#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

task main()

{

// Table of Contents

int defaultMotorSpeed = 20;

int defaultReducedSpeed = 2;

int turningSpeed = 30;

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//Default speed values

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//\*\*\*Move block out of way

if(getUSDistance(TouchSensor)<=15)

{

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//down

setMotorTarget(crane, -180, 30);

//Move it out of way

setMotorSpeed(motorA,0);

setMotorSpeed(motorB, turningSpeed);

wait1Msec(1000);

//up

//setMotorTarget(crane, 180, 100);

}

// If the color reflected is white

if (SensorValue[ColorSensor] >= 20) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

}

// Else it sees black

else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}

#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

/\* TO DO:

1)MAKE CRANE MOVE SHIT OUT OF THE WAY

2)CONFIGURE THE VALUES FOR THE BEST RESULT

3)CLEAN UP CODE, ADD UNDERSTANDABLE COMMENTS

\*/

task main()

{

int defaultMotorSpeed = 20;

int defaultReducedSpeed = 2;

int turningSpeed = 30;

//1) RESET THE CRANE

setMotorTarget(crane, 180, 200);

//2) ACTIVATION OF ROBOT

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//2.1) DRIVE FORWARD

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//3) Move block out of way, CHANGE DISTANCE WHERE IT JUST REACHES THE HOOK OR INCREASE HOOK LENGTH WITH A HOOK SHAPE

if(getUSDistance(TouchSensor)<=15)

{

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//MOVE CRANE DOWN

setMotorTarget(crane, -180, 200);

//Move it out of way,MOVE IT TO SIDE, MIGHT HAVE TO MAKE IT MOVE BACK THEN TO THE SIDE WITH THE HOOKED CRANE

setMotorSpeed(motorA,0);

setMotorSpeed(motorB, turningSpeed);

//MOVE CRANE UP

setMotorTarget(crane, 180, 200);

}

//2.2) If the color reflected is white, TURN RIGHT

if (SensorValue[ColorSensor] >= 20) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

}

//2.3) Else it sees black, TURN LEFT

else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}

#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

/\* TO DO:

1)MAKE CRANE MOVE SHIT OUT OF THE WAY

2)CONFIGURE THE VALUES FOR THE BEST RESULT

3)CLEAN UP CODE, ADD UNDERSTANDABLE COMMENTS

\*/

task main()

{

int defaultMotorSpeed = 20;

int defaultReducedSpeed = 2;

int turningSpeed = 30;

int backwardSpeed = -20;

int whiteColor = 20;

int distanceDetect = 15;

int timeReturnPos = 2000;

//1)RESET THE CRANE

setMotorTarget(crane, 180, 200);

//2)ACTIVATION OF ROBOT

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//2.1)DRIVE FORWARD

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//3)Move block out of way, CHANGE DISTANCE WHERE IT JUST REACHES THE HOOK OR INCREASE HOOK LENGTH WITH A HOOK SHAPE

if(getUSDistance(TouchSensor) <= distanceDetect) {

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//MOVE CRANE DOWN

setMotorTarget(crane, -180, 200);

//Move it out of way, MOVE IT TO SIDE, MIGHT HAVE TO MAKE IT MOVE BACK THEN TO THE SIDE WITH THE HOOKED CRANE

setMotorSpeed(motorA, backwardSpeed);

setMotorSpeed(motorB, turningSpeed);

wait1Msec = timeReturnPos;

//MOVE CRANE UP

setMotorTarget(crane, 180, 200);

//Move the robot back onto the track.

setMotorSpeed(motorA, turningSpeed));

setMotorSpeed(motorB, backwardSpeed);

wait1Msec = timeReturnPos;

//2.2)If the color reflected is white, TURN RIGHT

if (SensorValue[ColorSensor] >= whiteColor) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

//2.3)Else it sees black, TURN LEFT

} else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}

#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

/\* TO DO:

1)MAKE CRANE MOVE SHIT OUT OF THE WAY

2)CONFIGURE THE VALUES FOR THE BEST RESULT

3)CLEAN UP CODE, ADD UNDERSTANDABLE COMMENTS

\*/

task main()

{

int defaultMotorSpeed = 20;

int defaultReducedSpeed = 2;

int turningSpeed = 30;

int backwardSpeed = -20;

int whiteColor = 20;

int distanceDetect = 15;

int timeReturnPos = 1000;

//1)RESET THE CRANE

setMotorTarget(crane, 180, 200);

//2)ACTIVATION OF ROBOT

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//2.1)DRIVE FORWARD

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//3)Move block out of way, CHANGE DISTANCE WHERE IT JUST REACHES THE HOOK OR INCREASE HOOK LENGTH WITH A HOOK SHAPE

if(getUSDistance(TouchSensor) <= distanceDetect) {

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//MOVE CRANE DOWN

setMotorTarget(crane, -180, 200);

//Move it out of way, MOVE IT TO SIDE, MIGHT HAVE TO MAKE IT MOVE BACK THEN TO THE SIDE WITH THE HOOKED CRANE

setMotorSpeed(motorA, backwardSpeed);

setMotorSpeed(motorB, turningSpeed);

wait1Msec(timeReturnPos);

//MOVE CRANE UP

setMotorTarget(crane, 180, 200);

//Move the robot back onto the track.

setMotorSpeed(motorA, turningSpeed);

setMotorSpeed(motorB, backwardSpeed);

wait1Msec(timeReturnPos);

break;

//2.2)If the color reflected is white, TURN RIGHT

if (SensorValue[ColorSensor] >= whiteColor) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

//2.3)Else it sees black, TURN LEFT

} else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}

#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

/\* TO DO:

1)MAKE CRANE MOVE SHIT OUT OF THE WAY

2)CONFIGURE THE VALUES FOR THE BEST RESULT

3)CLEAN UP CODE, ADD UNDERSTANDABLE COMMENTS

\*/

task main()

{

int defaultMotorSpeed = 20;

int defaultReducedSpeed = 2;

int turningSpeed = 30;

int backwardSpeed = -20;

int whiteColor = 20;

int distanceDetect = 15;

int timeReturnPos = 1000;

//1)RESET THE CRANE

setMotorTarget(crane, 180, 200);

//2)ACTIVATION OF ROBOT

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//2.1)DRIVE FORWARD

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//3)Move block out of way, CHANGE DISTANCE WHERE IT JUST REACHES THE HOOK OR INCREASE HOOK LENGTH WITH A HOOK SHAPE

if(getUSDistance(TouchSensor) <= distanceDetect) {

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//MOVE CRANE DOWN

setMotorTarget(crane, -180, 200);

//Move it out of way, MOVE IT TO SIDE, MIGHT HAVE TO MAKE IT MOVE BACK THEN TO THE SIDE WITH THE HOOKED CRANE

setMotorSpeed(motorA, backwardSpeed);

setMotorSpeed(motorB, turningSpeed);

wait1Msec(timeReturnPos);

//MOVE CRANE UP

setMotorTarget(crane, 180, 200);

//Move the robot back onto the track.

setMotorSpeed(motorA, turningSpeed);

setMotorSpeed(motorB, backwardSpeed);

wait1Msec(timeReturnPos);

break;

}

//2.2)If the color reflected is white, TURN RIGHT

if (SensorValue[ColorSensor] >= whiteColor) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

//2.3)Else it sees black, TURN LEFT

} else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}

#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

/\* TO DO:

1)MAKE CRANE MOVE SHIT OUT OF THE WAY

2)CONFIGURE THE VALUES FOR THE BEST RESULT

3)CLEAN UP CODE, ADD UNDERSTANDABLE COMMENTS

\*/

task main()

{

int defaultMotorSpeed = 20;

int defaultReducedSpeed = 2;

int turningSpeed = 30;

int backwardSpeed = -20;

int whiteColor = 20;

int distanceDetect = 15;

int timeReturnPos = 1000;

//1)RESET THE CRANE

setMotorTarget(crane, 180, 200);

//2)ACTIVATION OF ROBOT

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//2.1)DRIVE FORWARD

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//3)Move block out of way, CHANGE DISTANCE WHERE IT JUST REACHES THE HOOK OR INCREASE HOOK LENGTH WITH A HOOK SHAPE

if(getUSDistance(TouchSensor) <= distanceDetect) {

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//MOVE CRANE DOWN

setMotorTarget(crane, -180, 200);

//Move it out of way, MOVE IT TO SIDE, MIGHT HAVE TO MAKE IT MOVE BACK THEN TO THE SIDE WITH THE HOOKED CRANE

setMotorSpeed(motorA, backwardSpeed);

setMotorSpeed(motorB, turningSpeed);

wait1Msec(timeReturnPos);

//MOVE CRANE UP

setMotorTarget(crane, 180, 200);

//Move the robot back onto the track.

setMotorSpeed(motorA, turningSpeed);

setMotorSpeed(motorB, backwardSpeed);

wait1Msec(timeReturnPos);

}

//2.2)If the color reflected is white, TURN RIGHT

if (SensorValue[ColorSensor] >= whiteColor) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

//2.3)Else it sees black, TURN LEFT

} else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}

#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

/\* TO DO:

1)MAKE CRANE MOVE SHIT OUT OF THE WAY

2)CONFIGURE THE VALUES FOR THE BEST RESULT

3)CLEAN UP CODE, ADD UNDERSTANDABLE COMMENTS

\*/

task main()

{

int defaultMotorSpeed = 15;

int defaultReducedSpeed = -25;

int turningSpeed = 55;

int whiteColor = 20;

int distanceDetect = 15;

int timeReturnPos = 1000;

//1)RESET THE CRANE

setMotorTarget(crane, 180, 200);

//2)ACTIVATION OF ROBOT

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//2.1)DRIVE FORWARD

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//3)Move block out of way, CHANGE DISTANCE WHERE IT JUST REACHES THE HOOK OR INCREASE HOOK LENGTH WITH A HOOK SHAPE

if(getUSDistance(TouchSensor) <= distanceDetect) {

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//MOVE CAR RIGHT

setMotorSpeed(motorA, turningSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

wait1Msec(timeReturnPos-500);

//MOVE CRANE DOWN

setMotorTarget(crane, -180, 200);

//Move it out of way, MOVE IT TO SIDE, MIGHT HAVE TO MAKE IT MOVE BACK THEN TO THE SIDE WITH THE HOOKED CRANE

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

wait1Msec(timeReturnPos+500);

//MOVE CRANE UP

setMotorTarget(crane, 180, 200);

//Move the robot back onto the track.

setMotorSpeed(motorA, turningSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

wait1Msec(timeReturnPos);

}

//2.2)If the color reflected is white, TURN RIGHT

if (SensorValue[ColorSensor] >= whiteColor) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

//2.3)Else it sees black, TURN LEFT

} else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}

#pragma config(Sensor, S1, ColorSensor, sensorEV3\_Color)

#pragma config(Sensor, S2, TouchSensor, sensorEV3\_Ultrasonic)

#pragma config(Motor, motorA, LeftWheel, tmotorEV3\_Large, PIDControl, driveLeft, encoder)

#pragma config(Motor, motorB, RightWheel, tmotorEV3\_Large, PIDControl, driveRight, encoder)

#pragma config(Motor, motorC, crane, tmotorEV3\_Medium, PIDControl, encoder)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

//!!Code automatically generated by 'ROBOTC' configuration wizard !!

/\* TO DO:

1)MAKE CRANE MOVE SHIT OUT OF THE WAY X

2)CONFIGURE THE VALUES FOR THE BEST RESULT

3)CLEAN UP CODE, ADD UNDERSTANDABLE COMMENTS

\*/

task main()

{

int defaultMotorSpeed = 20;

int defaultReducedSpeed = -25;

int turningSpeed = 25;

int whiteColor = 20;

int distanceDetect = 10;

int timeReturnPos = 1000;

//1)RESET THE CRANE

setMotorTarget(crane, 180, 200);

//2)ACTIVATION OF ROBOT

while(true) {

displayCenteredBigTextLine(5, " %d",SensorValue[ColorSensor]);

//2.1)DRIVE FORWARD

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultMotorSpeed);

//3)Move block out of way, CHANGE DISTANCE WHERE IT JUST REACHES THE HOOK OR INCREASE HOOK LENGTH WITH A HOOK SHAPE

if(getUSDistance(TouchSensor) <= distanceDetect) {

playTone(300, 100);

setMotorSpeed(motorA, 0);

setMotorSpeed(motorB, 0);

//MOVE CAR RIGHT

setMotorSpeed(motorA, turningSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

wait1Msec(timeReturnPos-600);

//MOVE CRANE DOWN

setMotorTarget(crane, -180, 100);

//Move it out of way, MOVE IT TO SIDE, MIGHT HAVE TO MAKE IT MOVE BACK THEN TO THE SIDE WITH THE HOOKED CRANE

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed+40);

wait1Msec(timeReturnPos-200);

//MOVE CRANE UP

wait1Msec(300);

setMotorTarget(crane, 180, 200);

//Move the robot back onto the track.

setMotorSpeed(motorA, turningSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

wait1Msec(timeReturnPos-275);

}

//2.2)If the color reflected is white, TURN RIGHT

if (SensorValue[ColorSensor] >= whiteColor) {

setMotorSpeed(motorA, defaultReducedSpeed);

setMotorSpeed(motorB, turningSpeed);

//2.3)Else it sees black, TURN LEFT

} else {

setMotorSpeed(motorA, defaultMotorSpeed);

setMotorSpeed(motorB, defaultReducedSpeed);

}

}

}