Using Professional Coding Tools with VEX IQ - Visual Studio Code Extension

Jimmy Lin, PhD
Director of Computer Science Education
VEX Robotics

Agenda

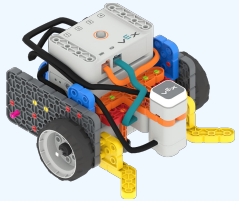
- Introduction to VS Code
- Clawbot – First Part
- Clawbot – Stack Cubes
- Clawbot – Sensors
- Wrap up

The VEX Continuum

VEX 123

Coding Starts Early

Ages 4+



VEX GO

STEM Starts Early

Ages 8+

VEX AIM

Real World Coding

Ages 8+

VEX IQ

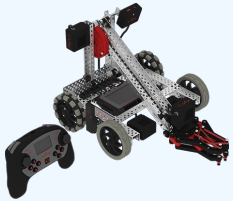
Applied STEM Learning

Ages 11+

VEX EXP

Real World STEM for Classrooms

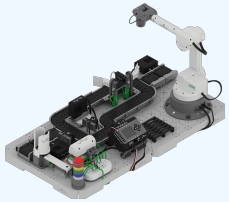
Ages 14+



VEX V5

Real World STEM for Competition

Ages 14+



VEX CTE

Workforce Readiness

Ages 14+

VEX AIR

STEM Skills Take Flight

Ages 14+

VEX CODE VR

Virtual Robot Coding

Ages 8+

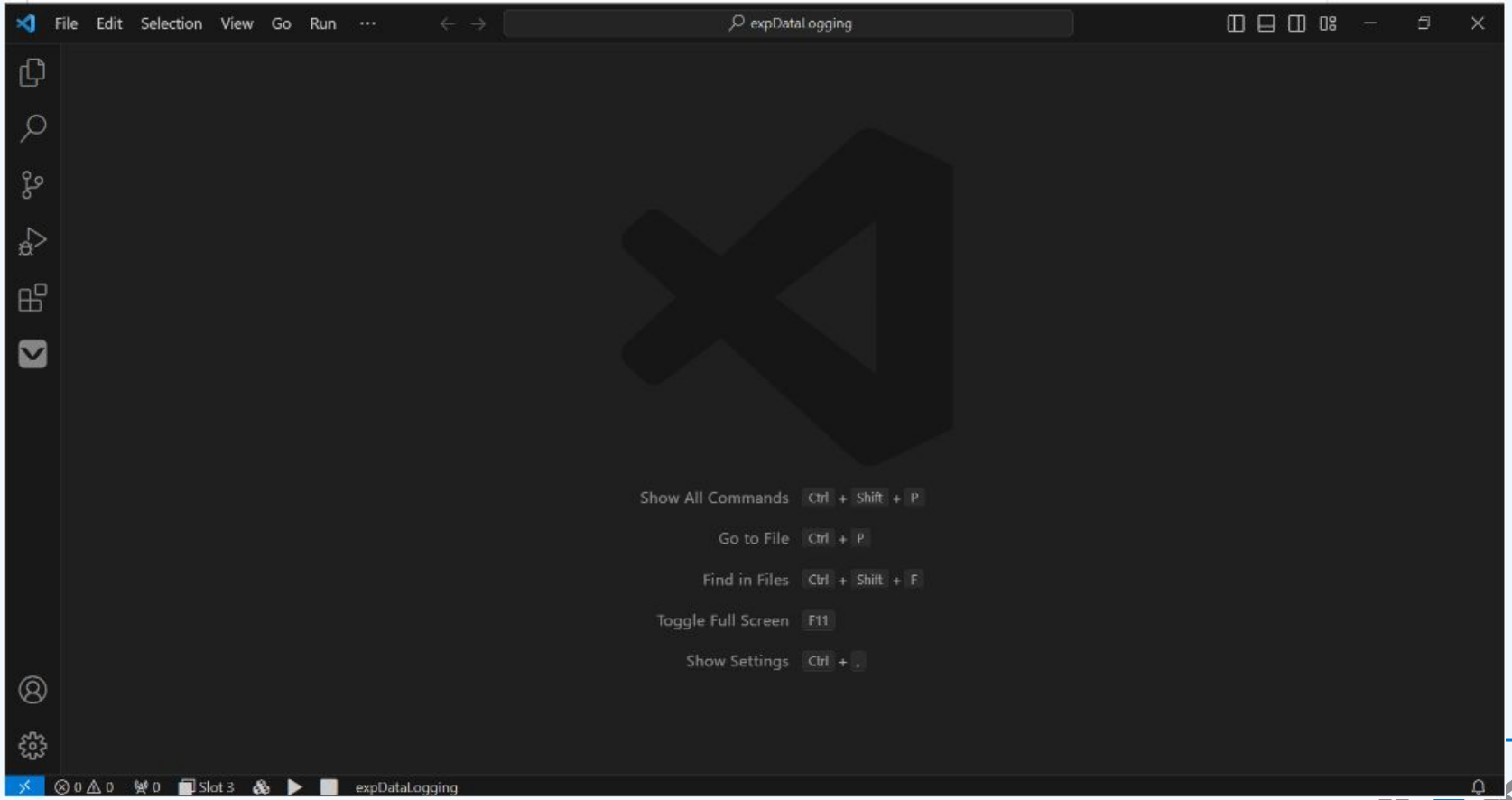
Why VS Code?

- Prepares students for professional coding
- Supports stronger coding habits
- Improves debugging and problem-solving

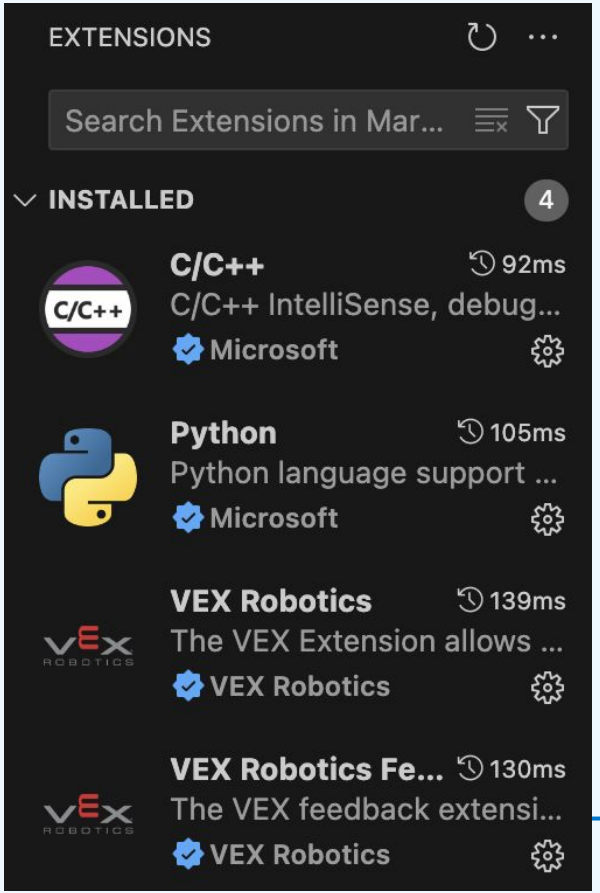
PITFALL ALERT

- Do NOT have VEXcode and VS Code opened at the same time!
- Mac needs Rosetta 2!
- VS Code Python only supports a single file.

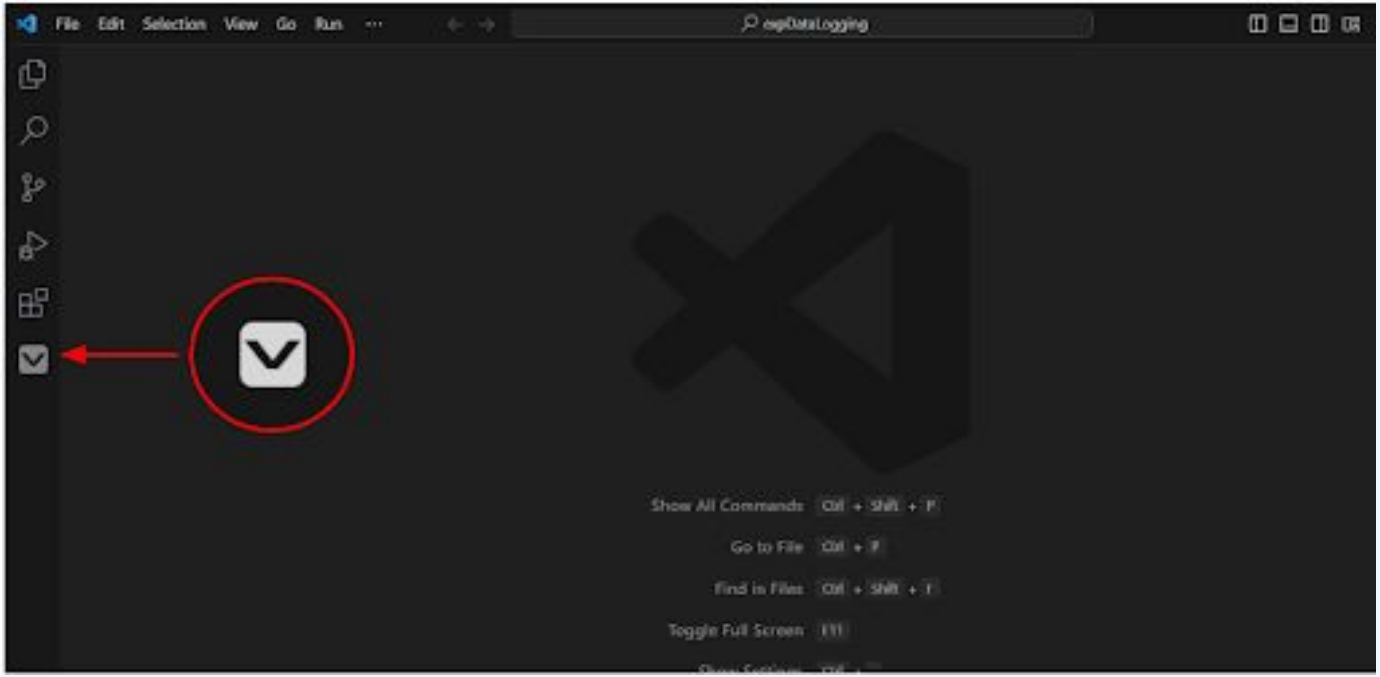
VS Code


File Edit Selection View Go Run ···
expDataLogging
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Toggle Full Screen F11
Show Settings Ctrl + ,
Slot 3
expDataLogging

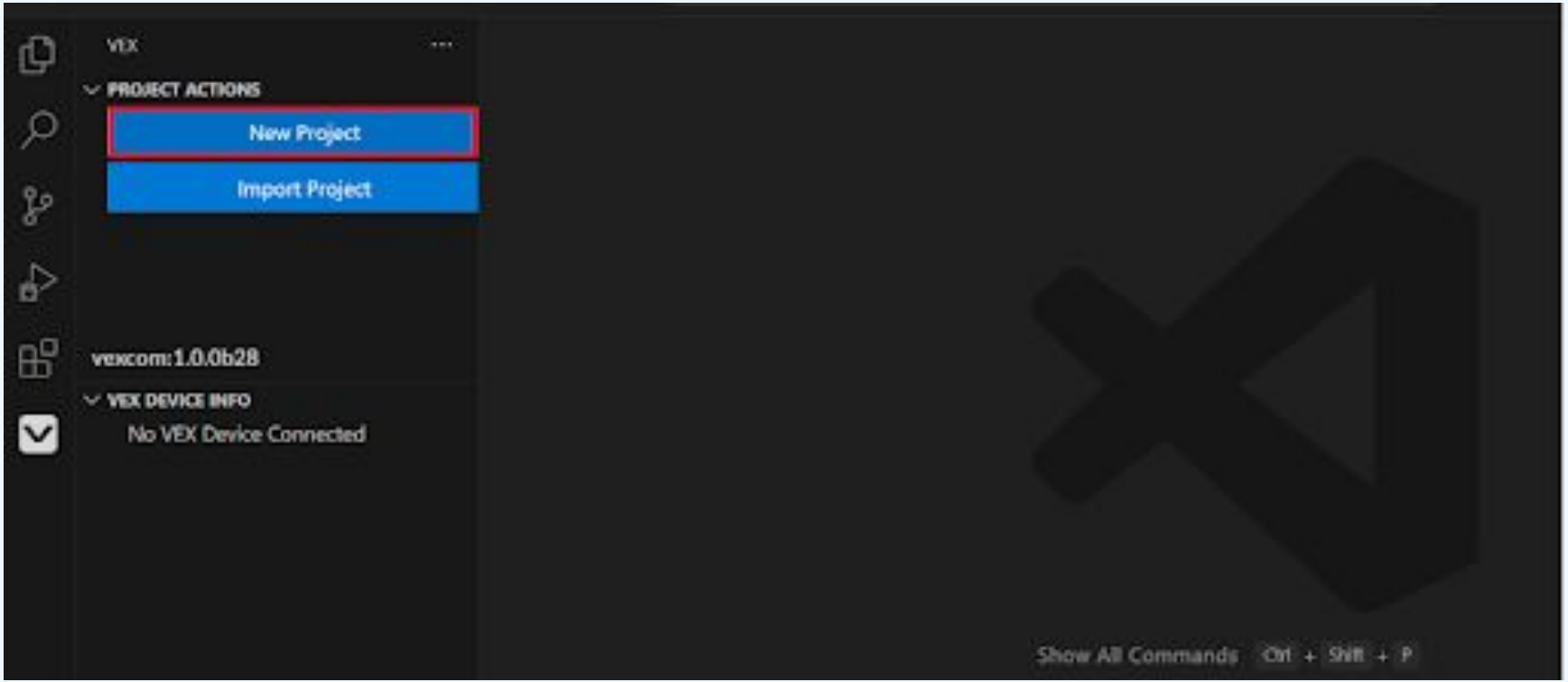
VS Code


EXTENSIONS
Search Extensions in Mar...
INSTALLED
4
C/C++
92ms
C/C++ IntelliSense, debug...
Microsoft
Python
105ms
Python language support ...
Microsoft
VEX Robotics
139ms
The VEX Extension allows ...
VEX Robotics
VEX Robotics Fe...
130ms
The VEX feedback extensi...
VEX Robotics

VS Code


File Edit Selection View Go Run ···
expDataLogging
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Toggle Full Screen F11

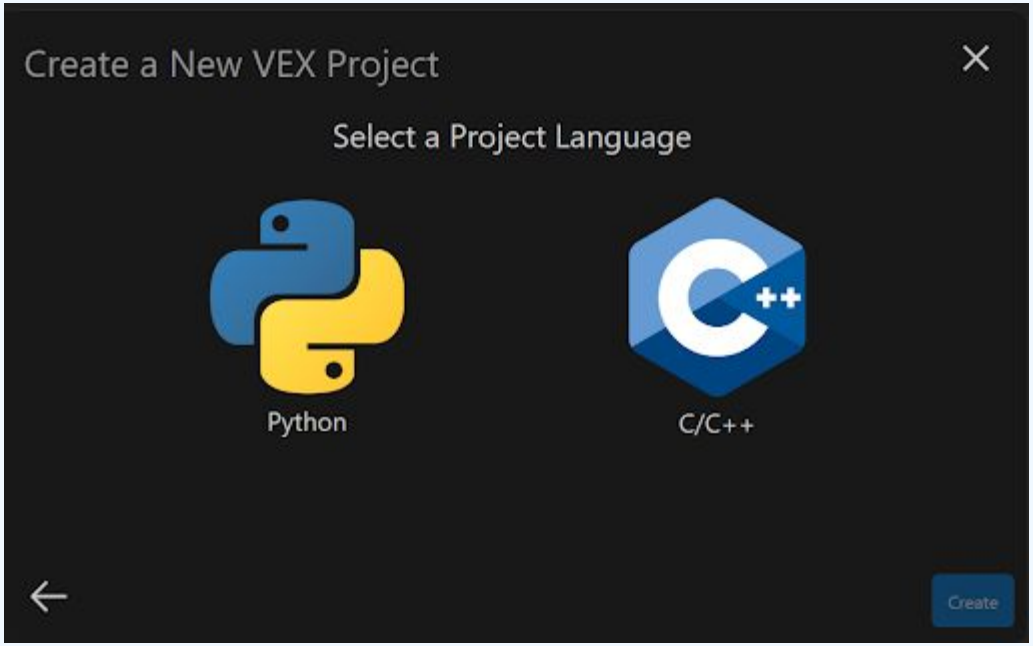
VS Code


VEX
PROJECT ACTIONS
New Project
Import Project
vexcom:1.0.0b28
VEX DEVICE INFO
No VEX Device Connected
Show All Commands Ctrl + Shift + P

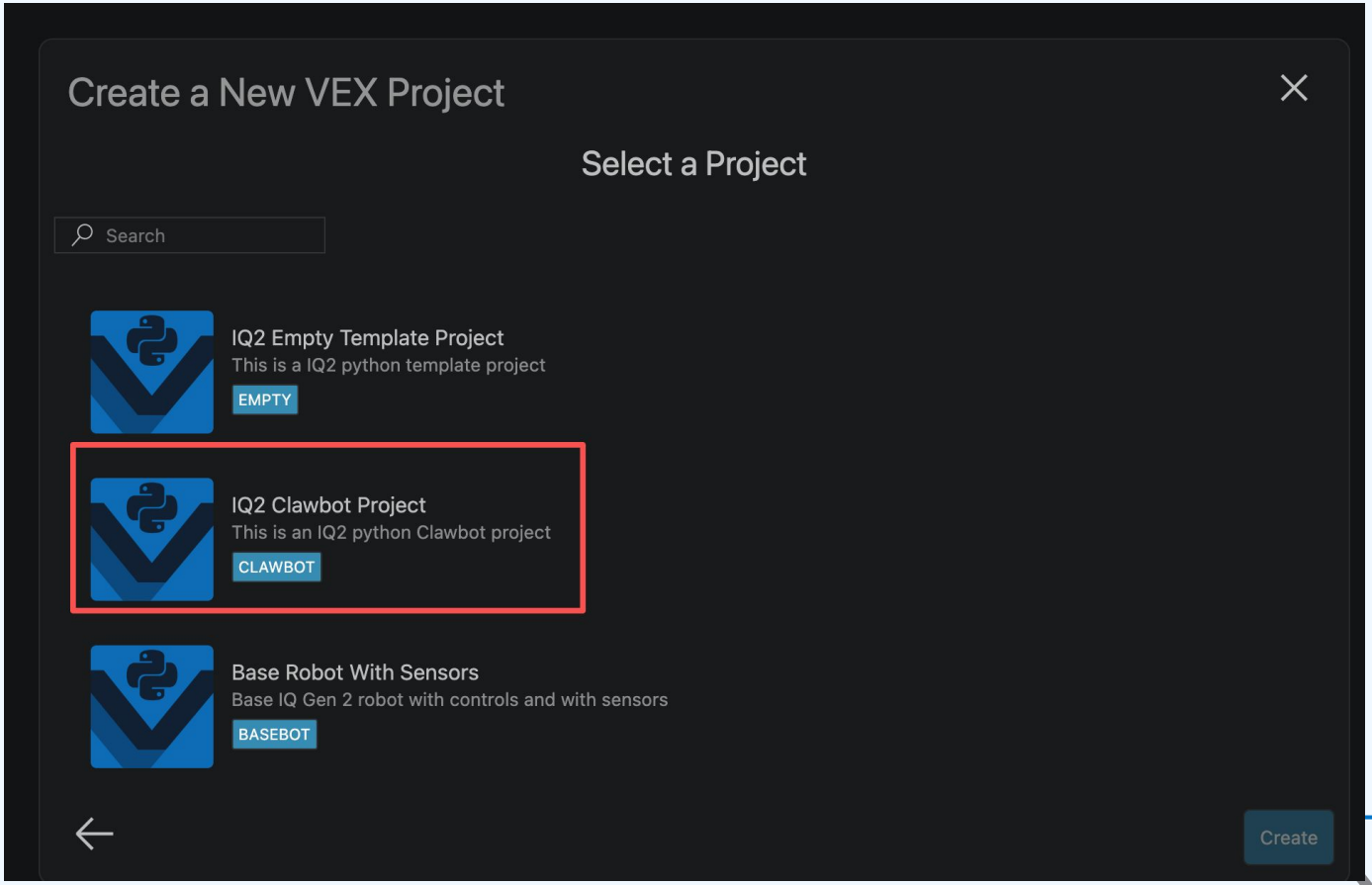
VS Code


Create a New VEX Project
Select a Project Platform
IQ
2nd GEN
IQ 2nd Generation
EXP
EXP
V5
V5
Create

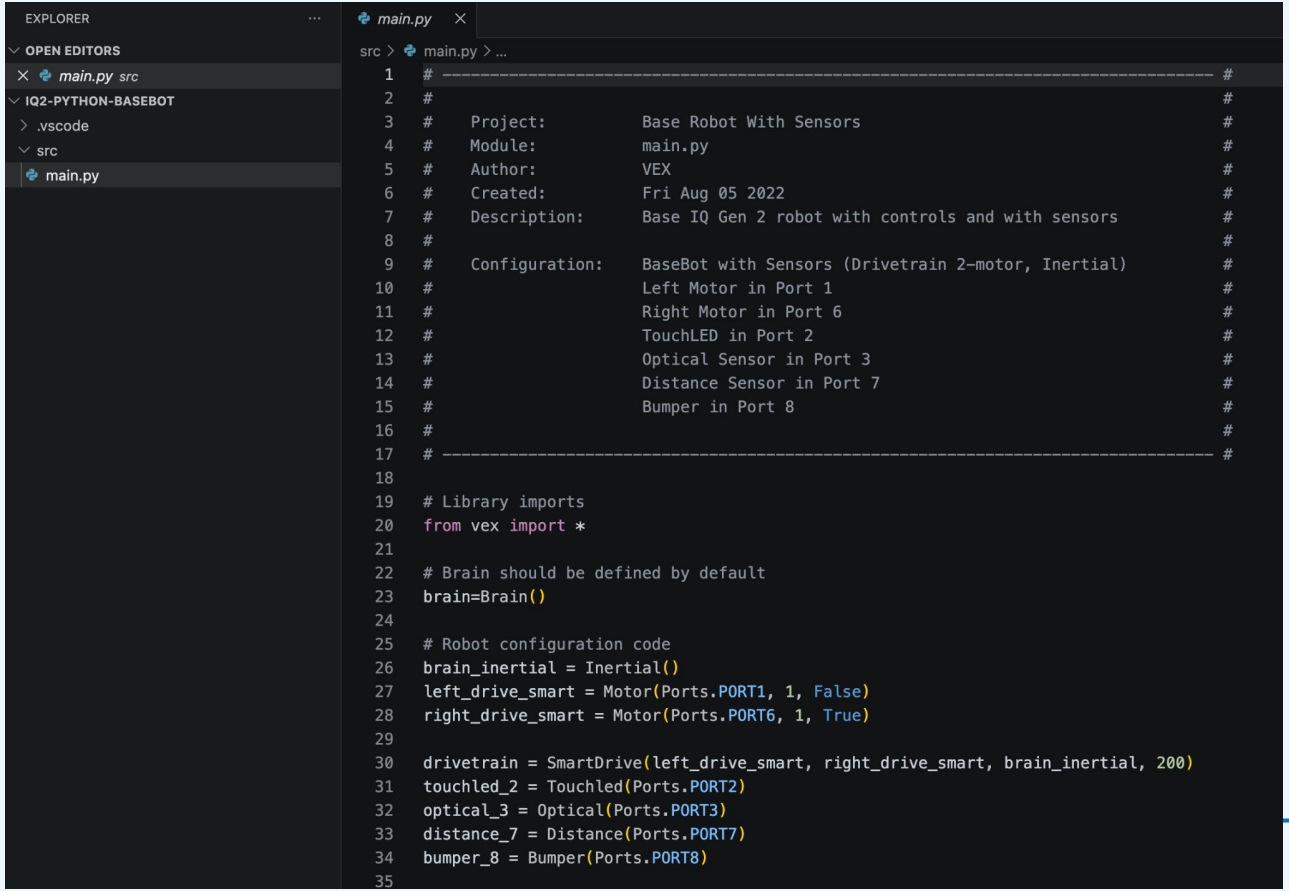
VS Code


Create a New VEX Project
Select a Project Language
Python
C/C++
Create

VS Code


Create a New VEX Project
Select a Project
Search
IQ2 Empty Template Project
This is a IQ2 python template project
EMPTY
IQ2 Clawbot Project
This is an IQ2 python Clawbot project
CLAWBOT
Base Robot With Sensors
Base IQ Gen 2 robot with controls and with sensors
BASEBOT
Create

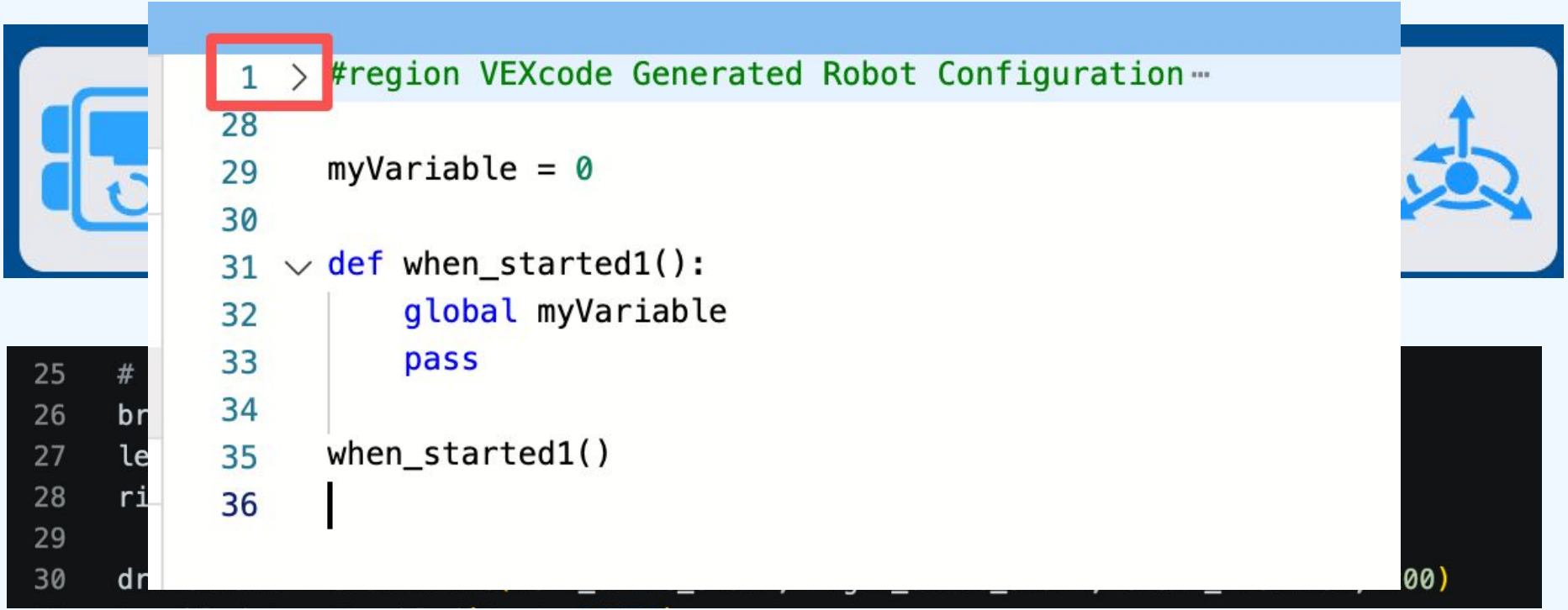
VS Code


EXPLORER
OPEN EDITORS
main.py src
IQ2-PYTHON-BASEBOT
.vscode
src
main.py
main.py
src > main.py > ...
1 # ---------- #
2 # #
3 # Project: Base Robot With Sensors #
4 # Module: main.py #
5 # Author: VEX #
6 # Created: Fri Aug 05 2022 #
7 # Description: Base IQ Gen 2 robot with controls and with sensors #
8 # #
9 # Configuration: BaseBot with Sensors (Drivetrain 2-motor, Inertial) #
10 # Left Motor in Port 1 #
11 # Right Motor in Port 6 #
12 # TouchLED in Port 2 #
13 # Optical Sensor in Port 3 #
14 # Distance Sensor in Port 7 #
15 # Bumper in Port 8 #
16 # #
17 # ---------- #
18
19 # Library imports
20 from vex import *
21
22 # Brain should be defined by default
23 brain=Brain()
24
25 # Robot configuration code
26 brain_inertial = Inertial()
27 left_drive_smart = Motor(Ports.PORT1, 1, False)
28 right_drive_smart = Motor(Ports.PORT6, 1, True)
29
30 drivetrain = SmartDrive(left_drive_smart, right_drive_smart, brain_inertial, 200)
31 touchled_2 = Touchled(Ports.PORT2)
32 optical_3 = Optical(Ports.PORT3)
33 distance_7 = Distance(Ports.PORT7)
34 bumper_8 = Bumper(Ports.PORT8)
35

VS Code


```
25  # Robot configuration code
26  brain_inertial = Inertial()
27  left_drive_smart = Motor(Ports.PORT1, 1, False)
28  right_drive_smart = Motor(Ports.PORT6, 1, True)
29
30  drivetrain = SmartDrive(left_drive_smart, right_drive_smart, brain_inertial, 200)
```

VS Code

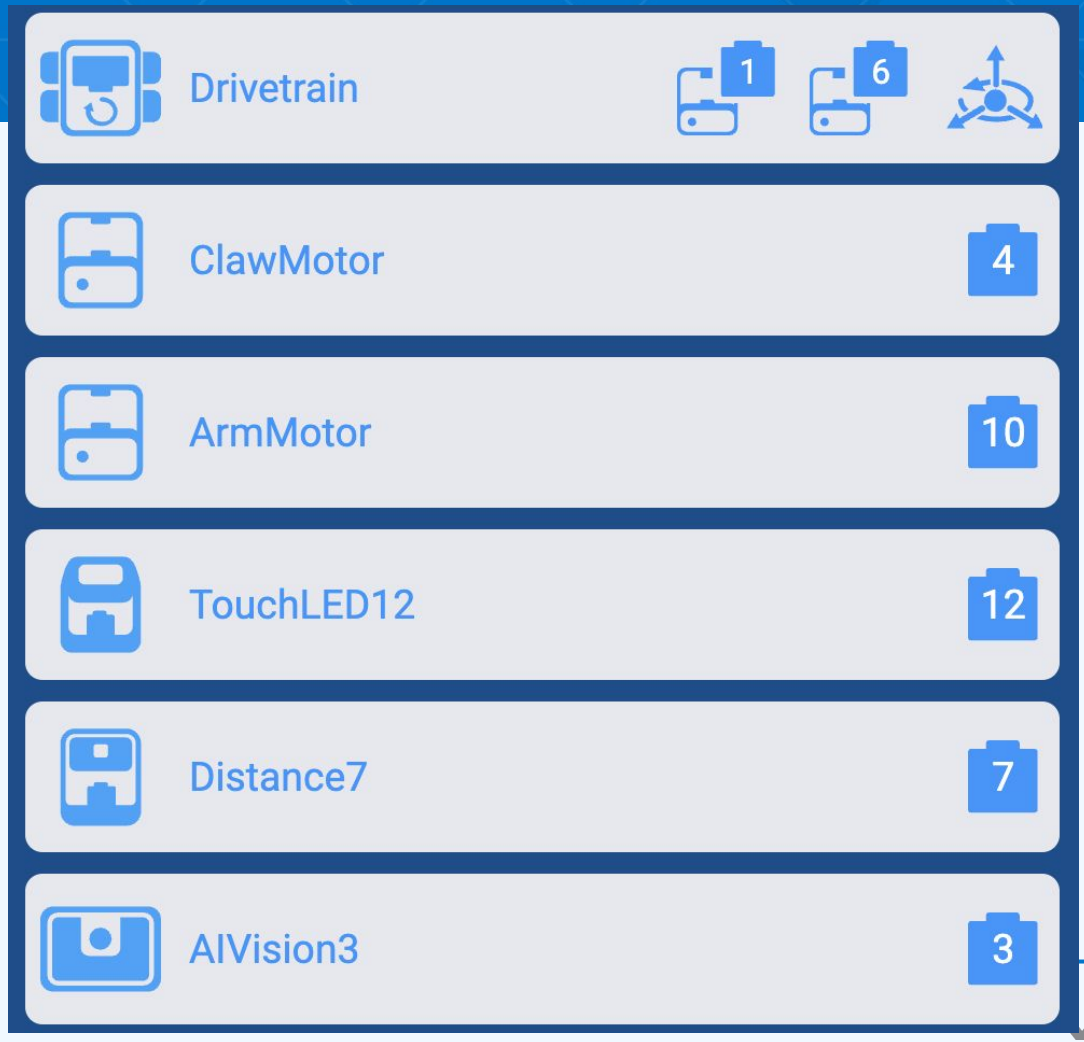

#region VEXcode Generated Robot Configuration
myVariable = 0
def when_started1():
global myVariable
pass
when_started1()

VS Code


0 0
0
IQ Brain (IQ)
Slot 1
IQ_Test
main.py


0 0
0
IQ Brain (IQ)
Slot 1
IQ_Test
main.py

Robot Config



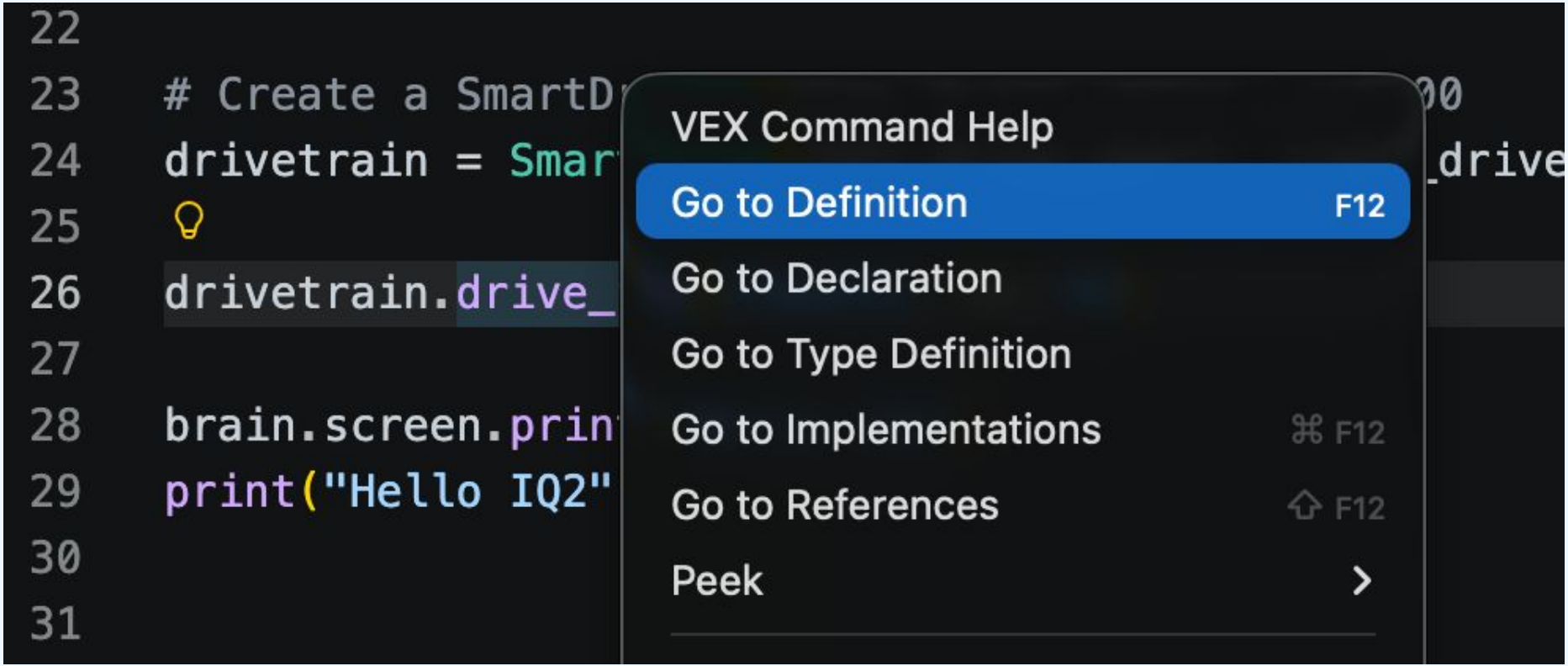
Activity 1: ClawBot – First part

Activity 1: ClawBot – First part

drivetrain.drive_for(FORWARD, 200, MM)

```
36
37  def calibrate_drivetrain():
38      # Calibrate the Drivetrain Inertial
39      sleep(200, MSEC)
40      brain.screen.print("Calibrating")
41      brain.screen.next_row()
42      brain.screen.print("Inertial")
43      brain_inertial.calibrate()
44      while brain_inertial.is_calibrating():
45          sleep(25, MSEC)
46      brain.screen.clear_screen()
47      brain.screen.set_cursor(1, 1)
48
49
50  # Begin project code
51
```

Activity 1: ClawBot – First part



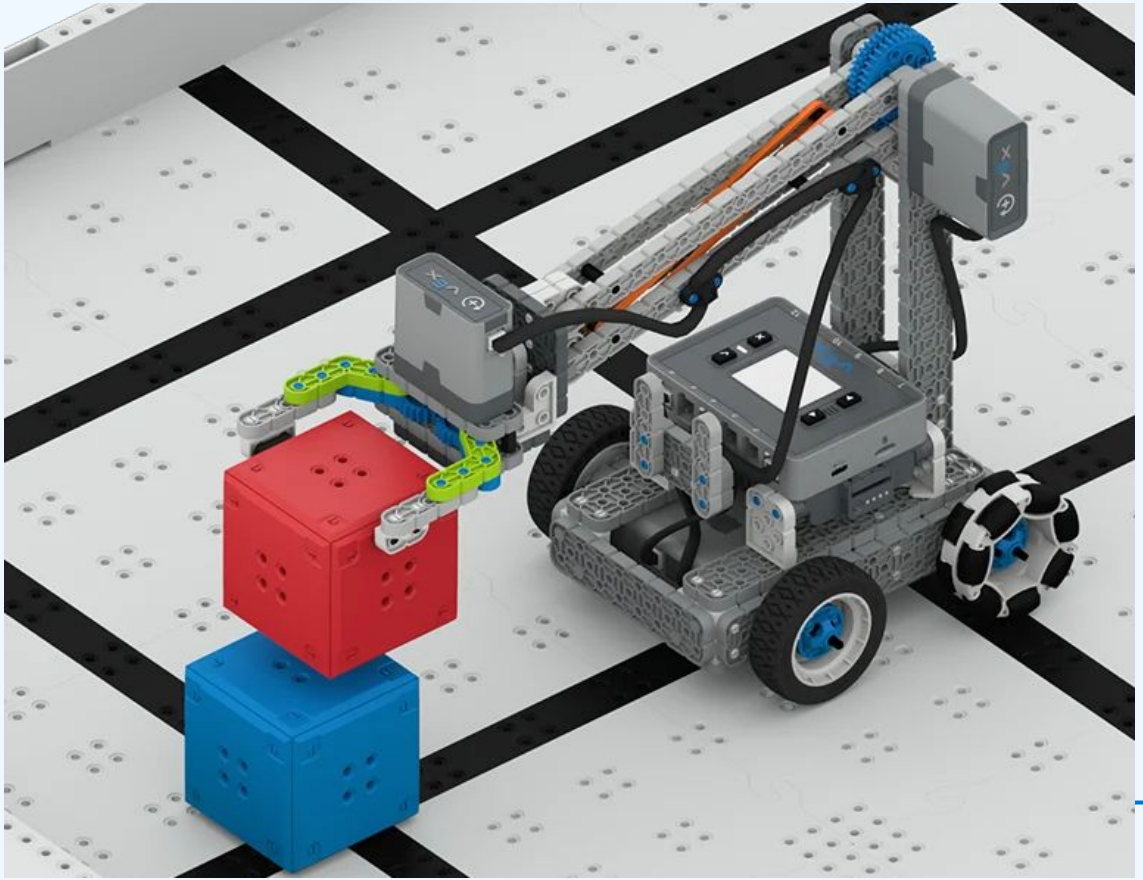
Activity 1: ClawBot – First part

Move forward and backward

Mid-Workshop Reflection

What is something that is different from VEXcode IQ?

Activity 2: Clawbot – Stack Cube

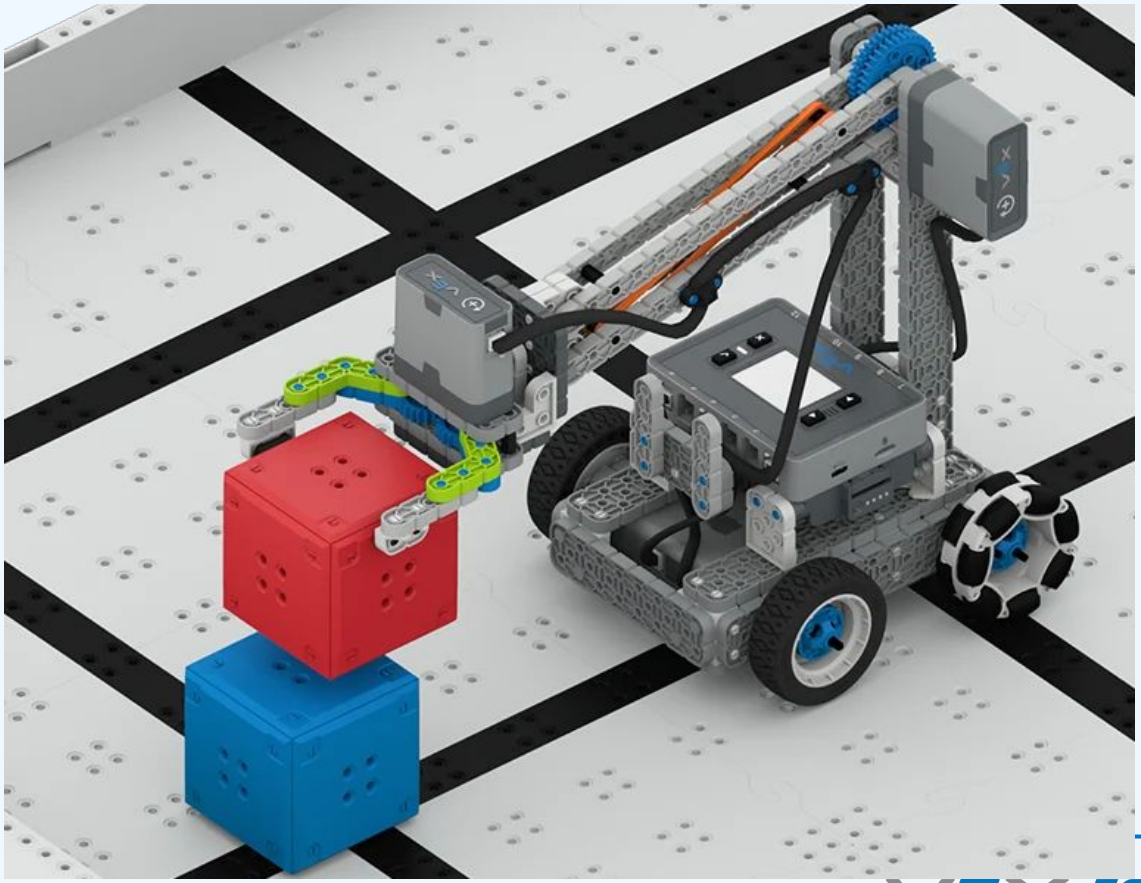


Activity 2: Clawbot – Stack Cube

- Autocomplete
- Go to definition
- Git version control

Activity 2: Clawbot – Stack Cube

Use the motor to stack a cube.



Activity 2: C

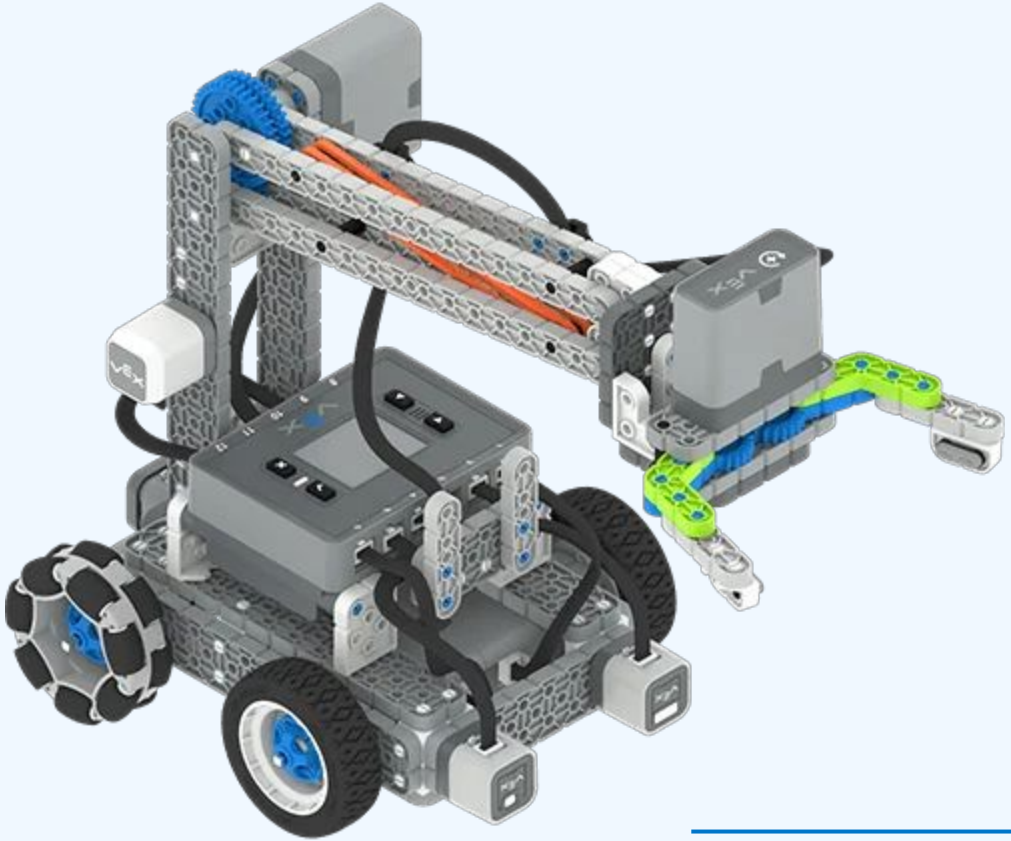
main.py

```
53  claw_motor.set_stopping(HOLD)
54  arm_motor.set_stopping(HOLD)
55
56  # Lower the arm
57  arm_motor.spin(FORWARD)
58  wait(500, MSEC)
59  arm_motor.stop()
60
61  # Open claw
62  claw_motor.spin(REVERSE)
63  wait(500, MSEC)
64  claw_motor.stop()
65
66  # Close claw and raise arm
67  claw_motor.spin(FORWARD)
68  wait(500, MSEC)
69  claw_motor.stop()
70  arm_motor.spin(REVERSE)
71  wait(5000, MSEC)
72  arm_motor.stop()
```

Mid-Workshop Reflection 2:

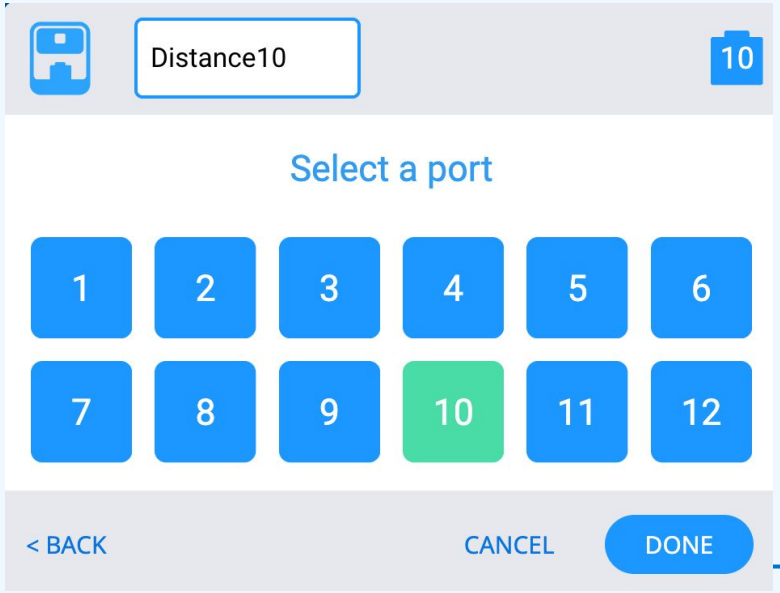
- Where do you find help: API or Definition?
- Vibe coding with VS Code

Activity 3: Clawbot – Sensors

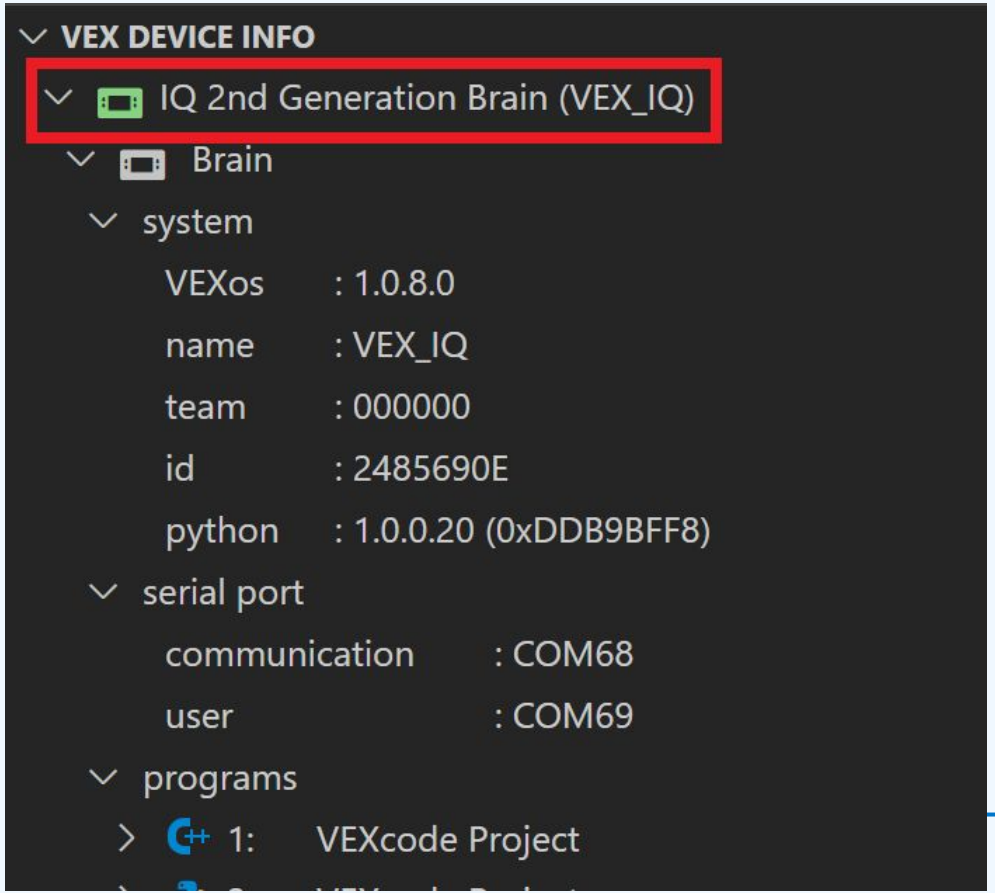


Activity 3: Clawbot – Sensors

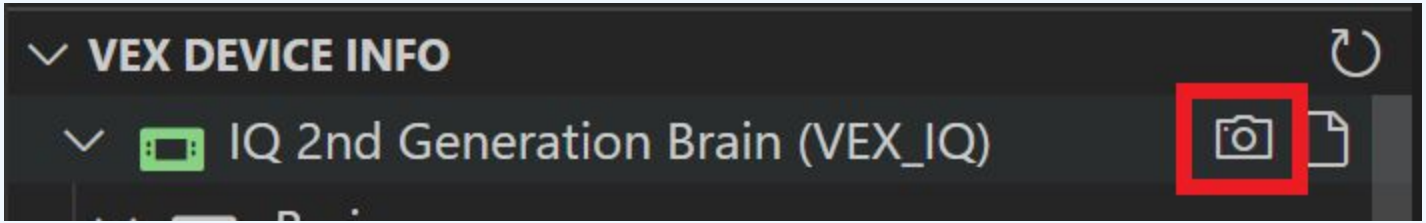
```
distance_sensor = Distance(Ports.PORT10)
print("distance", distance_sensor.object_distance(MM))
```



VS Code


VEX DEVICE INFO
IQ 2nd Generation Brain (VEX_IQ)
Brain
system
VEXos : 1.0.8.0
name : VEX_IQ
team : 000000
id : 2485690E
python : 1.0.0.20 (0xDDB9BFF8)
serial port
communication : COM68
user : COM69
programs
1: VEXcode Project

VS Code



VS Code

Activity 3: Clawbot – Sensors

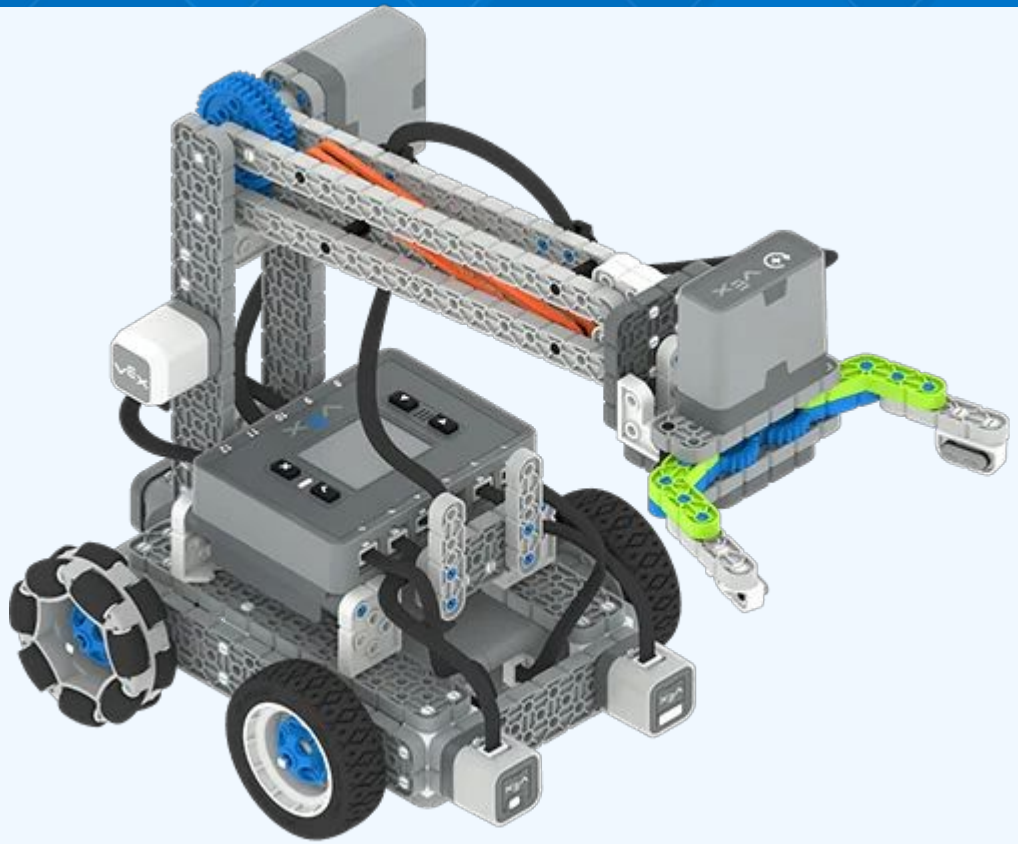
- api.vexnet.com IQ2 - Python
  - Logic - Controls
  - Sensing - Distance
  - Sensing - TouchLED
- Print in console
- Download to different slot

Activity 3: Clawbot – Sensors

- api.vexnet.com/iq2/home/python
  - /Logic/Controls.html
  - /Sensing/Distance.html
  - /Sensing/TouchLED.html
- Print in console
- Download to different slot

Activity 3: Clawbot – Sensors

Change the Touch LED's color based on the detected cube color.



Wrap Up

- How does the VS Code extension prepare students for real-world coding?
- What VS Code features, such as Autocomplete, improve debugging and problem-solving?
- What key differences did you notice between the professional coding workflow in VS Code and VEXcode IQ?

Resources

- **VEX Library:** kb.vex.com
- **API Sites:** api.vex.com

Stay Connected

Let's Connect!

Tag me in the **VEX PD+ Community!**
@Jimmy_Lin

Want to Learn More? Join VEX PD+ as an All-Access Member!

Schedule a **1-on-1 Session** in VEX PD+

Take a VEX Masterclass