500) && (mouseY>=Y\_rev)) {

count\_click+=1;

clickX[count\_click] = int((mouseX -430+150-525/2)/2.5);

clickY[count\_click] = int(-(mouseY -370-525/3)/2.5);

cx=clickX[count\_click];

cy=clickY[count\_click];

}}}}

float v22=0.0;

int v33, v44;

void draw()

{cp5.getController("RESET\_KOORDINAT").moveTo("fiture");

cp5.getController("RUN").moveTo("fiture");

cp5.getController("v22").moveTo("fiture");

cp5.getController("v33").moveTo("fiture");

cp5.getController("v44").moveTo("fiture");

//println(mouseX, mouseY);

if (statustab==1) //TAB MAIN

{

background(bg);

obj();

images();

fw();

Send\_To\_Arduino();

}

if (statustab==2) //TAB CHART

{

background(bg); //0-->Black

fw();

fill(255);

imageMode(CENTER);

pushMatrix();

scale(0.45);

image(workspace, 1187, 948);

popMatrix();

image(judul, width/2, 45);

draw\_coordinat\_target();

draw\_coordinat();

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

**Program Processing IDE SCARA Serpent**

import processing.serial.Serial; // serial library

import controlP5.\*; // controlP5 library

Serial serial;

ControlP5 cp5;

InverseKinematic ik1;

PFont font9, font10, font12, font14, font18, font20, font25, font30, font35;

Chart current\_chart, respon\_chart;

color yellow\_ = color(200, 200, 20), green\_ = color(46, 209, 2), red\_ = color(120, 30, 30), blue\_ = color (0, 102, 200);

color white\_ = color(255, 255, 255), black\_ = color(0, 0, 0), pink\_ = color(255, 0, 255);

//UART Variable

int serial\_conect = 0;

int commListMax;

int[] data = null;

Textlabel txtlblWhichcom;

ListBox commListbox;

ListBox portlist;

// coded by Eberhard Rensch

// Truncates a long port name for better (readable) display in the GUI

String shortifyPortName(String portName, int maxlen)

{

String shortName = portName;

if (shortName.startsWith("/dev/")) shortName = shortName.substring(5);

if (shortName.startsWith("tty.")) shortName = shortName.substring(4); // get rid off leading tty. part of device name

if (portName.length()>maxlen) shortName = shortName.substring(0, (maxlen-1)/2) + "~" +shortName.substring(shortName.length()-(maxlen-(maxlen-1)/2));

if (shortName.startsWith("cu.")) shortName = "";// only collect the corresponding tty. devices

return shortName;

}

boolean START = false;

boolean MODE = false;

boolean ik = true;

boolean P1 = false;

boolean P2 = false;

int KoordinatX, KoordinatY;

int KoordinatX\_, KoordinatY\_;

float fs, fe, fw, rs, re, rw, rp1, rp2;

int ps=380;

int pe=280;

float beta, gamma;

int Ibeta, Igamma;

int cb1, cb2;

float cb3;

int mosxe, mosye, mosex, mosey;

String textValue = "";

float KP;

float KI;

float KD;

int \_KoordinatX, \_KoordinatY;

int mosx, mosy;

int s=5;

//revision v1

int count\_click = 0;

int[] clickX = new int[100];

int[] clickY = new int[100];

int X\_rev = 50;

int Y\_rev = 120;

int statustab=1;

int[] X\_odometry = new int[5];

int[] Y\_odometry = new int[5];

int[][] color\_point = new int[6][3];

float[][] coordinat\_input= new float[6][2];

int []fxcoordinat=new int[100];

int []fycoordinat=new int[100];

void setup()

{

for (int a=0; a<100; a++) {

colorR[a] = random(255);

colorG[a] = random(255);

colorB[a] = random(255);

}

size(1374, 750, OPENGL); //ukuran window

image\_();

cp5 = new ControlP5(this);

ik1 = new InverseKinematic(ps, pe);

font9 = createFont("Arial Bold", 9, false);

font10 = createFont("Arial Bold", 10, false);

font12 = createFont("Arial Bold", 12, false);

font14 = createFont("Arial Bold", 14, false);

font18 = createFont("Arial Bold", 18, false);

font20 = createFont("Arial Bold", 20, false);

font25 = createFont("Arial Bold", 25, false);

font30 = createFont("Arial Bold", 30, false);

font35 = createFont("Arial Bold", 35, false);

GUI\_setup();

setup\_UART();

sobj();

}

float koorx=0;

float koory=90;

int cx, cy;

float r2;

int koorx1, koory1;

void mouseClicked() {

r= sqrt(pow(mouseX-680, 2)+pow(mouseY-545, 2));

r2= sqrt(pow(mouseX-540, 2)+pow(mouseY-545, 2));

if (r<600/2 && r > 351/2 && mouseY <545) {

koorx1=mouseX;

koory1=mouseY;

koorx=int(map(mouseX, 381, 979, -660, 660));

koory=int(map(mouseY, 246, 546, 660, 0));

}

if(r2<225&& r2> 115/\*&& mouseY <545\*/) {

if ((mouseX<=X\_rev+130+500-150+525) && (mouseX>=X\_rev+130-150)) {

if ((mouseY<=Y\_rev+500) && (mouseY>=Y\_rev)) {

count\_click+=1;

clickX[count\_click] = int((mouseX -430+150-525/2)/2.5);

clickY[count\_click] = int(-(mouseY -370-525/3)/2.5);

cx=clickX[count\_click];

cy=clickY[count\_click];

}}}}

float v22=0.0;

int v33, v44;

void draw()

{cp5.getController("RESET\_KOORDINAT").moveTo("fiture");

cp5.getController("RUN").moveTo("fiture");

cp5.getController("v22").moveTo("fiture");

cp5.getController("v33").moveTo("fiture");

cp5.getController("v44").moveTo("fiture");

//println(mouseX, mouseY);

if (statustab==1) //TAB MAIN

{

background(bg);

obj();

images();

fw();

Send\_To\_Arduino();

}

if (statustab==2) //TAB CHART

{

background(bg); //0-->Black

fw();

fill(255);

imageMode(CENTER);

pushMatrix();

scale(0.45);

image(workspace, 1187, 948);

popMatrix();

image(judul, width/2, 45);

draw\_coordinat\_target();

draw\_coordinat();

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

import processing.serial.Serial; // serial library

import controlP5.\*; // controlP5 library

Serial serial;

ControlP5 cp5;

InverseKinematic ik1;

PFont font9, font10, font12, font14, font18, font20, font25, font30, font35;

Chart current\_chart, respon\_chart;

color yellow\_ = color(200, 200, 20), green\_ = color(46, 209, 2), red\_ = color(120, 30, 30), blue\_ = color (0, 102, 200);

color white\_ = color(255, 255, 255), black\_ = color(0, 0, 0), pink\_ = color(255, 0, 255);

//UART Variable

int serial\_conect = 0;

int commListMax;

int[] data = null;

Textlabel txtlblWhichcom;

ListBox commListbox;

ListBox portlist;

// coded by Eberhard Rensch

// Truncates a long port name for better (readable) display in the GUI

String shortifyPortName(String portName, int maxlen)

{

String shortName = portName;

if (shortName.startsWith("/dev/")) shortName = shortName.substring(5);

if (shortName.startsWith("tty.")) shortName = shortName.substring(4); // get rid off leading tty. part of device name

if (portName.length()>maxlen) shortName = shortName.substring(0, (maxlen-1)/2) + "~" +shortName.substring(shortName.length()-(maxlen-(maxlen-1)/2));

if (shortName.startsWith("cu.")) shortName = "";// only collect the corresponding tty. devices

return shortName;

}

boolean START = false;

boolean MODE = false;

boolean ik = true;

boolean P1 = false;

boolean P2 = false;

int KoordinatX, KoordinatY;

int KoordinatX\_, KoordinatY\_;

float fs, fe, fw, rs, re, rw, rp1, rp2;

int ps=380;

int pe=280;

float beta, gamma;

int Ibeta, Igamma;

int cb1, cb2;

float cb3;

int mosxe, mosye, mosex, mosey;

String textValue = "";

float KP;

float KI;

float KD;

int \_KoordinatX, \_KoordinatY;

int mosx, mosy;

int s=5;

//revision v1

int count\_click = 0;

int[] clickX = new int[100];

int[] clickY = new int[100];

int X\_rev = 50;

int Y\_rev = 120;

int statustab=1;

int[] X\_odometry = new int[5];

int[] Y\_odometry = new int[5];

int[][] color\_point = new int[6][3];

float[][] coordinat\_input= new float[6][2];

int []fxcoordinat=new int[100];

int []fycoordinat=new int[100];

void setup()

{

for (int a=0; a<100; a++) {

colorR[a] = random(255);

colorG[a] = random(255);

colorB[a] = random(255);

}

size(1374, 750, OPENGL); //ukuran window

image\_();

cp5 = new ControlP5(this);

ik1 = new InverseKinematic(ps, pe);

font9 = createFont("Arial Bold", 9, false);

font10 = createFont("Arial Bold", 10, false);

font12 = createFont("Arial Bold", 12, false);

font14 = createFont("Arial Bold", 14, false);

font18 = createFont("Arial Bold", 18, false);

font20 = createFont("Arial Bold", 20, false);

font25 = createFont("Arial Bold", 25, false);

font30 = createFont("Arial Bold", 30, false);

font35 = createFont("Arial Bold", 35, false);

GUI\_setup();

setup\_UART();

sobj();

}

float koorx=0;

float koory=90;

int cx, cy;

float r2;

int koorx1, koory1;

void mouseClicked() {

r= sqrt(pow(mouseX-680, 2)+pow(mouseY-545, 2));

r2= sqrt(pow(mouseX-540, 2)+pow(mouseY-545, 2));

if (r<600/2 && r > 351/2 && mouseY <545) {

koorx1=mouseX;

koory1=mouseY;

koorx=int(map(mouseX, 381, 979, -660, 660));

koory=int(map(mouseY, 246, 546, 660, 0));

}

if(r2<225&& r2> 115/\*&& mouseY <545\*/) {

if ((mouseX<=X\_rev+130+500-150+525) && (mouseX>=X\_rev+130-150)) {

if ((mouseY<=Y\_rev+500) && (mouseY>=Y\_rev)) {

count\_click+=1;

clickX[count\_click] = int((mouseX -430+150-525/2)/2.5);

clickY[count\_click] = int(-(mouseY -370-525/3)/2.5);

cx=clickX[count\_click];

cy=clickY[count\_click];

}}}}

float v22=0.0;

int v33, v44;

void draw()

{cp5.getController("RESET\_KOORDINAT").moveTo("fiture");

cp5.getController("RUN").moveTo("fiture");

cp5.getController("v22").moveTo("fiture");

cp5.getController("v33").moveTo("fiture");

cp5.getController("v44").moveTo("fiture");

//println(mouseX, mouseY);

if (statustab==1) //TAB MAIN

{

background(bg);

obj();

images();

fw();

Send\_To\_Arduino();

}

if (statustab==2) //TAB CHART

{

background(bg); //0-->Black

fw();

fill(255);

imageMode(CENTER);

pushMatrix();

scale(0.45);

image(workspace, 1187, 948);

popMatrix();

image(judul, width/2, 45);

draw\_coordinat\_target();

draw\_coordinat();

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

import processing.serial.Serial; // serial library

import controlP5.\*; // controlP5 library

Serial serial;

ControlP5 cp5;

InverseKinematic ik1;

PFont font9, font10, font12, font14, font18, font20, font25, font30, font35;

Chart current\_chart, respon\_chart;

color yellow\_ = color(200, 200, 20), green\_ = color(46, 209, 2), red\_ = color(120, 30, 30), blue\_ = color (0, 102, 200);

color white\_ = color(255, 255, 255), black\_ = color(0, 0, 0), pink\_ = color(255, 0, 255);

//UART Variable

int serial\_conect = 0;

int commListMax;

int[] data = null;

Textlabel txtlblWhichcom;

ListBox commListbox;

ListBox portlist;

// coded by Eberhard Rensch

// Truncates a long port name for better (readable) display in the GUI

String shortifyPortName(String portName, int maxlen)

{

String shortName = portName;

if (shortName.startsWith("/dev/")) shortName = shortName.substring(5);

if (shortName.startsWith("tty.")) shortName = shortName.substring(4); // get rid off leading tty. part of device name

if (portName.length()>maxlen) shortName = shortName.substring(0, (maxlen-1)/2) + "~" +shortName.substring(shortName.length()-(maxlen-(maxlen-1)/2));

if (shortName.startsWith("cu.")) shortName = "";// only collect the corresponding tty. devices

return shortName;

}

boolean START = false;

boolean MODE = false;

boolean ik = true;

boolean P1 = false;

boolean P2 = false;

int KoordinatX, KoordinatY;

int KoordinatX\_, KoordinatY\_;

float fs, fe, fw, rs, re, rw, rp1, rp2;

int ps=380;

int pe=280;

float beta, gamma;

int Ibeta, Igamma;

int cb1, cb2;

float cb3;

int mosxe, mosye, mosex, mosey;

String textValue = "";

float KP;

float KI;

float KD;

int \_KoordinatX, \_KoordinatY;

int mosx, mosy;

int s=5;

//revision v1

int count\_click = 0;

int[] clickX = new int[100];

int[] clickY = new int[100];

int X\_rev = 50;

int Y\_rev = 120;

int statustab=1;

int[] X\_odometry = new int[5];

int[] Y\_odometry = new int[5];

int[][] color\_point = new int[6][3];

float[][] coordinat\_input= new float[6][2];

int []fxcoordinat=new int[100];

int []fycoordinat=new int[100];

void setup()

{

for (int a=0; a<100; a++) {

colorR[a] = random(255);

colorG[a] = random(255);

colorB[a] = random(255);

}

size(1374, 750, OPENGL); //ukuran window

image\_();

cp5 = new ControlP5(this);

ik1 = new InverseKinematic(ps, pe);

font9 = createFont("Arial Bold", 9, false);

font10 = createFont("Arial Bold", 10, false);

font12 = createFont("Arial Bold", 12, false);

font14 = createFont("Arial Bold", 14, false);

font18 = createFont("Arial Bold", 18, false);

font20 = createFont("Arial Bold", 20, false);

font25 = createFont("Arial Bold", 25, false);

font30 = createFont("Arial Bold", 30, false);

font35 = createFont("Arial Bold", 35, false);

GUI\_setup();

setup\_UART();

sobj();

}

float koorx=0;

float koory=90;

int cx, cy;

float r2;

int koorx1, koory1;

void mouseClicked() {

r= sqrt(pow(mouseX-680, 2)+pow(mouseY-545, 2));

r2= sqrt(pow(mouseX-540, 2)+pow(mouseY-545, 2));

if (r<600/2 && r > 351/2 && mouseY <545) {

koorx1=mouseX;

koory1=mouseY;

koorx=int(map(mouseX, 381, 979, -660, 660));

koory=int(map(mouseY, 246, 546, 660, 0));

}

if(r2<225&& r2> 115/\*&& mouseY <545\*/) {

if ((mouseX<=X\_rev+130+500-150+525) && (mouseX>=X\_rev+130-150)) {

if ((mouseY<=Y\_rev+500) && (mouseY>=Y\_rev)) {

count\_click+=1;

clickX[count\_click] = int((mouseX -430+150-525/2)/2.5);

clickY[count\_click] = int(-(mouseY -370-525/3)/2.5);

cx=clickX[count\_click];

cy=clickY[count\_click];

}}}}

float v22=0.0;

int v33, v44;

void draw()

{cp5.getController("RESET\_KOORDINAT").moveTo("fiture");

cp5.getController("RUN").moveTo("fiture");

cp5.getController("v22").moveTo("fiture");

cp5.getController("v33").moveTo("fiture");

cp5.getController("v44").moveTo("fiture");

//println(mouseX, mouseY);

if (statustab==1) //TAB MAIN

{

background(bg);

obj();

images();

fw();

Send\_To\_Arduino();

}

if (statustab==2) //TAB CHART

{

background(bg); //0-->Black

fw();

fill(255);

imageMode(CENTER);

pushMatrix();

scale(0.45);

image(workspace, 1187, 948);

popMatrix();

image(judul, width/2, 45);

draw\_coordinat\_target();

draw\_coordinat();

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

boolean START = false;

boolean MODE = false;

boolean ik = true;

boolean P1 = false;

boolean P2 = false;

int KoordinatX, KoordinatY;

int KoordinatX\_, KoordinatY\_;

float fs, fe, fw, rs, re, rw, rp1, rp2;

int ps=380;

int pe=280;

float beta, gamma;

int Ibeta, Igamma;

int cb1, cb2;

float cb3;

int mosxe, mosye, mosex, mosey;

String textValue = "";

float KP;

float KI;

float KD;

int \_KoordinatX, \_KoordinatY;

int mosx, mosy;

int s=5;

//revision v1

int count\_click = 0;

int[] clickX = new int[100];

int[] clickY = new int[100];

int X\_rev = 50;

int Y\_rev = 120;

int statustab=1;

int[] X\_odometry = new int[5];

int[] Y\_odometry = new int[5];

int[][] color\_point = new int[6][3];

float[][] coordinat\_input= new float[6][2];

int []fxcoordinat=new int[100];

int []fycoordinat=new int[100];

void setup()

{

for (int a=0; a<100; a++) {

colorR[a] = random(255);

colorG[a] = random(255);

colorB[a] = random(255);

}

size(1374, 750, OPENGL); //ukuran window

image\_();

cp5 = new ControlP5(this);

ik1 = new InverseKinematic(ps, pe);

font9 = createFont("Arial Bold", 9, false);

font10 = createFont("Arial Bold", 10, false);

font12 = createFont("Arial Bold", 12, false);

font14 = createFont("Arial Bold", 14, false);

font18 = createFont("Arial Bold", 18, false);

font20 = createFont("Arial Bold", 20, false);

font25 = createFont("Arial Bold", 25, false);

font30 = createFont("Arial Bold", 30, false);

font35 = createFont("Arial Bold", 35, false);

GUI\_setup();

setup\_UART();

sobj();

}

float koorx=0;

float koory=90;

int cx, cy;

float r2;

int koorx1, koory1;

void mouseClicked() {

r= sqrt(pow(mouseX-680, 2)+pow(mouseY-545, 2));

r2= sqrt(pow(mouseX-540, 2)+pow(mouseY-545, 2));

if (r<600/2 && r > 351/2 && mouseY <545) {

koorx1=mouseX;

koory1=mouseY;

koorx=int(map(mouseX, 381, 979, -660, 660));

koory=int(map(mouseY, 246, 546, 660, 0));

}

if(r2<225&& r2> 115/\*&& mouseY <545\*/) {

if ((mouseX<=X\_rev+130+500-150+525) && (mouseX>=X\_rev+130-150)) {

if ((mouseY<=Y\_rev+500) && (mouseY>=Y\_rev)) {

count\_click+=1;

clickX[count\_click] = int((mouseX -430+150-525/2)/2.5);

clickY[count\_click] = int(-(mouseY -370-525/3)/2.5);

cx=clickX[count\_click];

cy=clickY[count\_click];

}}}}

float v22=0.0;

int v33, v44;

void draw()

{cp5.getController("RESET\_KOORDINAT").moveTo("fiture");

cp5.getController("RUN").moveTo("fiture");

cp5.getController("v22").moveTo("fiture");

cp5.getController("v33").moveTo("fiture");

cp5.getController("v44").moveTo("fiture");

//println(mouseX, mouseY);

if (statustab==1) //TAB MAIN

{

background(bg);

obj();

images();

fw();

Send\_To\_Arduino();

}

if (statustab==2) //TAB CHART

{

background(bg); //0-->Black

fw();

fill(255);

imageMode(CENTER);

pushMatrix();

scale(0.45);

image(workspace, 1187, 948);

popMatrix();

image(judul, width/2, 45);

draw\_coordinat\_target();

draw\_coordinat();

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

int statustab=1;

int[] X\_odometry = new int[5];

int[] Y\_odometry = new int[5];

int[][] color\_point = new int[6][3];

float[][] coordinat\_input= new float[6][2];

int []fxcoordinat=new int[100];

int []fycoordinat=new int[100];

void setup()

{

for (int a=0; a<100; a++) {

colorR[a] = random(255);

colorG[a] = random(255);

colorB[a] = random(255);

}

size(1374, 750, OPENGL); //ukuran window

image\_();

cp5 = new ControlP5(this);

ik1 = new InverseKinematic(ps, pe);

font9 = createFont("Arial Bold", 9, false);

font10 = createFont("Arial Bold", 10, false);

font12 = createFont("Arial Bold", 12, false);

font14 = createFont("Arial Bold", 14, false);

font18 = createFont("Arial Bold", 18, false);

font20 = createFont("Arial Bold", 20, false);

font25 = createFont("Arial Bold", 25, false);

font30 = createFont("Arial Bold", 30, false);

font35 = createFont("Arial Bold", 35, false);

GUI\_setup();

setup\_UART();

sobj();

}

float koorx=0;

float koory=90;

int cx, cy;

float r2;

int koorx1, koory1;

void mouseClicked() {

r= sqrt(pow(mouseX-680, 2)+pow(mouseY-545, 2));

r2= sqrt(pow(mouseX-540, 2)+pow(mouseY-545, 2));

if (r<600/2 && r > 351/2 && mouseY <545) {

koorx1=mouseX;

koory1=mouseY;

koorx=int(map(mouseX, 381, 979, -660, 660));

koory=int(map(mouseY, 246, 546, 660, 0));

}

if(r2<225&& r2> 115/\*&& mouseY <545\*/) {

if ((mouseX<=X\_rev+130+500-150+525) && (mouseX>=X\_rev+130-150)) {

if ((mouseY<=Y\_rev+500) && (mouseY>=Y\_rev)) {

count\_click+=1;

clickX[count\_click] = int((mouseX -430+150-525/2)/2.5);

clickY[count\_click] = int(-(mouseY -370-525/3)/2.5);

cx=clickX[count\_click];

cy=clickY[count\_click];

}}}}

float v22=0.0;

int v33, v44;

void draw()

{cp5.getController("RESET\_KOORDINAT").moveTo("fiture");

cp5.getController("RUN").moveTo("fiture");

cp5.getController("v22").moveTo("fiture");

cp5.getController("v33").moveTo("fiture");

cp5.getController("v44").moveTo("fiture");

//println(mouseX, mouseY);

if (statustab==1) //TAB MAIN

{

background(bg);

obj();

images();

fw();

Send\_To\_Arduino();

}

if (statustab==2) //TAB CHART

{

background(bg); //0-->Black

fw();

fill(255);

imageMode(CENTER);

pushMatrix();

scale(0.45);

image(workspace, 1187, 948);

popMatrix();

image(judul, width/2, 45);

draw\_coordinat\_target();

draw\_coordinat();

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

float v22=0.0;

int v33, v44;

void draw()

{cp5.getController("RESET\_KOORDINAT").moveTo("fiture");

cp5.getController("RUN").moveTo("fiture");

cp5.getController("v22").moveTo("fiture");

cp5.getController("v33").moveTo("fiture");

cp5.getController("v44").moveTo("fiture");

//println(mouseX, mouseY);

if (statustab==1) //TAB MAIN

{

background(bg);

obj();

images();

fw();

Send\_To\_Arduino();

}

if (statustab==2) //TAB CHART

{

background(bg); //0-->Black

fw();

fill(255);

imageMode(CENTER);

pushMatrix();

scale(0.45);

image(workspace, 1187, 948);

popMatrix();

image(judul, width/2, 45);

draw\_coordinat\_target();

draw\_coordinat();

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

fiture\_();

pushMatrix();

translate(width/8,0,0);

// obj1();

popMatrix();

// Send\_To\_Arduino();

}

//print(mouseX); print("\t"); println(mouseY);

}

public void setkp(String theText) {

KP= float(theText)\*100;//5

print(KP);

}

public void setki(String theText) {

KI= float(theText)\*100;//0.001

print(KI);

}

public void setkd(String theText) {

KD= float(theText)\*100;//10

println(KD);

}

public void KoordinatX(String Xkoor) {

\_KoordinatX= int(Xkoor);

}

public void KoordinatY(String Ykoor) {

\_KoordinatY= int(Ykoor);

}

void keyPressed() {

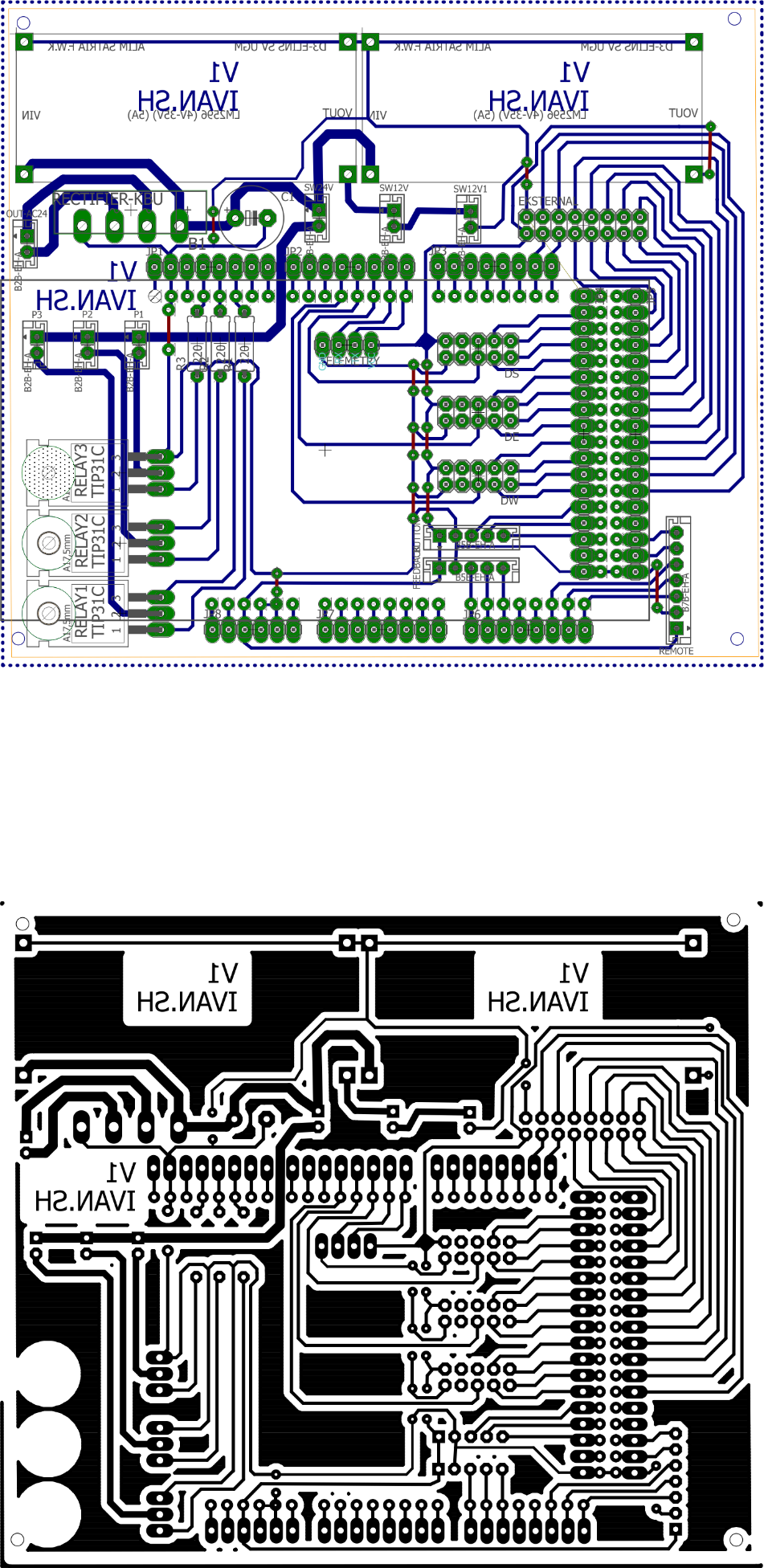
if(keyCode==TAB) {

cp5.getTab("fiture").bringToFront();

}

}

**Desain Eagle SCARA Serpent**



**Foto SCARA Serpent**

