

# 15034H VEXcode VR Skills Challenge

Binford Elementary, Bloomington, IN, USA

**VEX IQ**

ROBOTICS

**COMPETITION**

**FULL VOLUME**

**Peter Thies**

# Our Code

This block is our first thing the robot does during the run. It turns the robot towards the first red block, drives into it, and returns to the starting position.

```
define Red #1
  turn right + for 85 degrees +
  drive forward + for 16 inches +
  wait 1 seconds
  drive reverse + for 15 inches +
  turn left + for 87 degrees +
  drive reverse + for 1 inches +
```

This block has the code for the first tower we dump green blocks into, Tower #2. It sucks up the first green block in the row of 4 at the starting position, gets it into the tower, turns around, gets another green, and dumps that in too for the uniform bonus.

```
define Tower #2
  spin IntakeMotorGroup + Intake +
  drive forward + for 20 inches +
  turn left + for 90 degrees +
  stop IntakeMotorGroup +
  spin IntakeMotorGroup + Intake +
  drive forward + for 5 inches +
  turn left + for 90 degrees +
  turn right + for 15 degrees +
  spin IntakeMotorGroup + Intake +
  wait 1.5 seconds
  drive reverse + for 12 inches +
  turn right + for 180 degrees +
  turn right + for 10 degrees +
  spin IntakeMotorGroup + Intake +
  drive forward + for 10 inches +
  turn right + for 180 degrees +
  drive forward + for 22 inches +
  turn left + for 10 degrees +
  spin IntakeMotorGroup + Outtake +
  wait 1 seconds
```

This block of code runs all the different sections in order to keep the code from running crazy! The different sections already have the waits included.

```
define Red #2
  drive reverse + for 12 inches +
  turn left + for 160 degrees +
  spin IntakeMotorGroup + Intake +
  drive forward + for 22 inches +
  turn left + for 70 degrees +
  wait 1 seconds
```

```
define Tower #1 and Partial Park
  drive reverse + for 12 inches +
  turn left + for 180 degrees +
  drive forward + for 12 inches +
  turn left + for 90 degrees +
  spin IntakeMotorGroup + Intake +
  drive forward + for 5 inches +
  turn right + for 30 degrees +
  drive forward + for 50 inches +
  turn left + for 20 degrees +
  drive forward + for 8 inches +
  spin IntakeMotorGroup + Outtake +
  wait 1 seconds
  drive reverse + for 6 inches +
  turn left + for 170 degrees +
  spin IntakeMotorGroup + Intake +
  drive forward + for 10 inches +
  turn right + for 175 degrees +
  wait 1 seconds
  drive forward + for 16 inches +
  spin IntakeMotorGroup + Outtake +
  wait 1 seconds
  turn left + for 110 degrees +
  spin ArmMotorGroup + up + for 120 degrees +
  drive forward + for 30 inches +
  forever
  if FrontOptical + detects red + then
  stop driving
```

```
define Tower #3
  drive forward + for 14 inches +
  spin IntakeMotorGroup + Outtake +
  wait 1 seconds
  drive reverse + for 12 inches +
  turn left + for 220 degrees +
  spin IntakeMotorGroup + Intake +
  drive forward + for 16 inches +
  turn left + for 15 degrees +
  drive forward + for 15 inches +
  drive reverse + for 5 inches +
  turn right + for 180 degrees +
  drive forward + for 20 inches +
  turn right + for 55 degrees +
  drive forward + for 14 inches +
  spin IntakeMotorGroup + Intake +
  wait 1 seconds
```

This section is for the second red block we get. All it does is suck up a green block on the way and bump into a red, and also get lined up for dumping into the tower.

This section is the second to last thing we do. What it does is dump the block that the Red #2 program picked up into Tower #3, and goes back to get the other red and another block, lines up with the tower, and dumps it in.

This is a very important section. What it does is when a block is detected in the claw with the bumper sensor, it waits, and lifts the arm. This helps the program become more efficient and makes the program less complicated. It also saved some time when coding!

This section is very simple. All it does is get the virtual robot up to full speed to be able to do things faster to save time. This is required because the robot's default is to run them at 50%, not 100%.

This is another important section. When the claw registers a block is not in there anymore, it lowers the arm and stops the spinner. This helps make the program very efficient and saves time from waiting for the arm and spinner.

This final section is what I think is THE most important. It turns around from the middle of the field, gets a green block, hightails it to Tower #1, dumps it in, turns around and gets another nearby green block, dumps that in, and turns to partial park, and drives forward to do just that! At the end, it uses the optical sensor to be able to stop the robot when it reaches the supply zone bar, and that means it is parked!

```
when IntakeBumper + pressed +
  wait 1 seconds
  spin ArmMotorGroup + up + for 200 degrees +
```

```
when started
  set drive velocity to 1000 %
  set turn velocity to 100 %
  set IntakeMotorGroup + velocity to 100 % +
  set IntakeMotorGroup + velocity to 100 % +
  This gets the robot up to speed and ready for fast driving!
```

```
when IntakeBumper + released +
  wait 0.5 seconds
  spin ArmMotorGroup + down + for 200 degrees +
  stop IntakeMotorGroup +
```

# How VR Skills Helped Me In Coding

In VEXcode VR, when I was coding, I learned a lot of things that I did not know before. Here is a list of them!

1. Sensors; I have never used sensors before in coding. I noticed that it is very easy to do and makes programs more efficient. I probably should have used sensors in this year's actual robot more.
2. Organizing Code; I normally have a giant jumble of code in my programs. I have noticed that using the "My Block" feature makes it a lot easier to split up code and run programs more efficiently.
3. Coding long programs; Normally, I have been an assistant coder on our team. This is my first year doing VEXcode VR, and it helped me with my overall coding performance, even when working with a real robot.