OdometerData

---

double x
double y
double theta

int numberOfIntances
int MAX_INSTANCES

Lock lock

boolean isReseting

Condition doneReseting

OdometerData odoData

---

public synchronized static OdometerData getOdometerData()
throws OdometerExceptions

public double[] getXYT()
public void update(double x, double y, double theta)
public void setXYT(double x, double y , double theta)

public void setX(double x)
public void setY(double y)
public void setTheta(double theta)

OdometerExceptions
extends Exception

Odometer
extends OdometerData
implements Runnable

OdometerData odoData
Odometer odo

EV3LargeRegulatedMotor leftMotor
EV3LargeRegulatedMotor rightMotor

int leftMotorTachoCount
int rightMotorTachoCount

double TRACK
double WHEEL_RAD
double distanceLeft;
double distanceRight
double leftMotorLastTachoCount;
double rightMotorLastTachoCount
double deltaD
double deltaT
double Theta
double dX
double dY

long ODOMETER_PERIOD

---

public synchronized static Odometer
getOdometer(EV3LargeRegulatedMotor leftMotor,
EV3LargeRegulatedMotor rightMotor,
final double TRACK, final double WHEEL_RAD)

public synchronized static Odometer getOdometer()
throws OdometerExceptions

public void run()

Wifi

String SERVER_IP

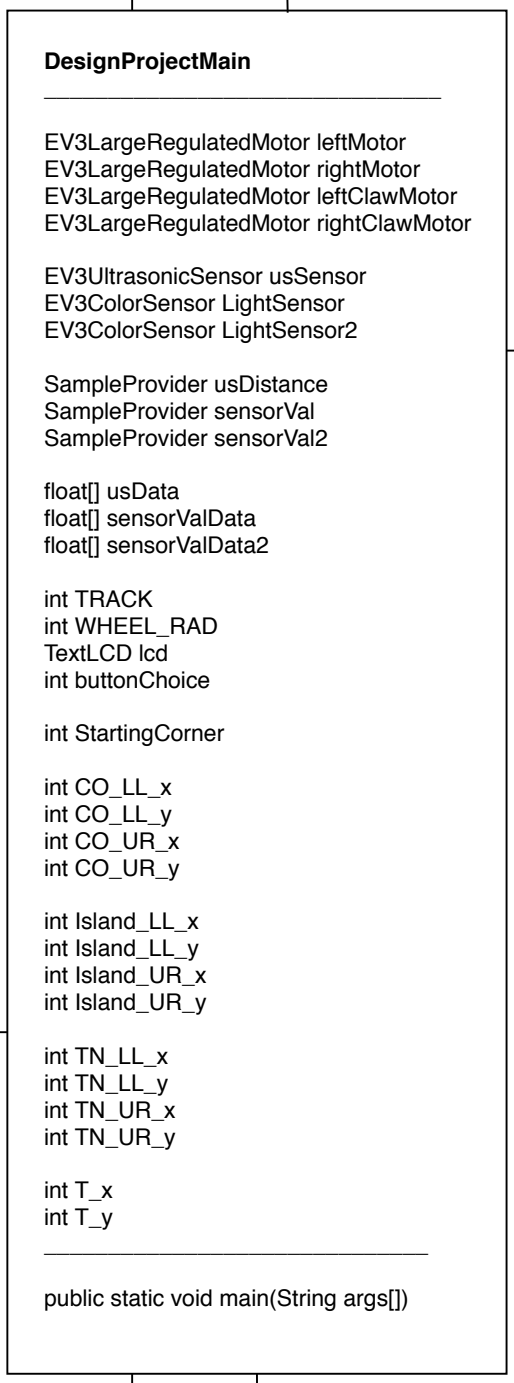
int TEAM_NUMBER

boolean dataReceived
boolean ENABLE_DEBUG_WIFI_PRINT

---

public static void main(String[] args)


Localization
implements Runnable
Odometer odometer
SampleProvider usDistance
SampleProvider sensorVal
SampleProvider sensorVal2
float[] usData
float[] sensorValData
float[] sensorValData2
int track
int wheel_rad
int ROTATE_SPEED
int FORWARD_SPEED
int fullRotation
int halfRotation
int DISTANCE_RANGE
int sensorOffset
boolean keepGoing
double lineDetection
double currentCoordinates
double angleBeta
double currentAngle
double finalAngle
double initialAngle
double firstFallingEdge
double secondFallingEdge
double previousReading []
public void run()
private void usLocalization()
private void lsLocalization()
void turnTo(double theta)
private static int convertDistance(double radius, double distance)
private static int convertAngle(double radius, double width, double angle)
private void setSpeed(int wheelSpeed)
private void reinitializePreviousReadings()
private void updatePreviousReadings(int distance)
private void motorsForward()
private void stopMotors()
private void sensorOffsetCorrection()
private void lineDetection()
private double fallingEdgeDetection(int distance)


DesignProjectMain
EV3LargeRegulatedMotor leftMotor
EV3LargeRegulatedMotor rightMotor
EV3LargeRegulatedMotor leftClawMotor
EV3LargeRegulatedMotor rightClawMotor
EV3UltrasonicSensor usSensor
EV3ColorSensor LightSensor
EV3ColorSensor LightSensor2
SampleProvider usDistance
SampleProvider sensorVal
SampleProvider sensorVal2
float[] usData
float[] sensorValData
float[] sensorValData2
int TRACK
int WHEEL_RAD
TextLCD lcd
int buttonChoice
int StartingCorner
int CO_LL_x
int CO_LL_y
int CO_UR_x
int CO_UR_y
int Island_LL_x
int Island_LL_y
int Island_UR_x
int Island_UR_y
int TN_LL_x
int TN_LL_y
int TN_UR_x
int TN_UR_y
int T_x
int T_y
public static void main(String args[])

TravelToTree
implements Runnable

SampleProvider sensorVal
SampleProvider sensorVal2

float[] sensorValData
float[] sensorValData2

int horizontal
int vertical
int bridgeOrientation
int FORWARD_SPEED
int ROTATE_SPEED
int StartingCorner
int BR_LL_x
int BR_LL_y
int BR_UR_x
int BR_UR_y
int T_x
int T_y

double xDistance
double yDistance
double angleBeta
double distanceToTravel
double finalAngle
double currentCoordinates []
double currentXposition
double currentYposition
double currentAngle
double WHEEL_RAD
double TRACK
double TILE_SIZE
double HALF_TILE
double BRIDGE_CROSSING

---

public void run()

private void travelToBridge()

private int bridgeOrientation()

private void travelThroughBridge()

private void travelToRingSet()

private void turnToTree(double x, double y)

private void travelTo(double x, double y)

private void turnTo(double theta)

private static int convertDistance(double radius, double distance)

private static int convertAngle(double radius, double width, double angle)

private void setSpeed(int wheelSpeed)

RingRetrieval
implements Runnable

Odometer odometer

EV3LargeRegulatedMotor leftClawMotor
EV3LargeRegulatedMotor rightClawMotor

Port lsPort
EV3ColorSensor lightSensor
SampleProvider colorSensor
float[] rgb

double wheelRad
double track
double retrievalDistance

double blueRingRedMean
double blueRingGreenMean
double blueRingBlueMean

double greenRingRedMean
double greenRingGreenMean
double greenRingBlueMean

double yellowRingRedMean
double yellowRingGreenMean
double yellowRingBlueMean

double orangeRingRedMean
double orangeRingGreenMean
double orangeRingBlueMean

double redSample
double greenSample
double blueSample
double normalizedRedSample
double normalizedGreenSample
double normalizedBlueSample

double blueRingDetector
double greenRingDetector
double yellowRingDetector
double orangeRingDetector
double detectionRange

int clawControl
int ROTATE_SPEED

---

public void run()

private void reachRingSet()

private static int convertDistance(double radius, double distance)

private void ringGrab()

private void setClawSpeed(int wheelSpeed)

private void colorIdentification()

Author: Maxime Cardinal
Date: October 29, 2018
Version: 1.1
Edit:
1. Maxime Cardinal-Oct. 22-
Creation of the preminilary diagram
1.1 Maxime Cardinal-Nov. 12-
Modification of the general layout and the Localization class, and addition of the Wifi, TravelToTree and RingRetrieval classes