#pragma region VEXcode Generated Robot Configuration

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#include <string.h>

#include "vex.h"

using namespace vex;

// Brain should be defined by default

brain Brain;

// START V5 MACROS

#define waitUntil(condition) \

do { \

wait(5, msec); \

} while (!(condition))

#define repeat(iterations) \

for (int iterator = 0; iterator < iterations; iterator++)

// END V5 MACROS

// Define the motors with your specific setup

motor leftDriveMotor = motor(PORT2, gearSetting::ratio18\_1, false);

motor rightDriveMotor = motor(PORT1, gearSetting::ratio18\_1, true);

motor intakeMotor = motor(PORT8, gearSetting::ratio18\_1, false);

motor conveyorMotor = motor(PORT3, gearSetting::ratio18\_1, false);

motor leftLiftMotor = motor(PORT20, gearSetting::ratio18\_1, false);

motor rightLiftMotor = motor(PORT19, gearSetting::ratio18\_1, false);

// Define the controller

controller Controller1;

// Define the piston

digital\_out piston = digital\_out(Brain.ThreeWirePort.A);

// Competition object

competition Competition;

// Autonomous code

void autonomous() {

intakeMotor.setVelocity(90, percent);

conveyorMotor.setVelocity(80, percent);

conveyorMotor.spin(forward);

intakeMotor.spin(forward);

// Example autonomous sequence

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(1, seconds);

leftDriveMotor.stop();

rightDriveMotor.stop();

piston.set(true);

wait(1, seconds);

piston.set(false);

wait(1, seconds);

// Add more autonomous commands as needed

}

// Driver control code

void driverControl() {

while (true) {

// Drive control

int leftSpeed = Controller1.Axis3.position(percent);

int rightSpeed = Controller1.Axis2.position(percent);

leftDriveMotor.spin(forward, leftSpeed, percent);

rightDriveMotor.spin(forward, rightSpeed, percent);

// Intake control

if (Controller1.ButtonR1.pressing()) {

intakeMotor.spin(forward);

} else if (Controller1.ButtonR2.pressing()) {

intakeMotor.spin(reverse);

} else {

intakeMotor.stop();

}

// Conveyor control

if (Controller1.ButtonL1.pressing()) {

conveyorMotor.spin(forward);

} else if (Controller1.ButtonL2.pressing()) {

conveyorMotor.spin(reverse);

} else {

conveyorMotor.stop();

}

// Lift control

if (Controller1.ButtonUp.pressing()) {

leftLiftMotor.spin(forward);

rightLiftMotor.spin(reverse);

} else if (Controller1.ButtonDown.pressing()) {

leftLiftMotor.spin(reverse);

rightLiftMotor.spin(forward);

} else if (Controller1.ButtonX.pressing()) {

leftLiftMotor.setBrake(brakeType::hold);

rightLiftMotor.setBrake(brakeType::hold);

leftLiftMotor.stop();

rightLiftMotor.stop();

} else {

leftLiftMotor.setBrake(brakeType::coast);

rightLiftMotor.setBrake(brakeType::coast);

leftLiftMotor.stop();

rightLiftMotor.stop();

}

// Grab piston control

if (Controller1.ButtonA.pressing()) {

piston.set(true);

} else if (Controller1.ButtonB.pressing()) {

piston.set(false);

}

task::sleep(20);

}

}

// Main function

int main() {

// Initializing Robot Configuration

vexcodeInit();

// Set up callbacks for autonomous and driver control periods

Competition.autonomous(autonomous);

Competition.drivercontrol(driverControl);

// Prevent main from exiting

while (true) {

task::sleep(100);

}

}

—-------------------

#pragma region VEXcode Generated Robot Configuration

// Make sure all required headers are included.

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#include <string.h>

#include "vex.h"

using namespace vex;

// Brain should be defined by default

brain Brain;

// START V5 MACROS

#define waitUntil(condition) \

do { \

wait(5, msec); \

} while (!(condition))

#define repeat(iterations) \

for (int iterator = 0; iterator < iterations; iterator++)

// END V5 MACROS

// Robot configuration code.

// generating and setting random seed

void initializeRandomSeed(){

int systemTime = Brain.Timer.systemHighResolution();

double batteryCurrent = Brain.Battery.current();

double batteryVoltage = Brain.Battery.voltage(voltageUnits::mV);

// Combine these values into a single integer

int seed = int(batteryVoltage + batteryCurrent \* 100) + systemTime;

// Set the seed

srand(seed);

}

void vexcodeInit() {

//Initializing random seed.

initializeRandomSeed();

}

// Helper to make playing sounds from the V5 in VEXcode easier and

// keeps the code cleaner by making it clear what is happening.

void playVexcodeSound(const char \*soundName) {

printf("VEXPlaySound:%s\n", soundName);

wait(5, msec);

}

#pragma endregion VEXcode Generated Robot Configuration

#include "vex.h"

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#include <string.h>

using namespace vex;

// Brain should be defined by default

// START V5 MACROS

#define waitUntil(condition) \

do { \

wait(5, msec); \

} while (!(condition))

#define repeat(iterations) \

for (int iterator = 0; iterator < iterations; iterator++)

// END V5 MACROS

// Define the motors with your specific setup

motor leftDriveMotor = motor(PORT2, gearSetting::ratio18\_1, false);

motor rightDriveMotor = motor(PORT1, gearSetting::ratio18\_1, true);

motor intakeMotor = motor(PORT8, gearSetting::ratio18\_1, false);

motor conveyorMotor = motor(PORT3, gearSetting::ratio18\_1, false);

motor leftLiftMotor = motor(PORT20, gearSetting::ratio18\_1, false);

motor rightLiftMotor = motor(PORT19, gearSetting::ratio18\_1, false);

// Define the controller

controller Controller1;

// Define the grab piston

digital\_out GrabPiston = digital\_out(Brain.ThreeWirePort.A);

int main() {

// Initializing Robot Configuration. DO NOT REMOVE!

vexcodeInit();

intakeMotor.setVelocity(90, percent);

conveyorMotor.setVelocity(80, percent);

while (true) {

// Drive control

int leftSpeed = Controller1.Axis3.position(percent);

int rightSpeed = Controller1.Axis2.position(percent);

leftDriveMotor.spin(forward, leftSpeed, percent);

rightDriveMotor.spin(forward, rightSpeed, percent);

// Intake control

if (Controller1.ButtonR1.pressing()) {

intakeMotor.spin(forward);

} else if (Controller1.ButtonR2.pressing()) {

intakeMotor.spin(reverse);

} else {

intakeMotor.stop();

}

// Conveyor control

if (Controller1.ButtonL1.pressing()) {

conveyorMotor.spin(forward);

} else if (Controller1.ButtonL2.pressing()) {

conveyorMotor.spin(reverse);

} else {

conveyorMotor.stop();

}

// Lift control

if (Controller1.ButtonUp.pressing()) {

leftLiftMotor.spin(forward);

rightLiftMotor.spin(reverse);

} else if (Controller1.ButtonDown.pressing()) {

leftLiftMotor.spin(reverse);

rightLiftMotor.spin(forward);

} else if (Controller1.ButtonX.pressing()) { // Lock the lift motors with Button X

leftLiftMotor.setBrake(brakeType::hold);

rightLiftMotor.setBrake(brakeType::hold);

leftLiftMotor.stop();

rightLiftMotor.stop();

} else {

leftLiftMotor.setBrake(brakeType::coast);

rightLiftMotor.setBrake(brakeType::coast);

leftLiftMotor.stop();

rightLiftMotor.stop();

}

// Grab piston control

if (Controller1.ButtonA.pressing()) { // Grab with piston

GrabPiston.set(true);

} else if (Controller1.ButtonB.pressing()) { // Release with piston

GrabPiston.set(false);

}

// A short delay to prevent wasted resources.

task::sleep(20);

}

}

—---------

#pragma region VEXcode Generated Robot Configuration

// Make sure all required headers are included.

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#include <string.h>

#include "vex.h"

using namespace vex;

// Brain should be defined by default

brain Brain;

// START V5 MACROS

#define waitUntil(condition) \

do { \

wait(5, msec); \

} while (!(condition))

#define repeat(iterations) \

for (int iterator = 0; iterator < iterations; iterator++)

// END V5 MACROS

// Robot configuration code.

// generating and setting random seed

void initializeRandomSeed(){

int systemTime = Brain.Timer.systemHighResolution();

double batteryCurrent = Brain.Battery.current();

double batteryVoltage = Brain.Battery.voltage(voltageUnits::mV);

// Combine these values into a single integer

int seed = int(batteryVoltage + batteryCurrent \* 100) + systemTime;

// Set the seed

srand(seed);

}

void vexcodeInit() {

//Initializing random seed.

initializeRandomSeed();

}

// Helper to make playing sounds from the V5 in VEXcode easier and

// keeps the code cleaner by making it clear what is happening.

void playVexcodeSound(const char \*soundName) {

printf("VEXPlaySound:%s\n", soundName);

wait(5, msec);

}

#pragma endregion VEXcode Generated Robot Configuration

#include "vex.h"

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#include <string.h>

using namespace vex;

// Brain should be defined by default

// START V5 MACROS

#define waitUntil(condition) \

do { \

wait(5, msec); \

} while (!(condition))

#define repeat(iterations) \

for (int iterator = 0; iterator < iterations; iterator++)

// END V5 MACROS

// Define the motors with your specific setup

motor leftDriveMotor = motor(PORT2, gearSetting::ratio18\_1, false);

motor rightDriveMotor = motor(PORT1, gearSetting::ratio18\_1, true);

motor intakeMotor = motor(PORT8, gearSetting::ratio18\_1, false);

motor conveyorMotor = motor(PORT3, gearSetting::ratio18\_1, false);

motor leftLiftMotor = motor(PORT20, gearSetting::ratio18\_1, false);

motor rightLiftMotor = motor(PORT19, gearSetting::ratio18\_1, false);

// Define the controller

controller Controller1;

// Define the piston (assuming it's connected to a digital output port)

digital\_out piston = digital\_out(Brain.ThreeWirePort.A);

void autonomous() {

// Set motor velocities

intakeMotor.setVelocity(90, percent);

conveyorMotor.setVelocity(80, percent);

conveyorMotor.spin(forward);

intakeMotor.spin(forward);

// Forward for 1 second

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(1, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Turn left 90 degrees (assumed 1 second)

leftDriveMotor.spin(reverse, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(1.0, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Backward for 1 second

leftDriveMotor.spin(reverse, 50, percent);

rightDriveMotor.spin(reverse, 50, percent);

wait(1.6, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Grab stake (piston command)

piston.set(true);

wait(1, seconds);

// Turn right 10 degrees (assumed 1 second)

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(reverse, 50, percent);

wait(0.1, seconds);

// Forward for 1 second

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(2.4, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Turn right 90 degrees (assumed 1 second)

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(reverse, 50, percent);

wait(.9, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Forward for 1 second

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(2, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Turn around (180 degrees, assumed 2 seconds)

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(reverse, 50, percent);

wait(1.55, seconds); //1.5

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Forward for 1 second

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(3.5, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Turn left 90 degrees (assumed 1 second)

leftDriveMotor.spin(reverse, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(.65, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Backward for 1 second

leftDriveMotor.spin(reverse, 50, percent);

rightDriveMotor.spin(reverse, 50, percent);

wait(4, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

wait(0.2, seconds); // Small delay

// Drop stake (release piston)

piston.set(false);

wait(1, seconds);

// Turn left 90 degrees (assumed 1 second)

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(reverse, 50, percent);

wait(.5, seconds);

// Forward for 1 second

leftDriveMotor.spin(forward, 50, percent);

rightDriveMotor.spin(forward, 50, percent);

wait(5, seconds);

// Stop motors

leftDriveMotor.stop();

rightDriveMotor.stop();

}

int main() {

// Initializing Robot Configuration. DO NOT REMOVE!

vexcodeInit();

// Run the autonomous sequence

autonomous();

}