Programming

Groups

Vex Coding Studio – Greg

RobotC (Cortex) - Judson

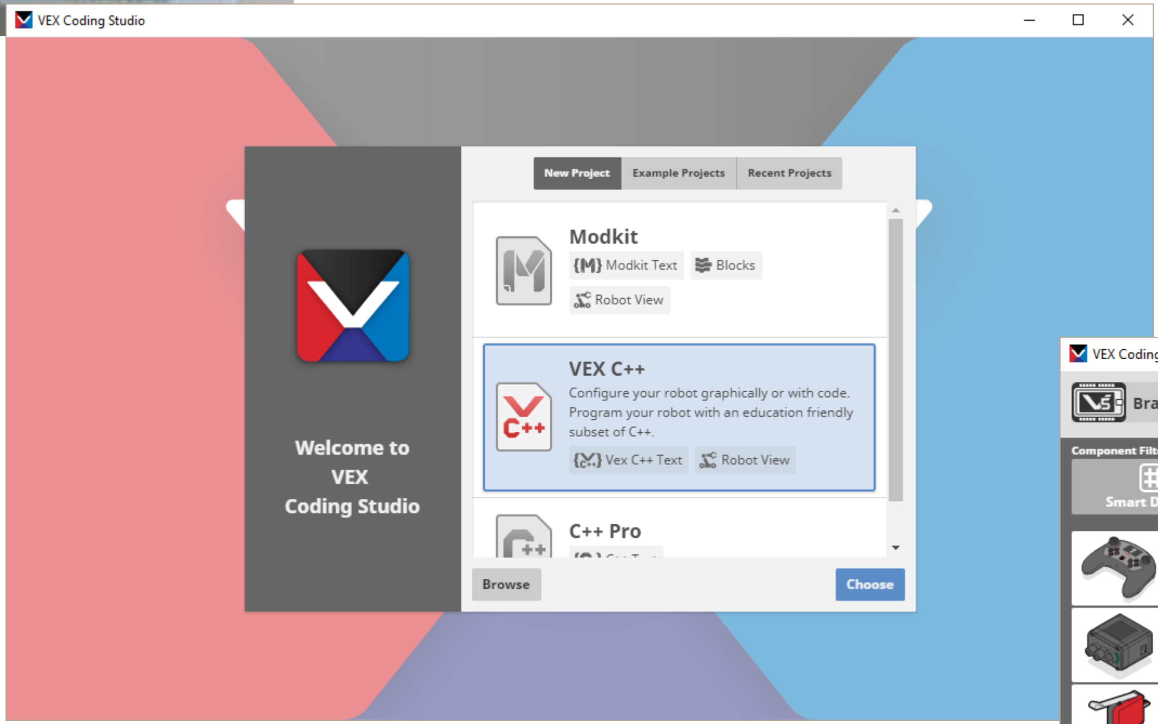
Programming the Parts: Some of the Options

- **VEX Coding Studio (What we will cover in this session)**
  - Free
  - MODKIT: Drag and Drop
  - **VEX C++ (This session)**
  - C++ Pro
- Robot Mesh Studio
  - Free
  - VEX iQ, VEX EDR Cortex, VEX V5 (Python), VEX V5 C++
- VEXCode: Still in pre-release. New version coming in August
  - Free
  - Similar to RobotC and VEX Coding Studio.

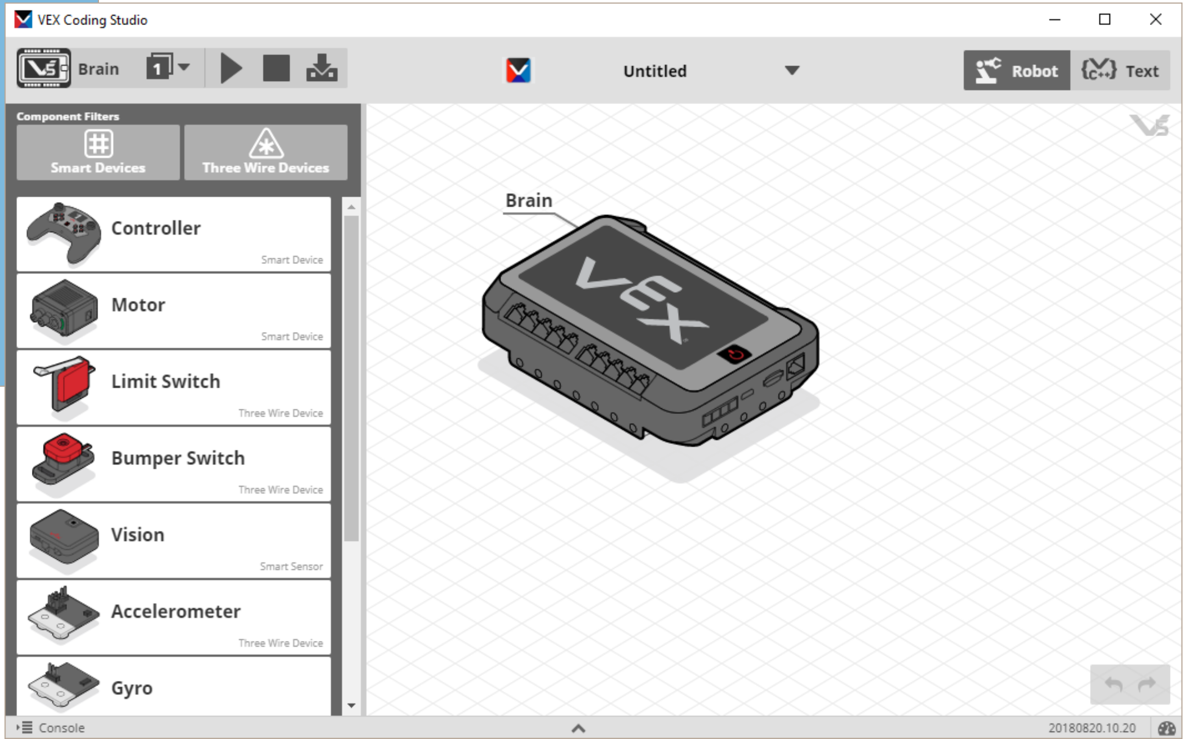
Steps to get up and running

- Download and install VEX Coding Studio
- Connect the battery to the charger and the robot brain
  - Keep battery on the charger during the robot firmware download
- Connect Smart Motor, Antennae, and remote control to the V5 Brain

VEX Coding Studio: Getting Started

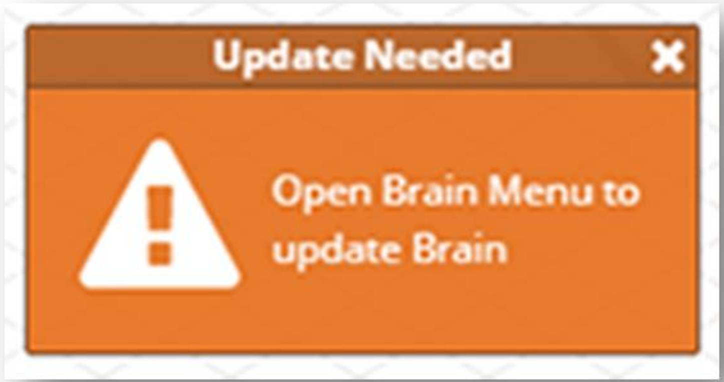


Configure your robot graphically or with code. Program your robot with an education friendly subset of C++.



Firmware Update

1. Connect all the Smart Devices to the V5 Brain
2. Connect the V5 Brain to a USB port on your computer with a Micro USB Cable.
3. Turn on the Robot Brain
4. Open VEX Coding Studio
   1. If you are connected to the internet, VEX Coding Studio will automatically check for updates.
5. VEX Coding Studio should find your Robot Brain. If not up to date, you will be an Update Needed message

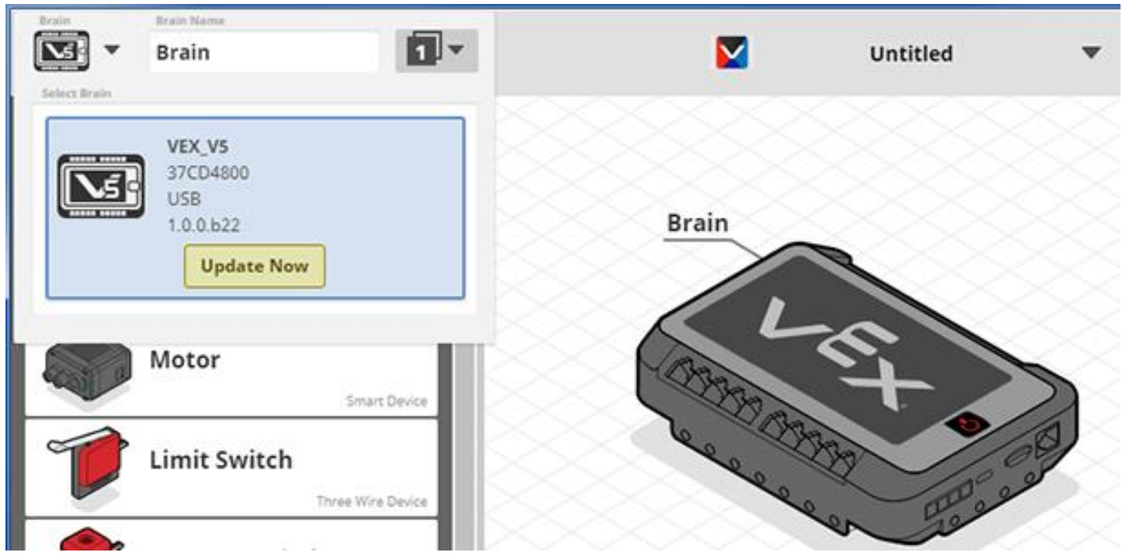


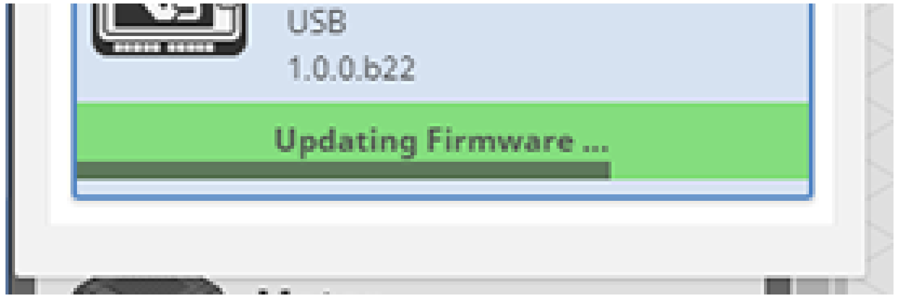
Update Needed

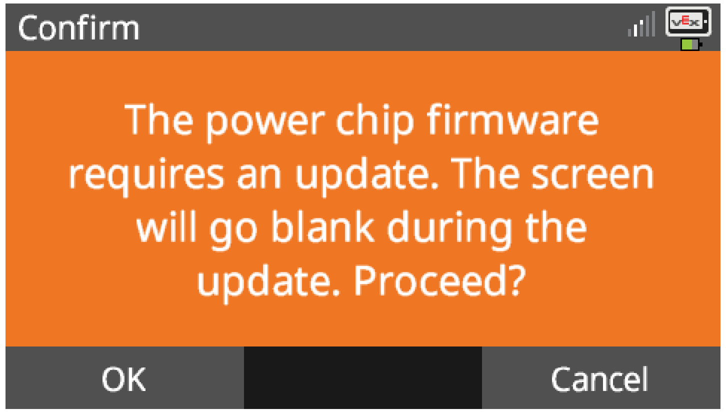
Open Brain Menu to update Brain

Updating Firmware

1. Click the small Robot Brain icon to open the Robot Brain Menu
2. Press 'Update Now'
3. After the update is completed, restart the Robot Brain
4. Restart the Brain
5. Update other V5 components as prompted.





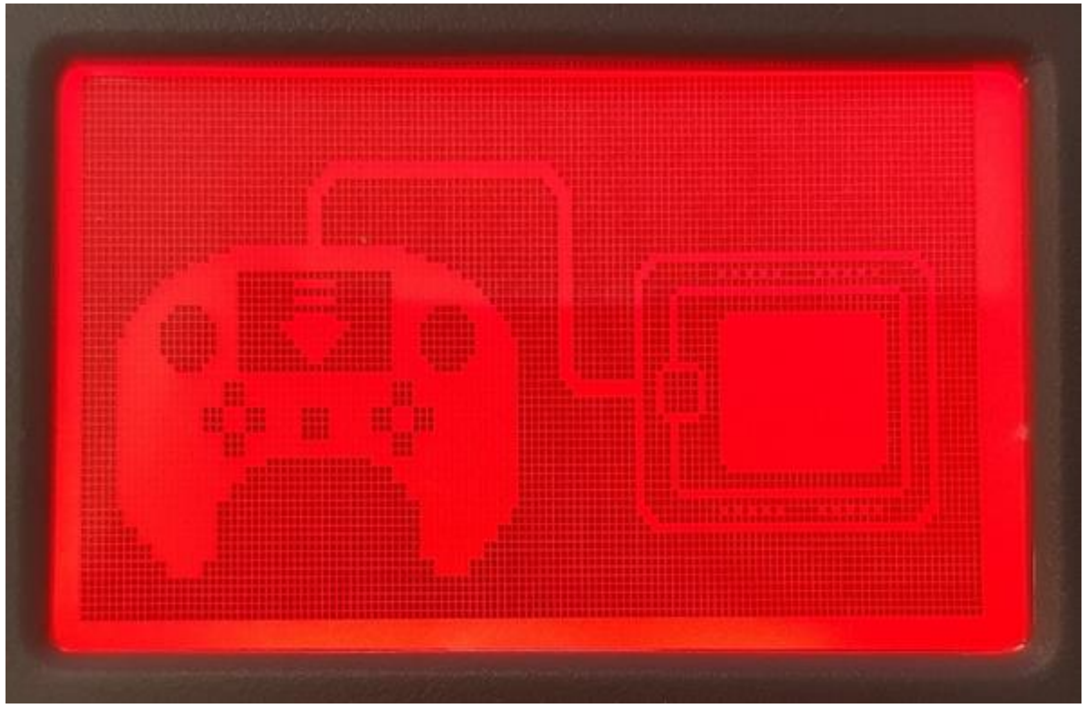

Confirm

The power chip firmware requires an update. The screen will go blank during the update. Proceed?

OK Cancel

Firmware update notes

1. Have **battery plugged into the charger** while updating.
2. **If** the Brain **DOES NOT** turn back on after the initial firmware update
    1. Press and hold the 'Power' button to make sure the system is turned off
    2. Unplug the battery cable and plug it back in. You should be able to turn on the robot.
3. If the Wireless Controller was also updated, the screen will turn red, and that is normal.



Programming the V5 Brain

The Editor

Brain Menu
Update Firmware: Select a port on the Brain for saving program

Play: Run the Program

Download the program to the Brain

Click to go to Motors/Sensors Setup

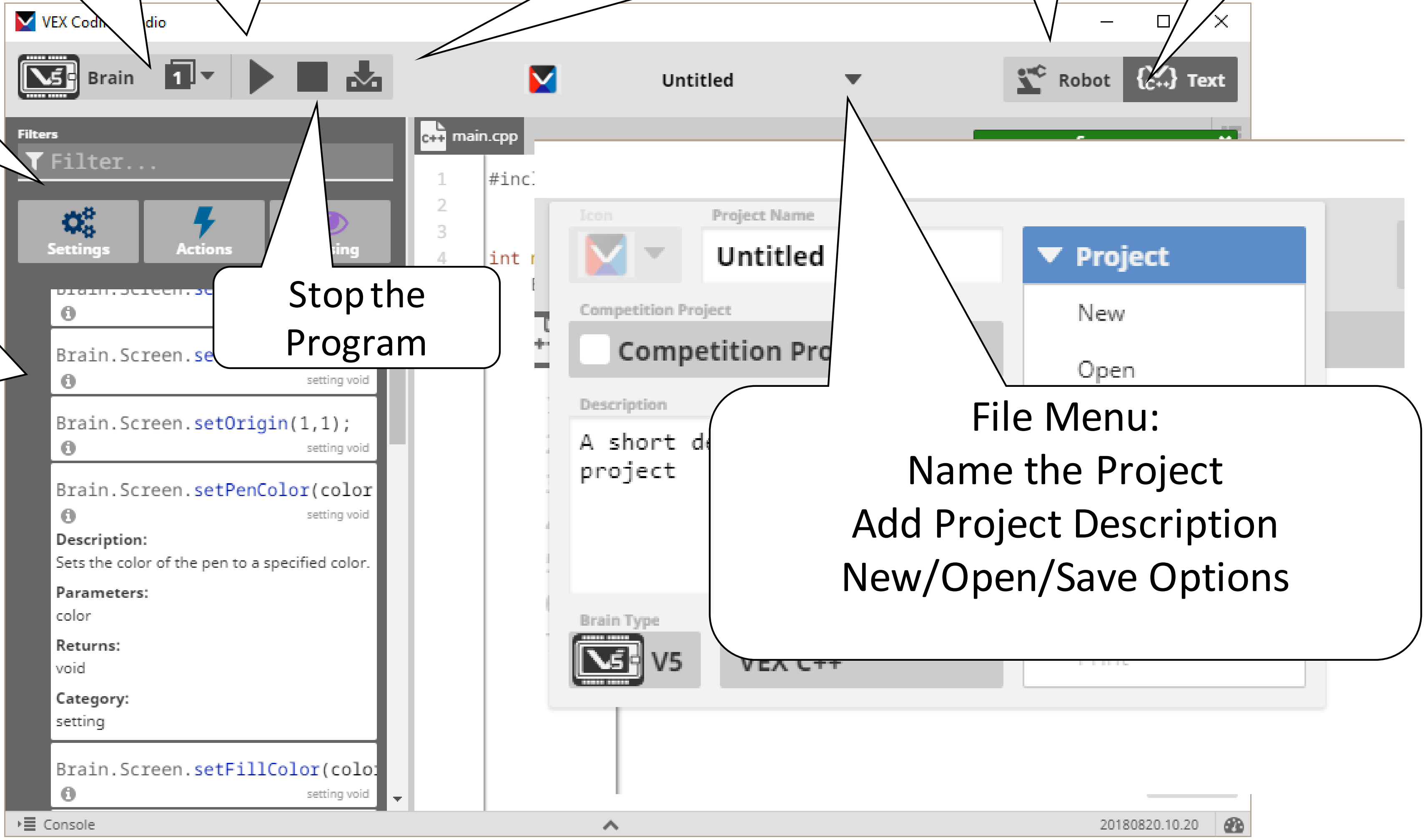
Click to Go to Coding screen

Code Libraries: Brings up different code options

Code Suggestions: Drag and Drop Options

Stop the Program

File Menu:
Name the Project
Add Project Description
New/Open/Save Options



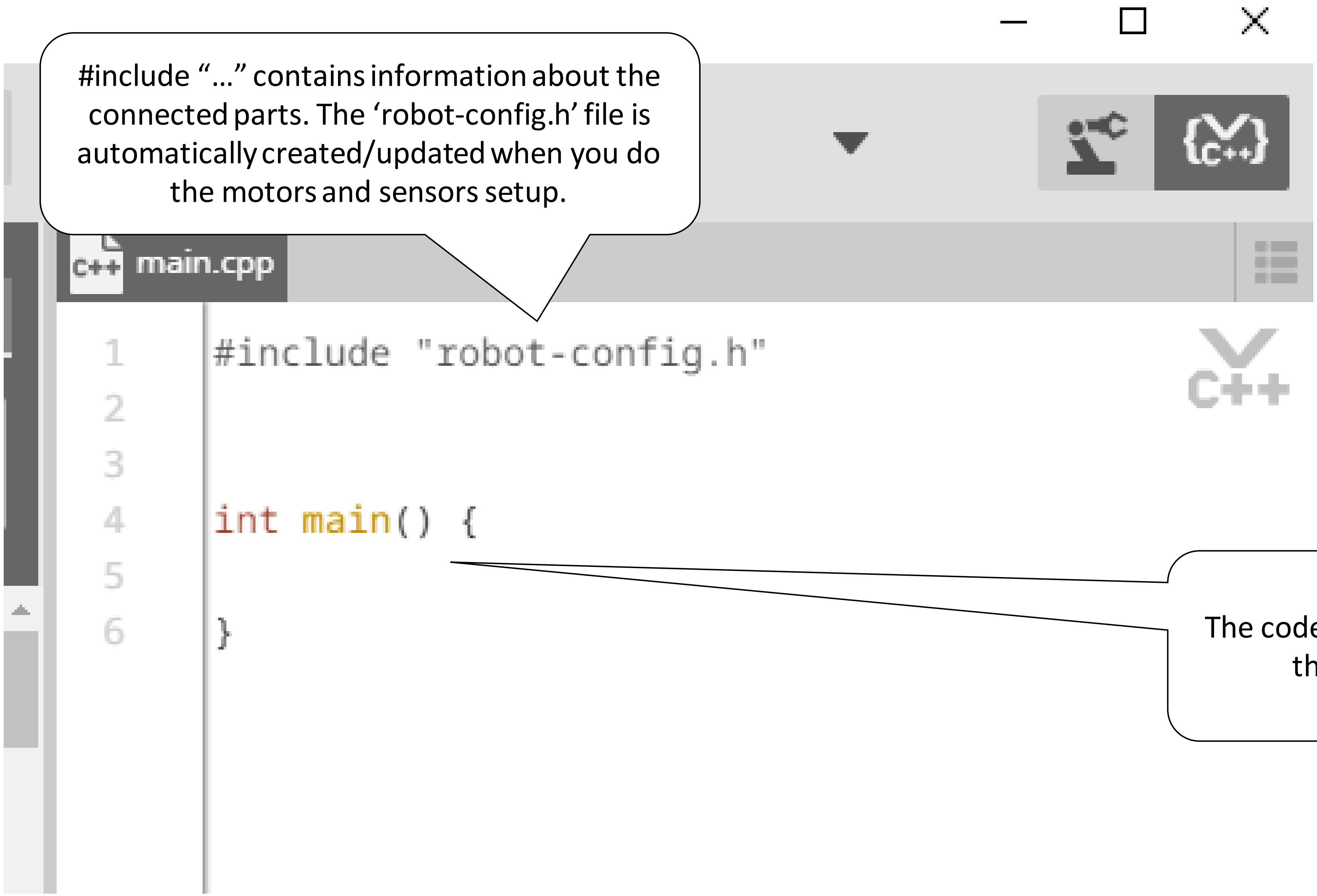
```
#incl
int r
```

```
Brain.Screen.se
setting void
Brain.Screen.setOrigin(1,1);
setting void
Brain.Screen.setPenColor(color
setting void
```

Description:
Sets the color of the pen to a specified color.
Parameters:
color
Returns:
void
Category:
setting

```
Brain.Screen.setFillColor(colo
setting void
```

A short de
project

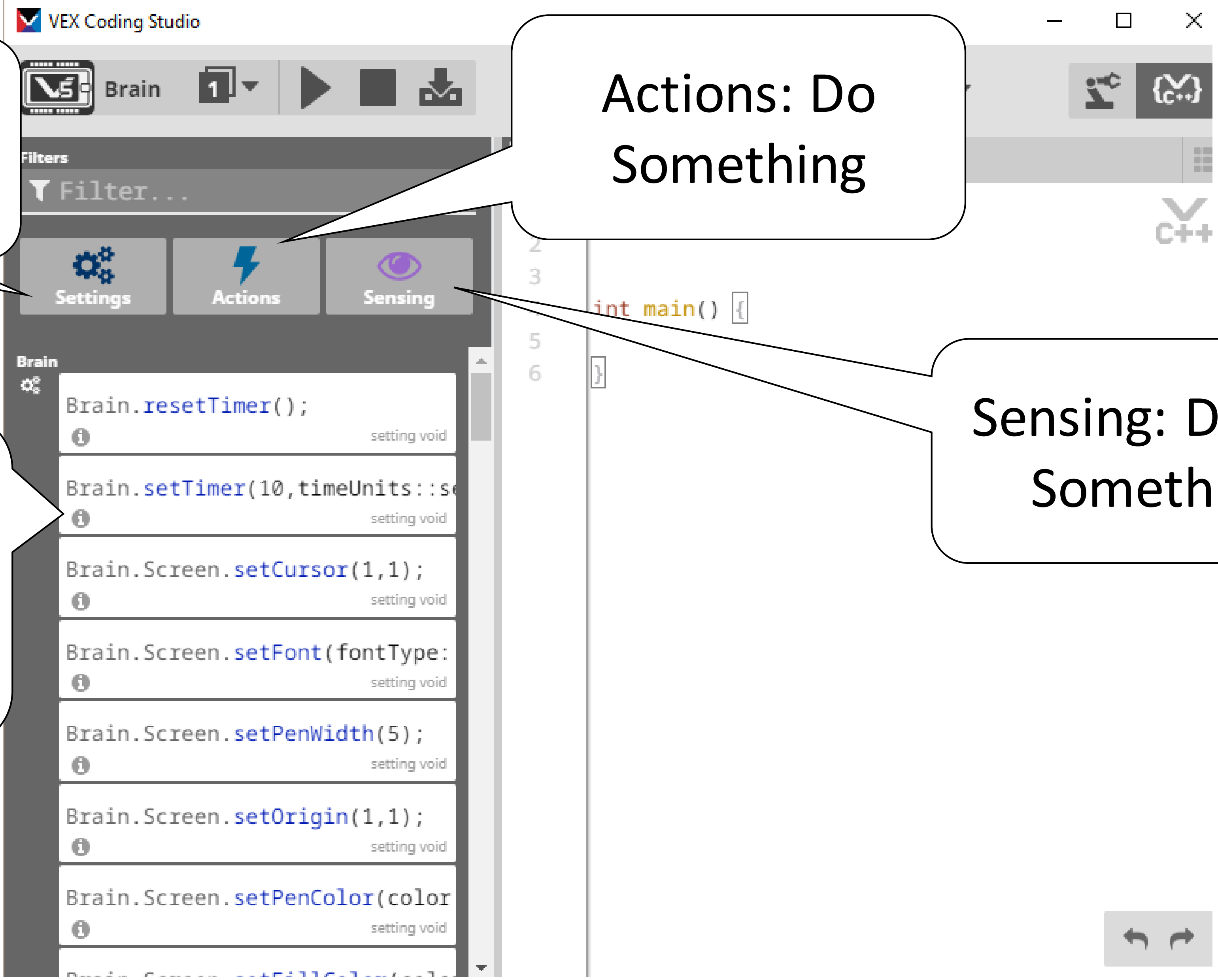


```
#include "robot-config.h"


int main() {

}
```

#include "…" contains information about the connected parts. The 'robot-config.h' file is automatically created/updated when you do the motors and sensors setup.

The code goes between the { and }

Code Libraries: Broken Down into Sections



```
Brain.resetTimer();
setting void
Brain.setTimer(10,timeUnits::se
setting void
Brain.Screen.setCursor(1,1);
setting void
Brain.Screen.setFont(fontType:
setting void
Brain.Screen.setPenWidth(5);
setting void
Brain.Screen.setOrigin(1,1);
setting void
Brain.Screen.setPenColor(color
setting void
```

```
int main() {

}
```

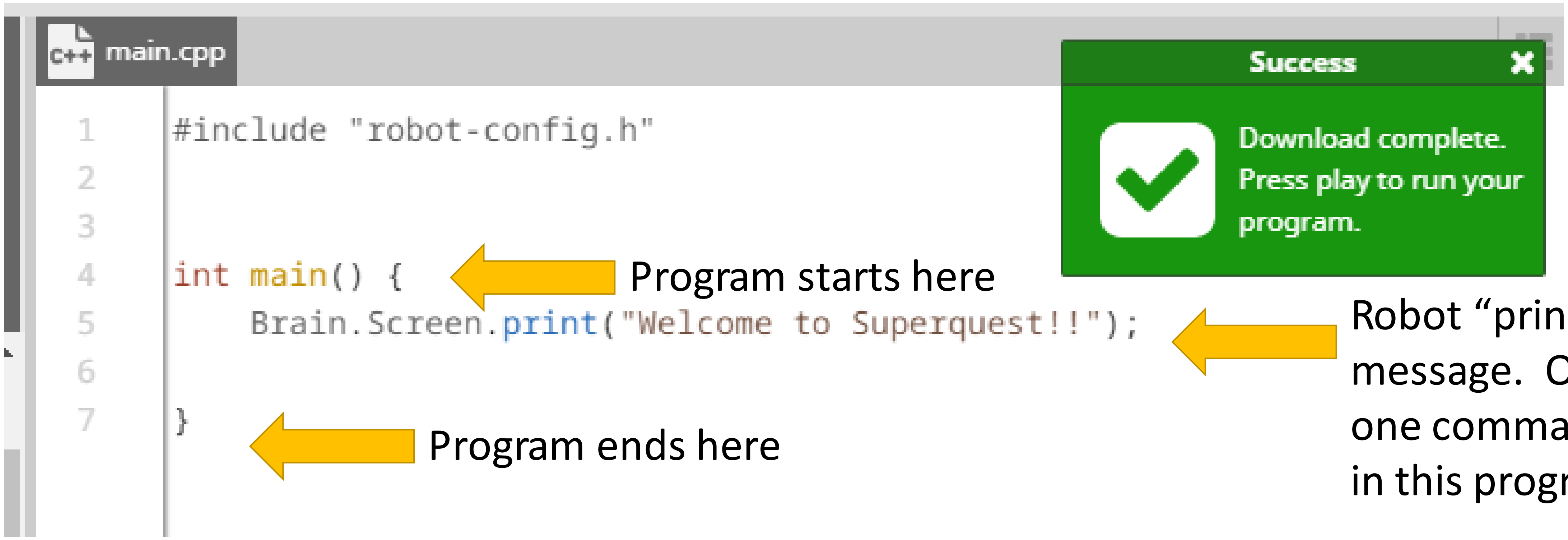
Settings: Change Something

Actions: Do Something

Sensing: Detect Something

Commands you can drag and drop into the code.

First Program



```
#include "robot-config.h"


int main() {
    Brain.Screen.print("Welcome to Superquest!!");

}
```

Download complete. Press play to run your program.

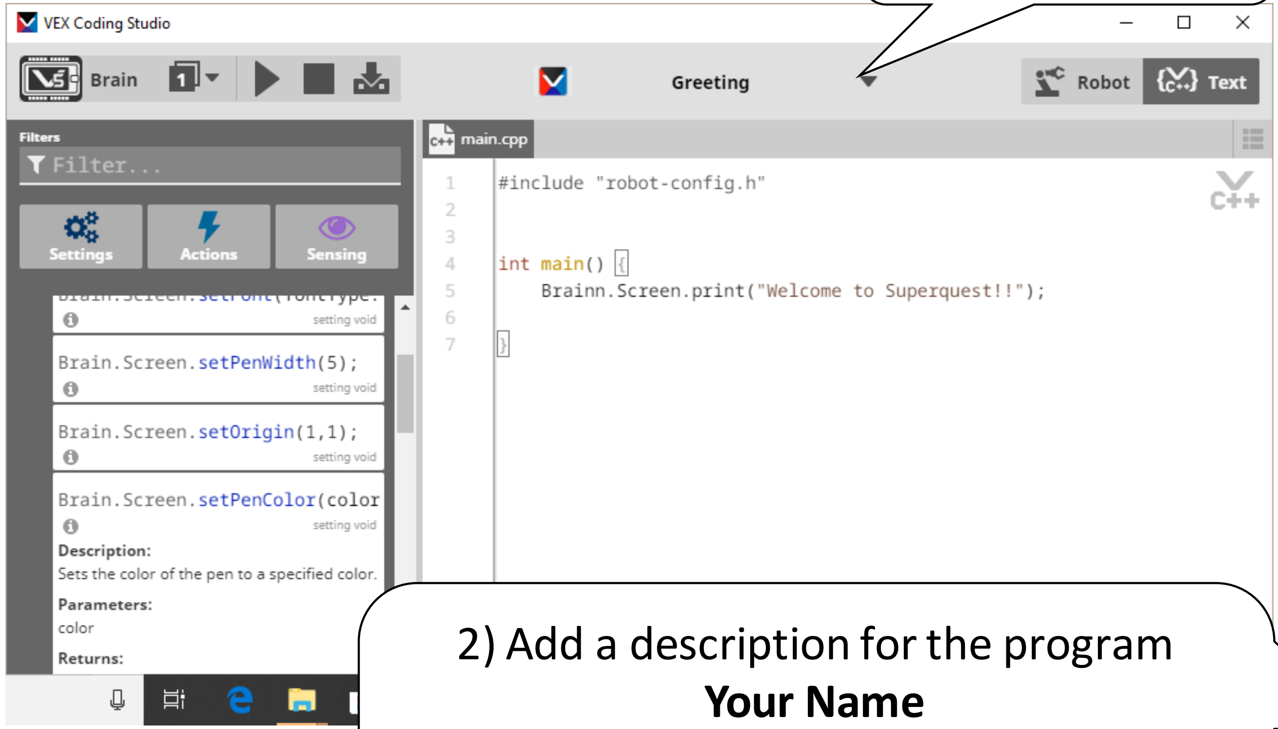
Program starts here

Robot “prints” message. Only one command in this program

Program ends here

Save the Project

Pull Down Menu for Saving…

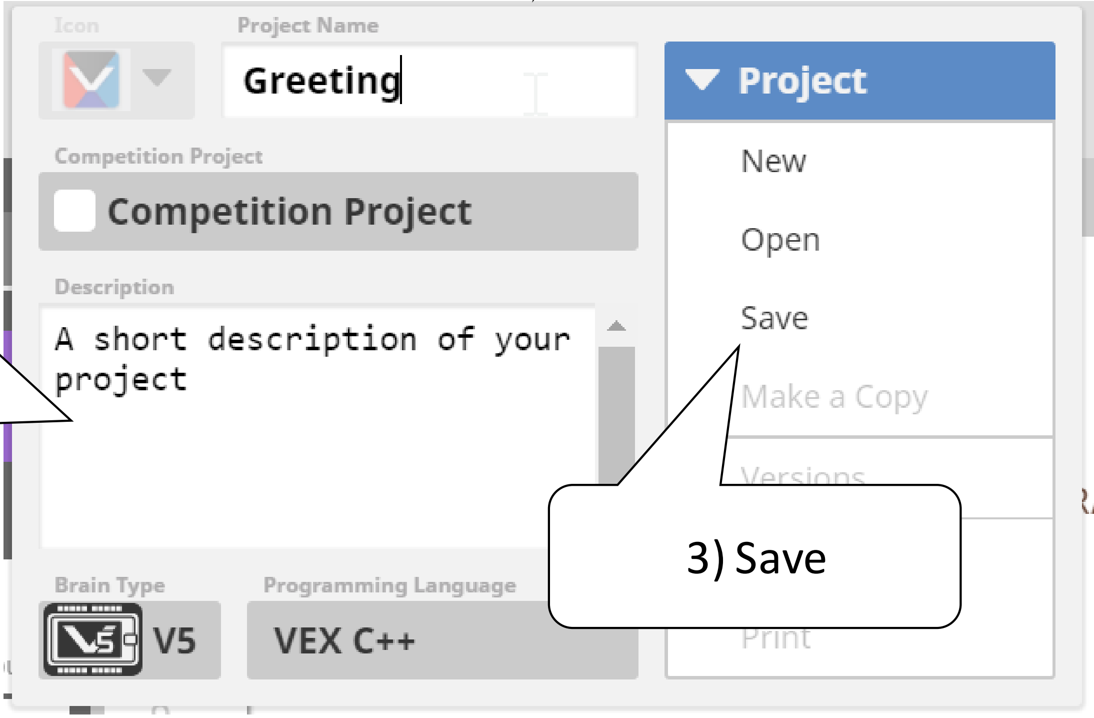


```
#include "robot-config.h"


int main() {
    Brainn.Screen.print("Welcome to Superquest!!");

}
```

```
Brain.Screen.setPenWidth(5);
Brain.Screen.setOrigin(1,1);
Brain.Screen.setPenColor(color
```

Description:
Sets the color of the pen to a specified color.
Parameters:
color
Returns:

1) Name the Program



Project Name: Greeting

Competition Project: Competition Project

Description: A short description of your project

Brain Type: V5

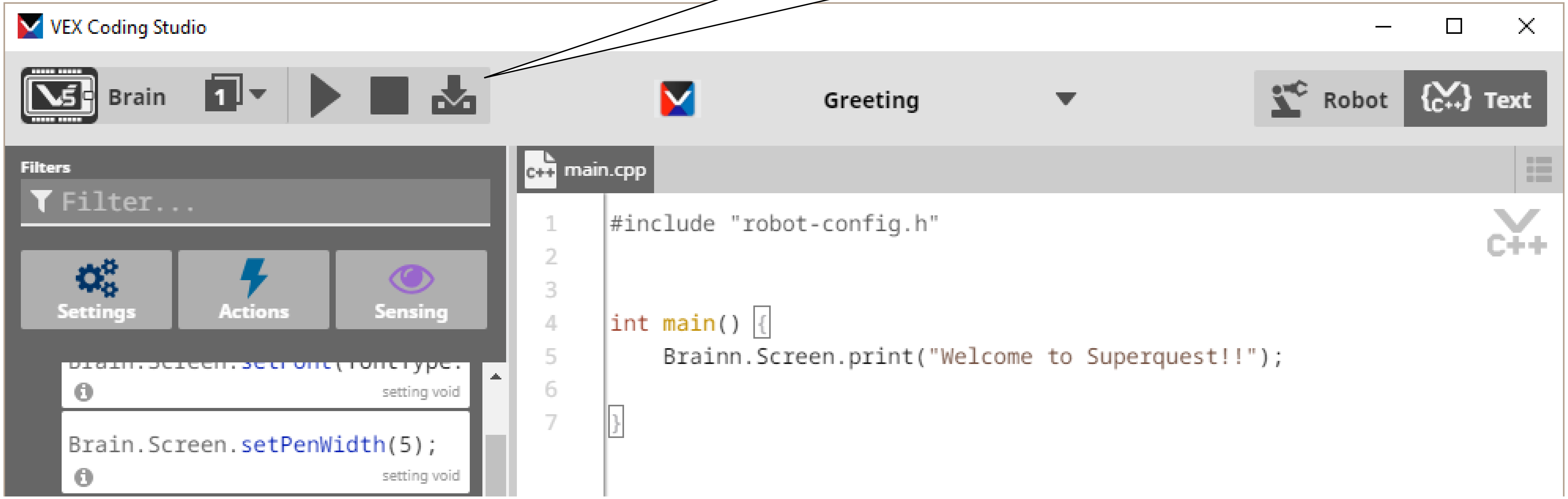
Programming Language: VEX C++

2) Add a description for the program
Your Name
Short Description of the program
Date
Version

3) Save

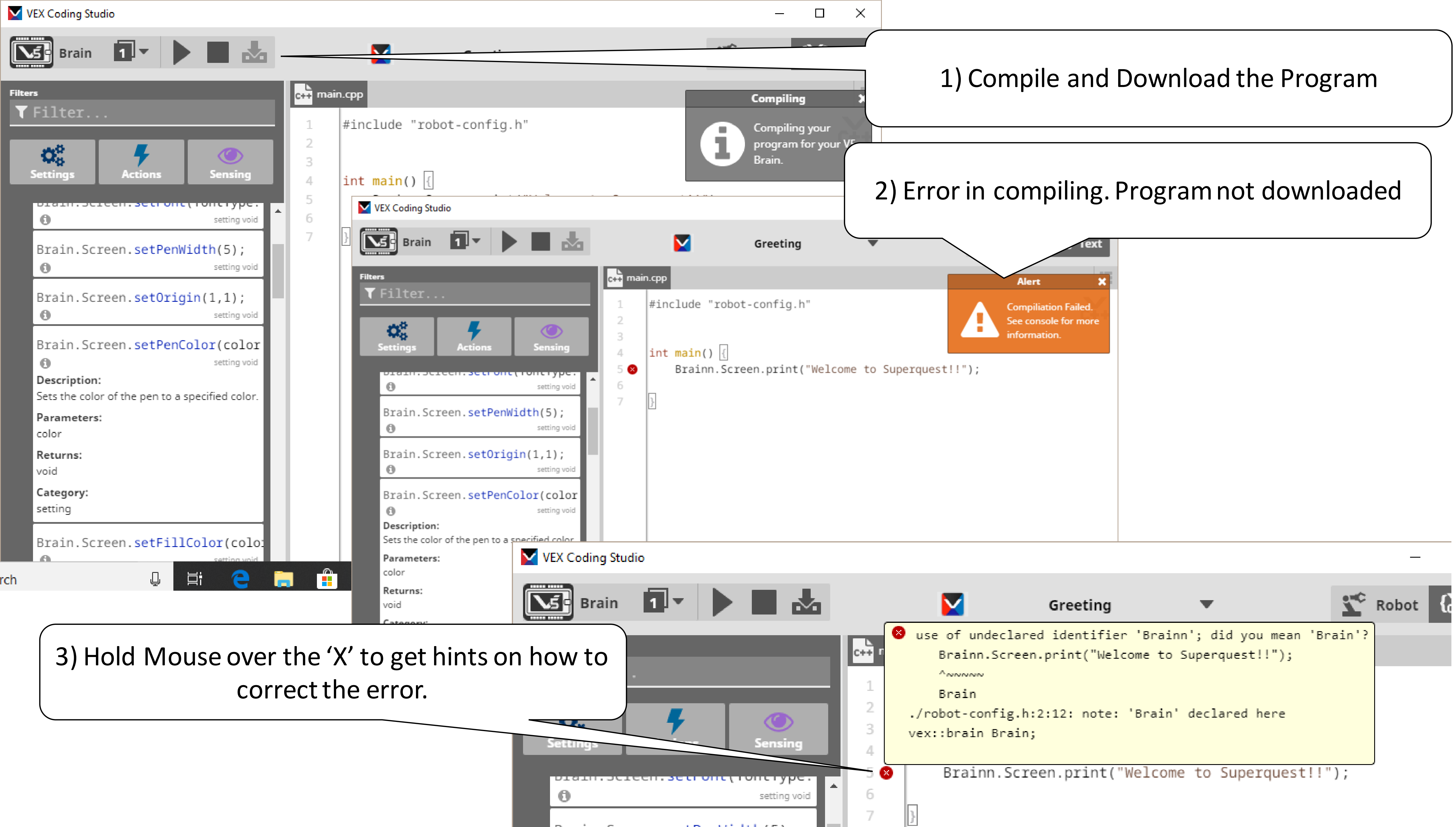
Download the Program to the Brain

Download the Program. This puts a compiled version of this program on the Brain.



```
#include "robot-config.h"


int main() {
    Brainn.Screen.print("Welcome to Superquest!!");

}
```

```
Brain.Screen.setPenWidth(5);
```


VEX Coding Studio
Brain
main.cpp
Filters
Filter...
Settings
Actions
Sensing
Greeting
Robot
Compiling
Compiling your program for your VEX Brain.
Alert
Compilation Failed. See console for more information.

1) Compile and Download the Program

2) Error in compiling. Program not downloaded

3) Hold Mouse over the ‘X’ to get hints on how to correct the error.

```
#include "robot-config.h"


int main() {
    Brainn.Screen.print("Welcome to Superquest!!");

}
```

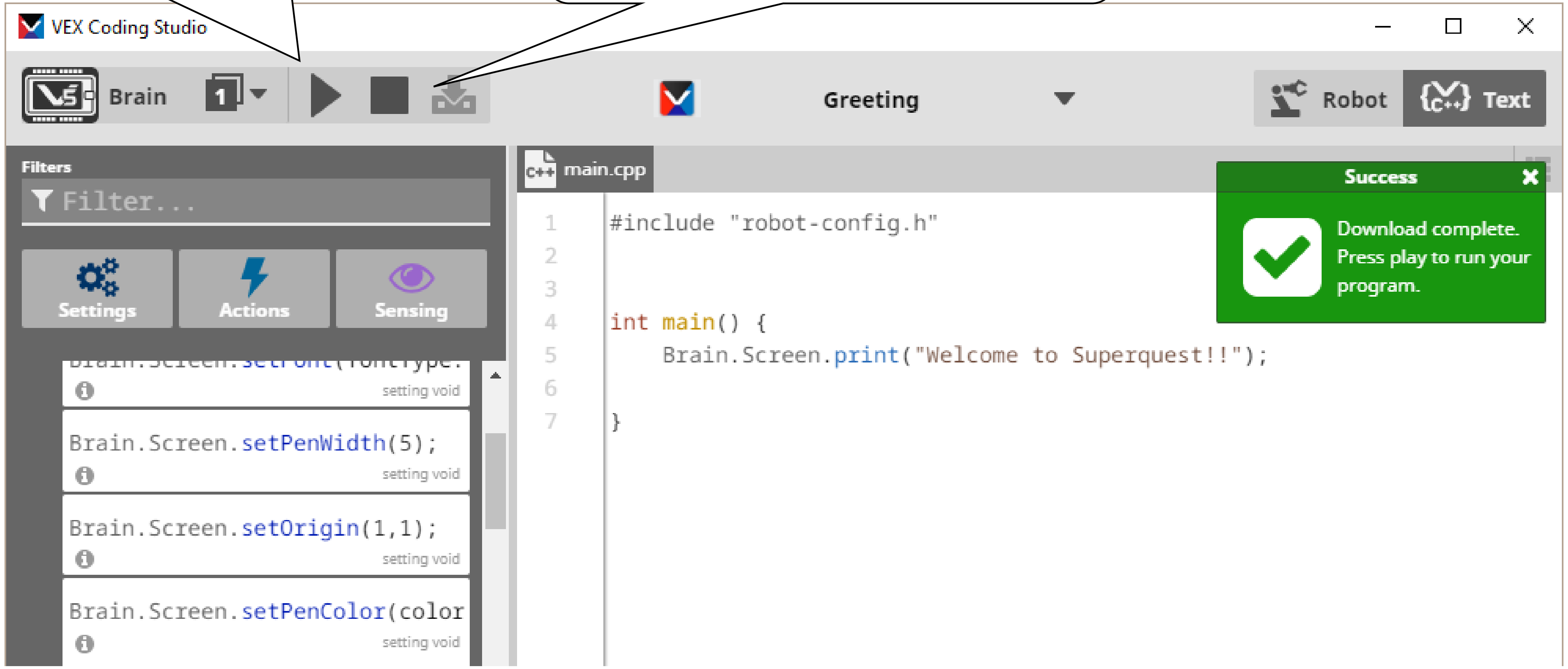
```
use of undeclared identifier 'Brainn'; did you mean 'Brain'?
    Brainn.Screen.print("Welcome to Superquest!!");
    ^~~~~~
    Brain
./robot-config.h:2:12: note: 'Brain' declared here
vex::brain Brain;
```

```
Brain.Screen.setPenWidth(5);
setting void
Brain.Screen.setOrigin(1,1);
setting void
Brain.Screen.setPenColor(color
setting void
Description:
Sets the color of the pen to a specified color.
Parameters:
color
Returns:
void
Category:
setting
Brain.Screen.setFillColor(color
```

Compiled and downloaded Successfully!

1) Play: This will run the last program downloaded to the Brain.

2) Stops the Program that is running on the Brain

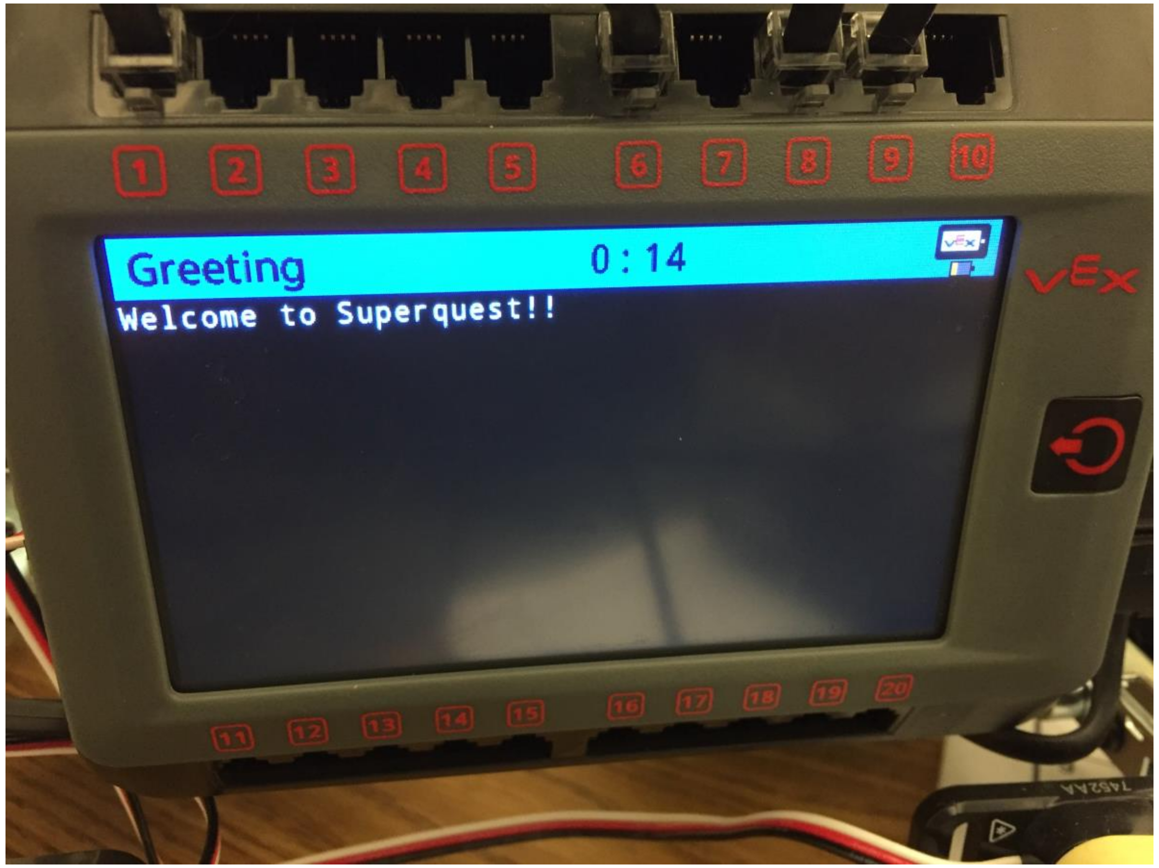


Download complete. Press play to run your program.

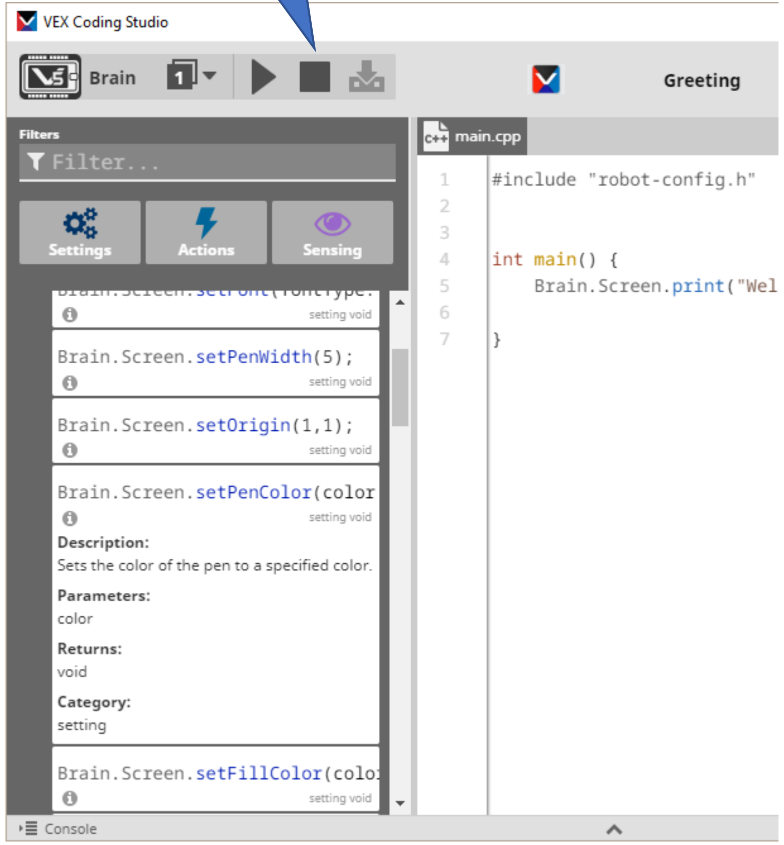
```
#include "robot-config.h"


int main() {
    Brain.Screen.print("Welcome to Superquest!!");

}
```

```
Brain.Screen.setPenWidth(5);
setting void
Brain.Screen.setOrigin(1,1);
setting void
Brain.Screen.setPenColor(color
setting void
```

Running Program



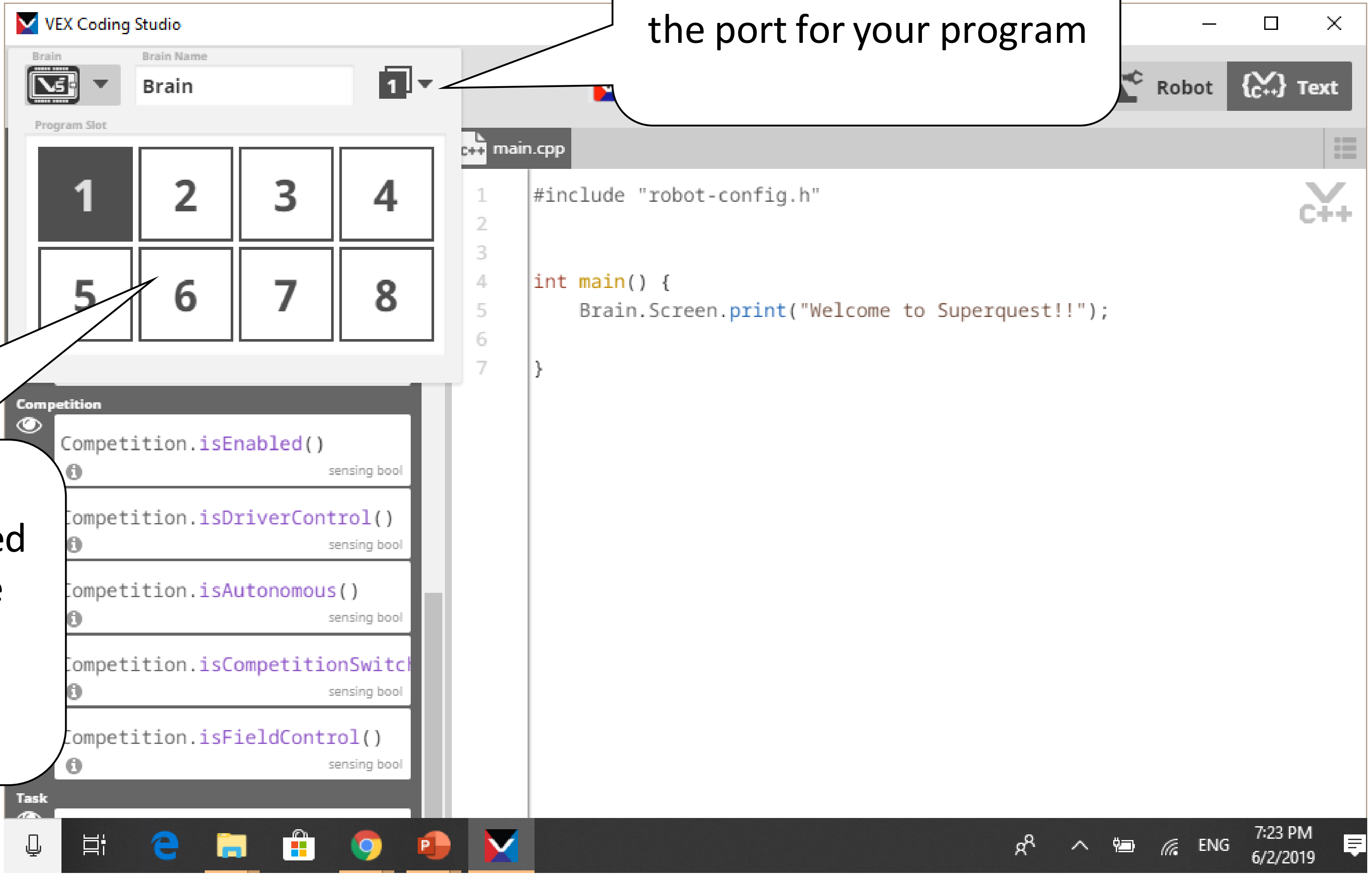
Press the Square to stop the program.



```
Brain.Screen.setPenWidth(5);
setting void
Brain.Screen.setOrigin(1,1);
setting void
Brain.Screen.setPenColor(color
setting void
Description:
Sets the color of the pen to a specified color.
Parameters:
color
Returns:
void
Category:
setting
Brain.Screen.setFillColor(colo
setting void
```

```
#include "robot-config.h"


int main() {
    Brain.Screen.print("Wel

}
```

More on Downloading: The Brain can store 8 programs



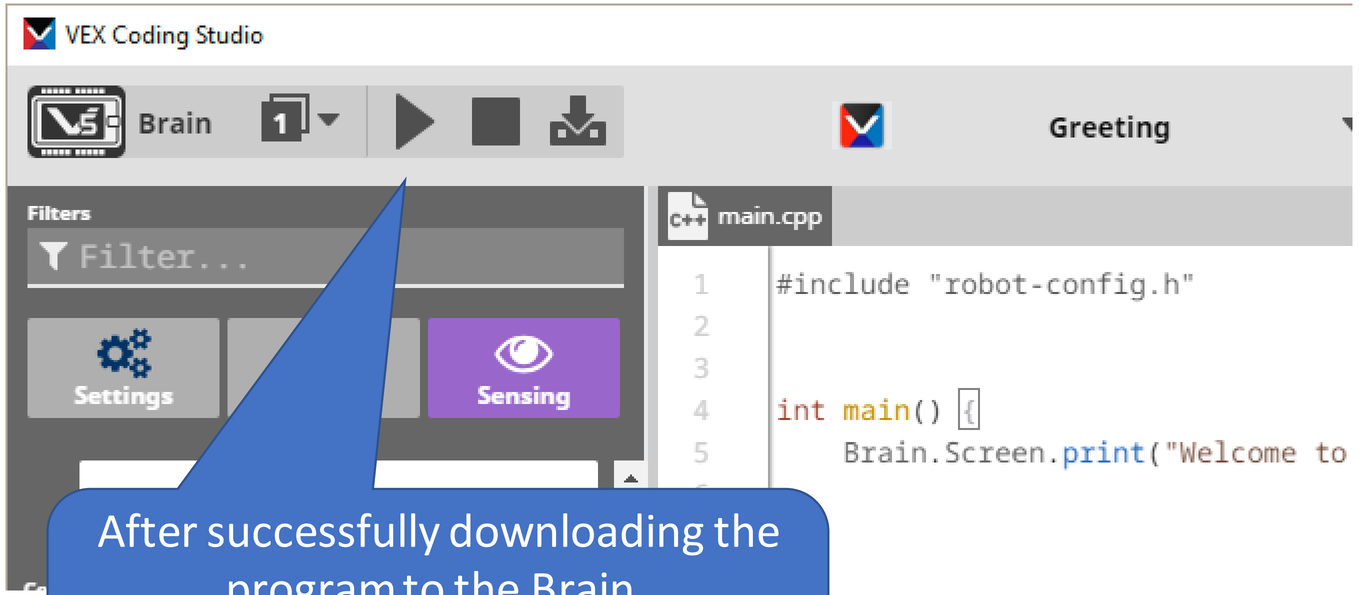
```
#include "robot-config.h"


int main() {
    Brain.Screen.print("Welcome to Superquest!!");

}
```

```
Competition.isEnabled()
sensing bool
Competition.isDriverControl()
sensing bool
Competition.isAutonomous()
sensing bool
Competition.isCompetitionSwitch
sensing bool
Competition.isFieldControl()
sensing bool
```

Pull down menu to select the port for your program

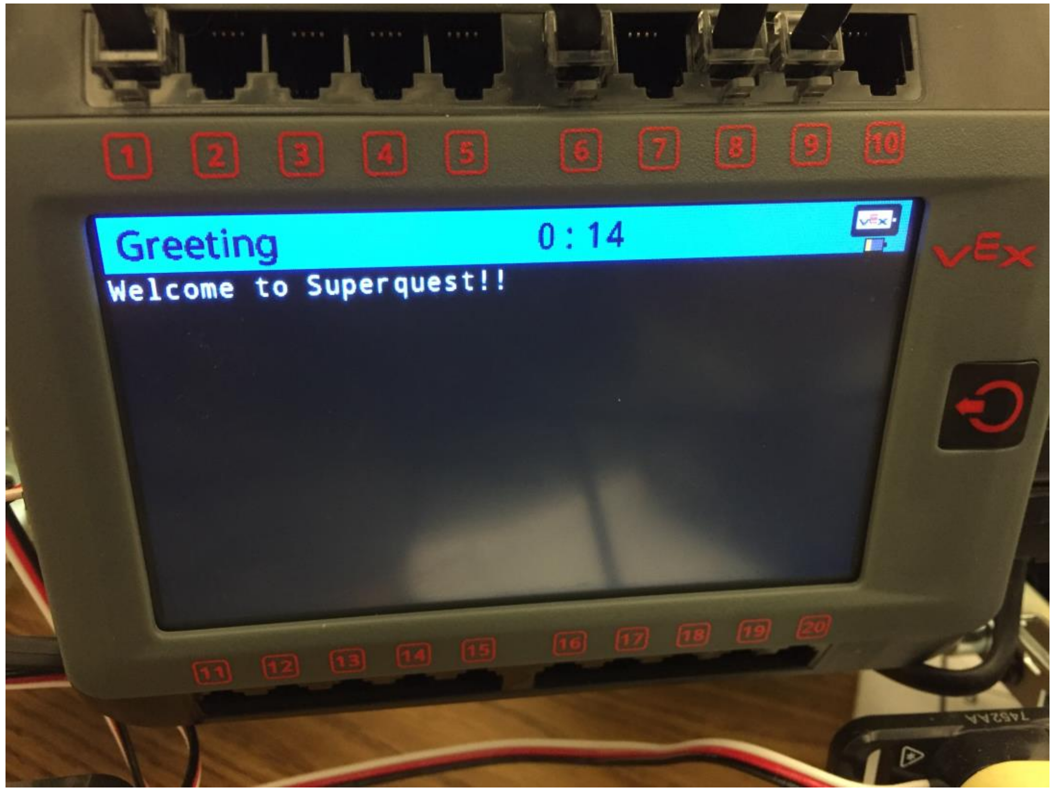
Select the port: When downloaded it is saved into this port. If there was a program in this port it is overwritten.

Run the Program from the Computer. Brain must be attached to the computer



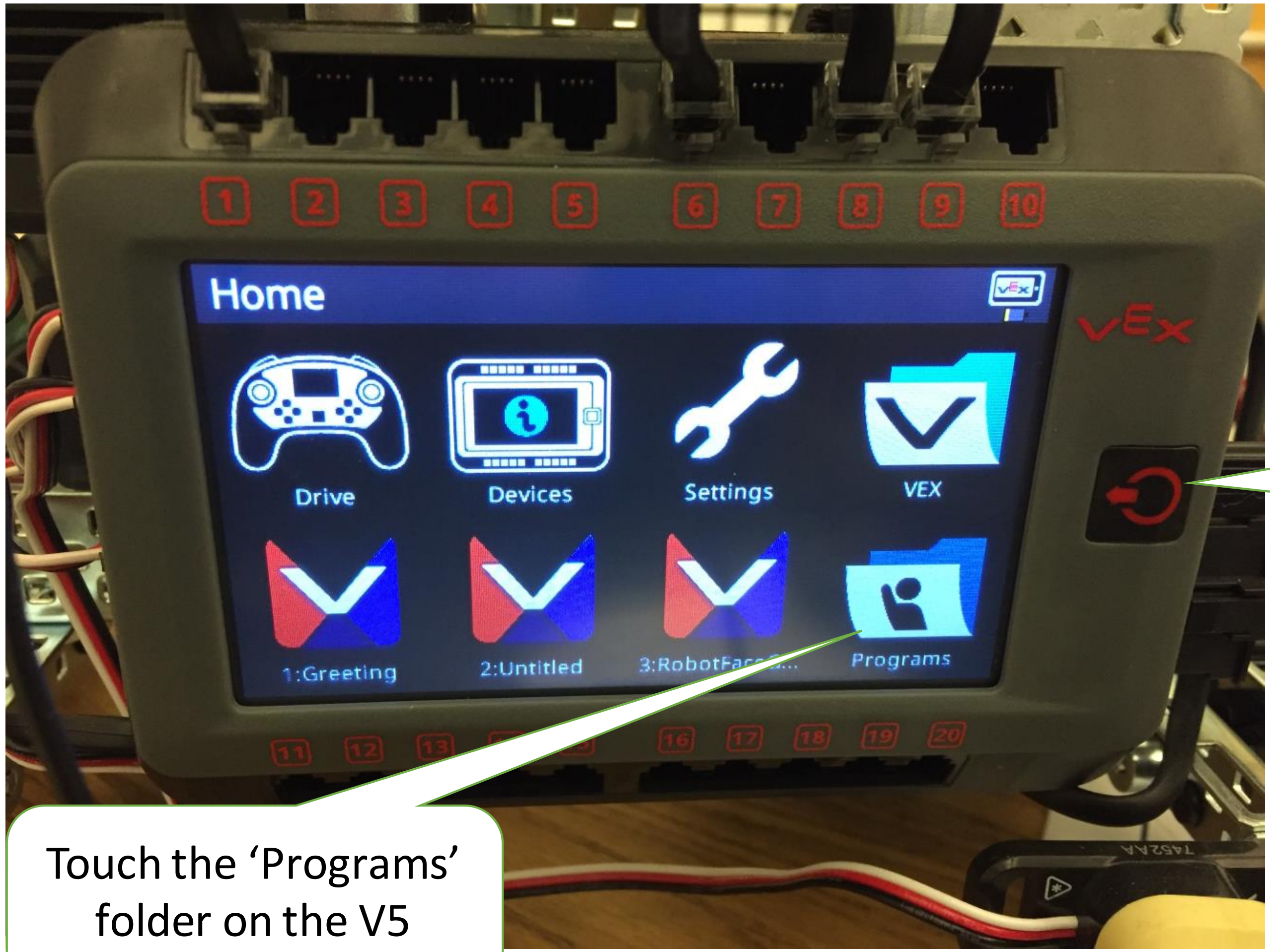
```
#include "robot-config.h"


int main() {
    Brain.Screen.print("Welcome to
```

After successfully downloading the program to the Brain.
Hit the Play button to run the program



```
Greeting                0:14
Welcome to Superquest!!
```

Running the program from V5 Brain. Does not have to be connected to the computer.

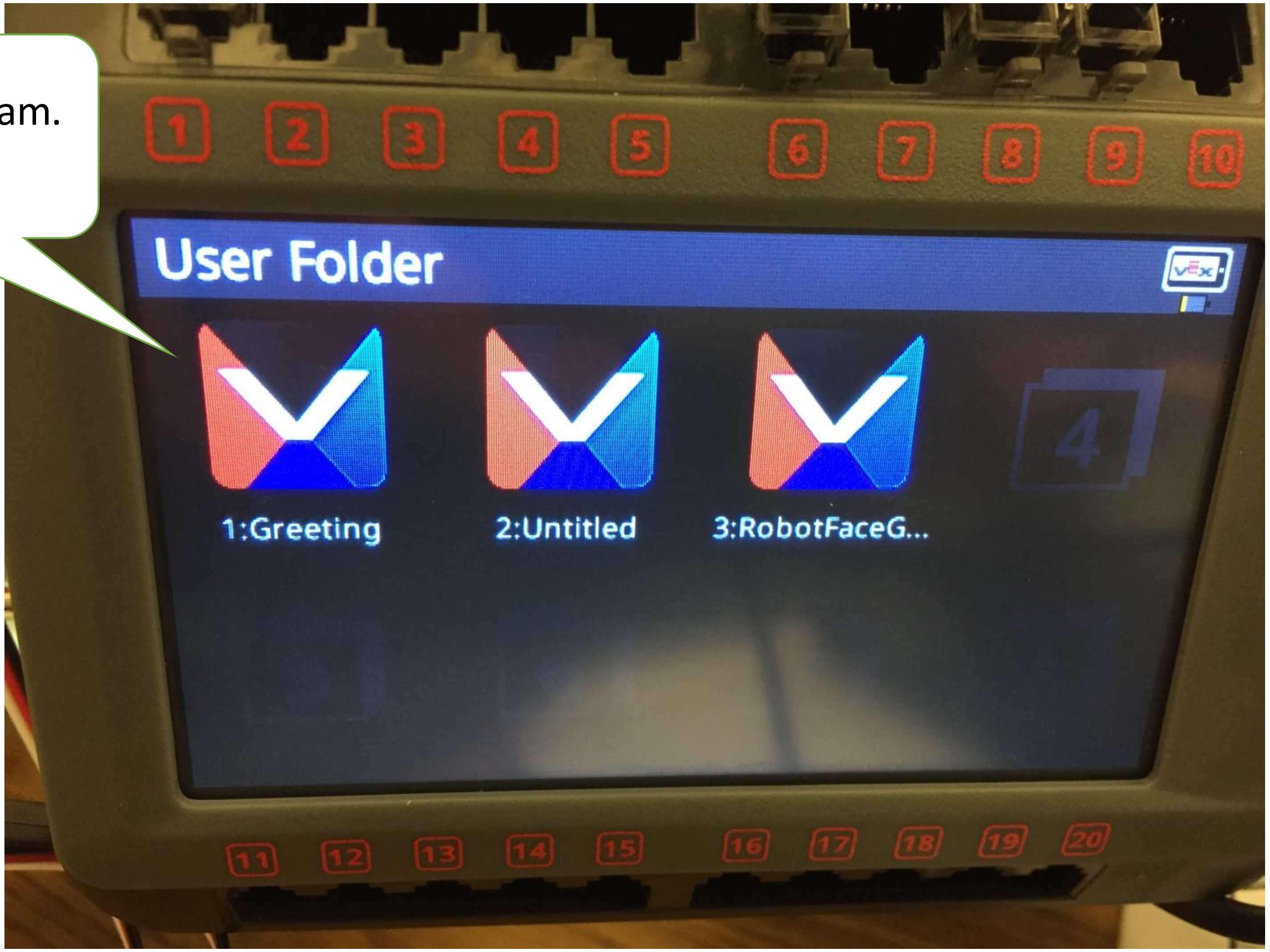


If there is a different display on the screen, you can click on the V5 Power Button to back out of the current screen

Touch the ‘Programs’ folder on the V5 display.

Select the program you want to run

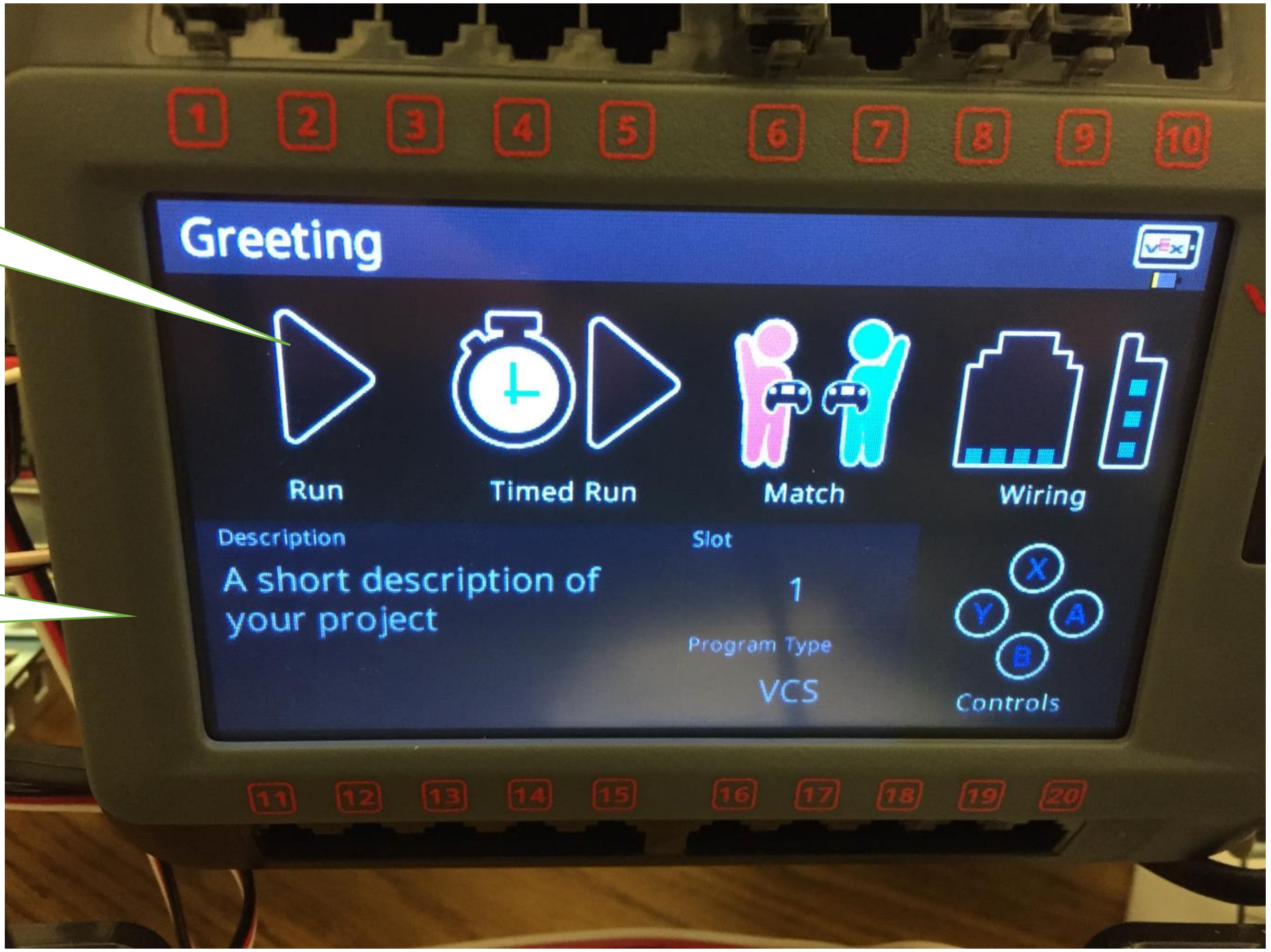
Click on the program.



Click the Triangle to run the program

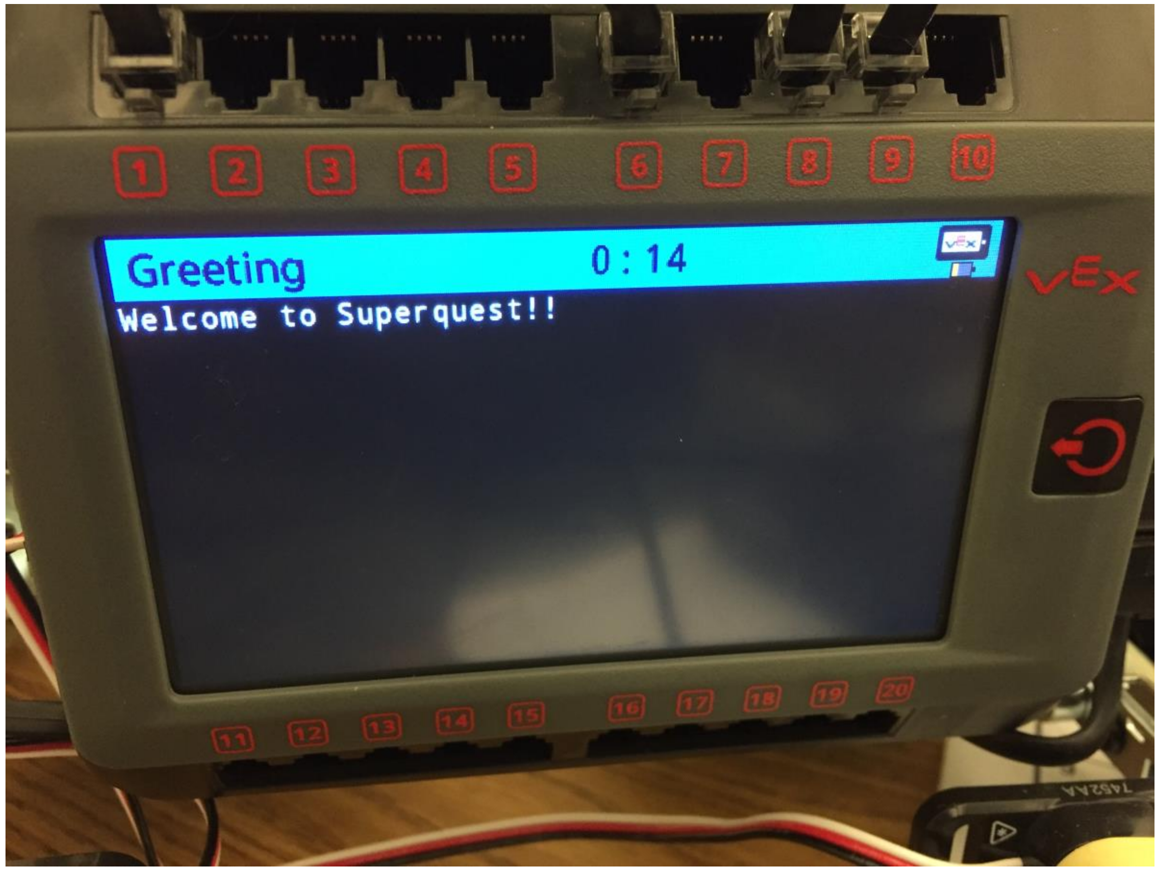
Click on ‘Run’ to run the program

Description can be set when saving the program.

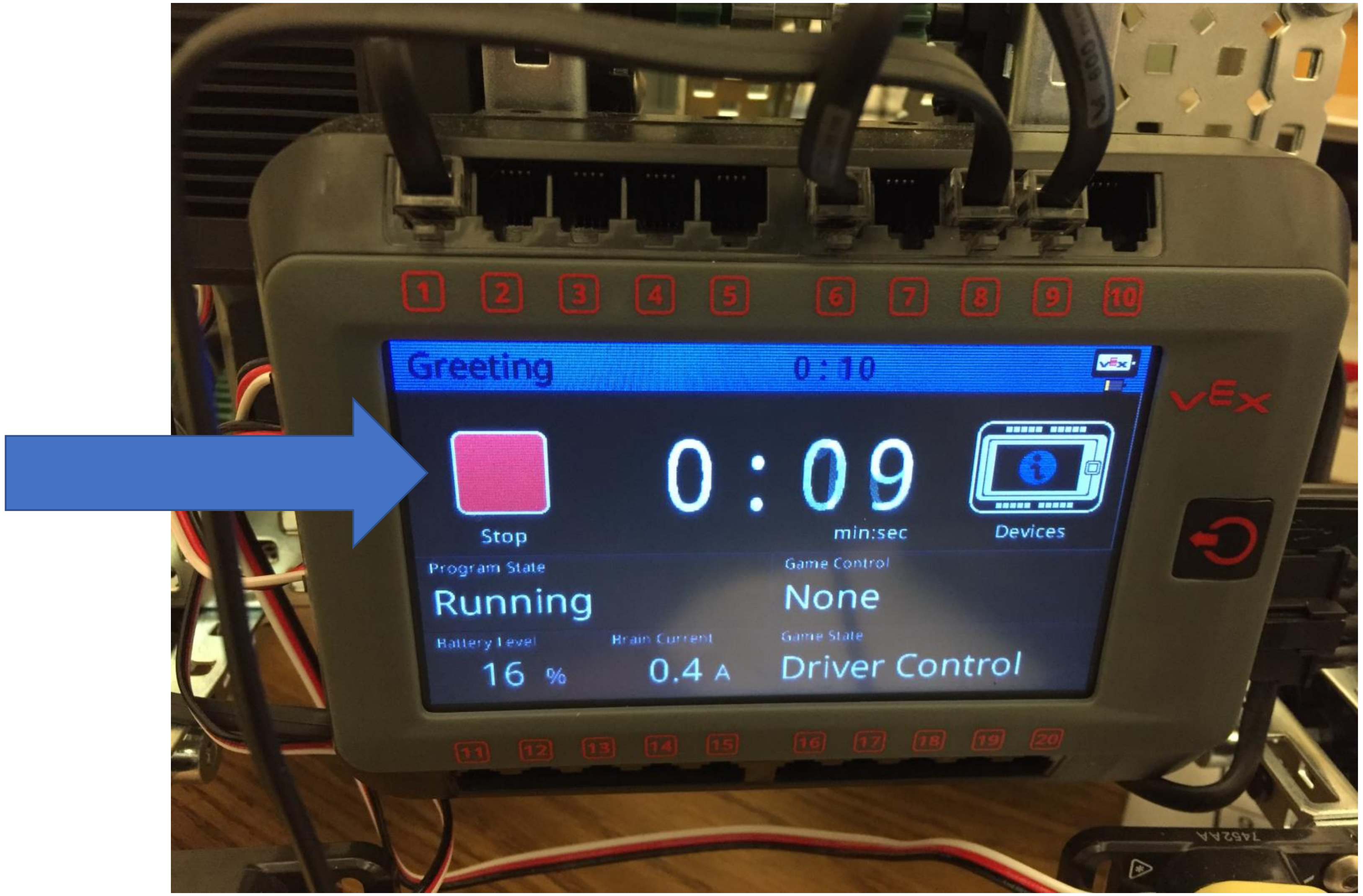


Running Program

- Can press the Power On button to move back to the menu



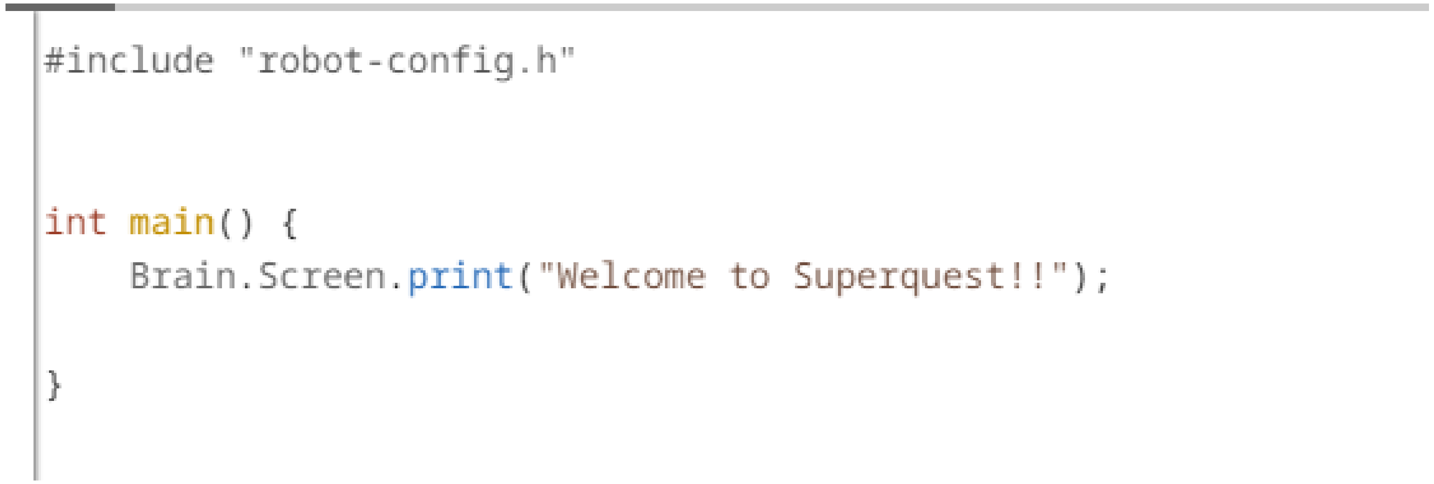
Press the ‘Stop’ button to stop the program



Coding Syntax Notes

<u>Code</u>, <u>compile</u>, <u>download</u> and <u>run</u> a program. You can use the code to the right or experiment with other **<u>Brain</u>** commands.

1. Punctuation
   1. () Used for the arguments sent to the command
   2. ; Marks the end of a command
   3. “” Surrounds the Characters that will show up on the screen
   4. // For a one line comment
   5. /* */ Multi line comment
   6. Brain.Screen.print

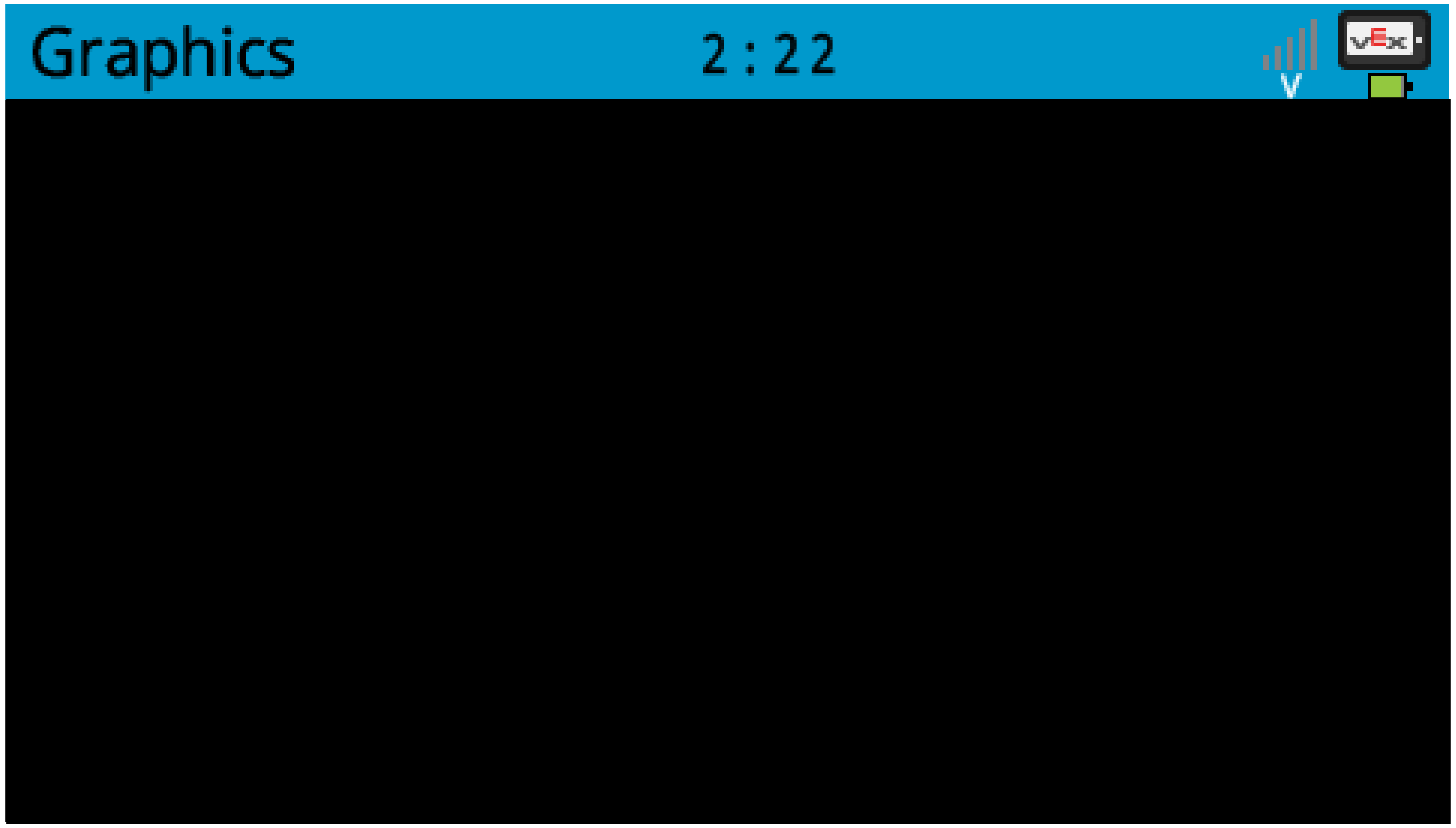


```
#include "robot-config.h"


int main() {
    Brain.Screen.print("Welcome to Superquest!!");

}
```

Adding Graphics on the Screen: Dimensions

- The Screen has input and output

(0, 0)

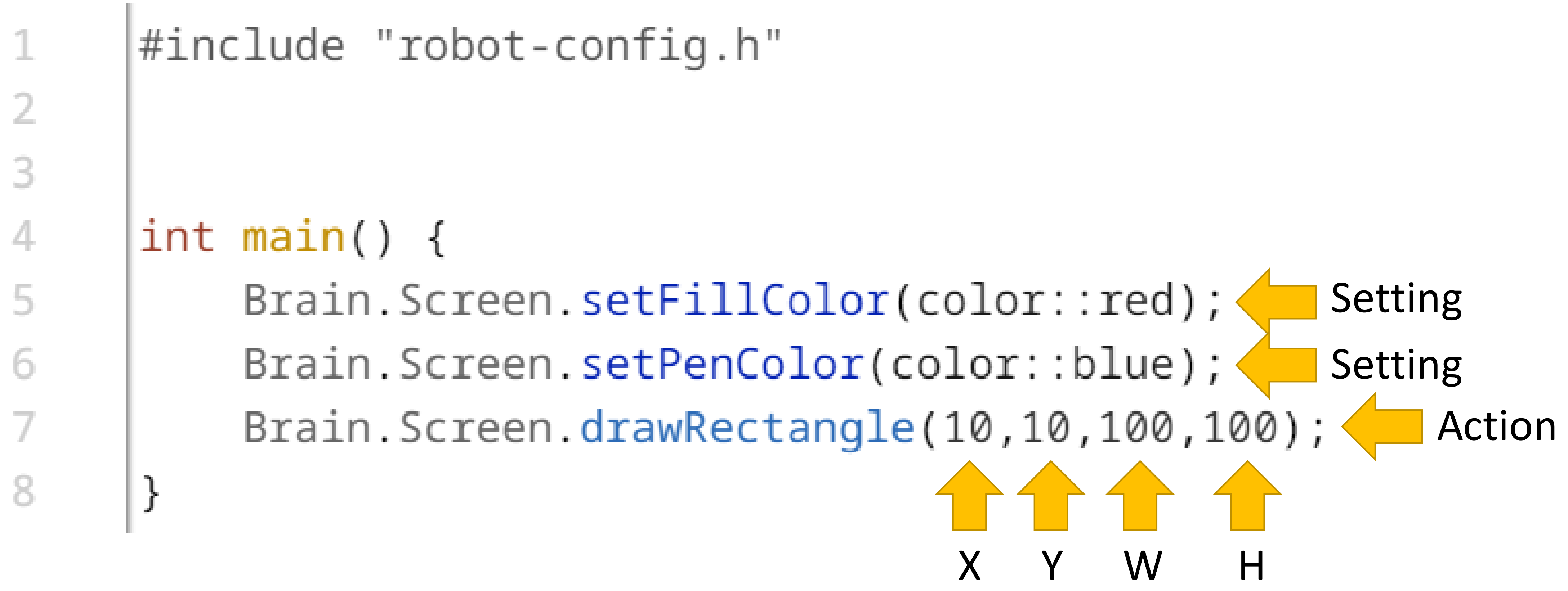


(479,0)

(0, 239)

(479,239)

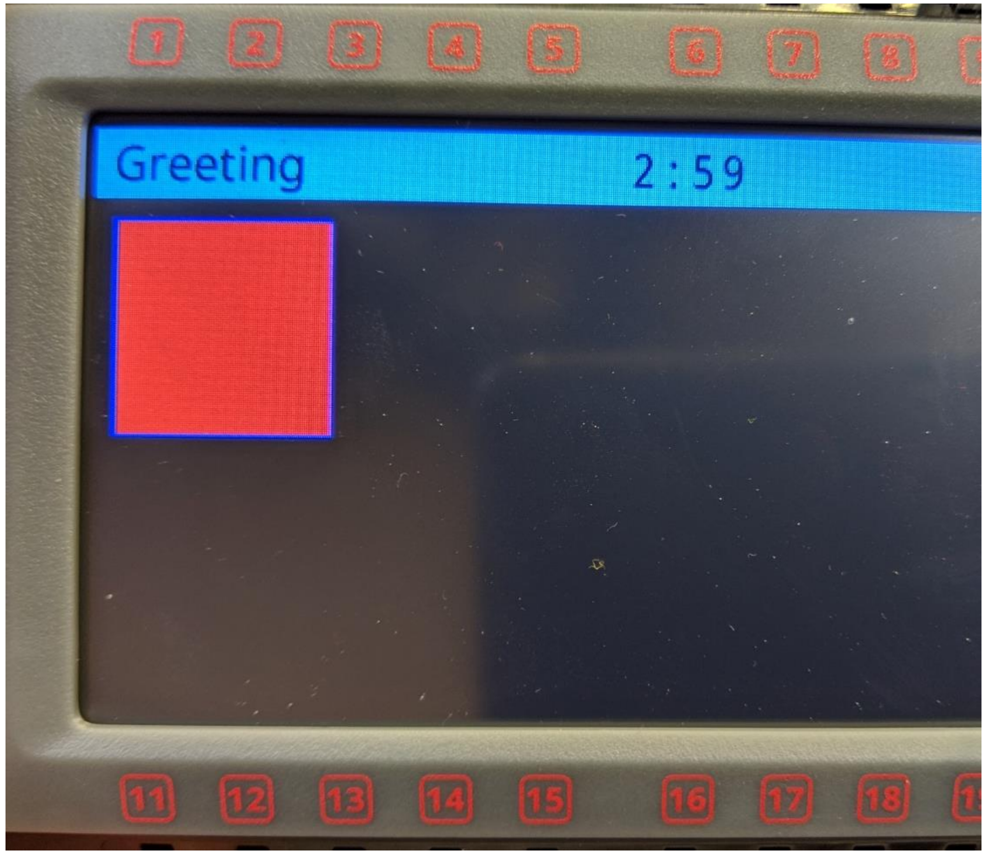
Programming Example: Putting Graphics on the Screen



```
#include "robot-config.h"


int main() {
    Brain.Screen.setFillColor(color::red);
    Brain.Screen.setPenColor(color::blue);
    Brain.Screen.drawRectangle(10,10,100,100);
}
```

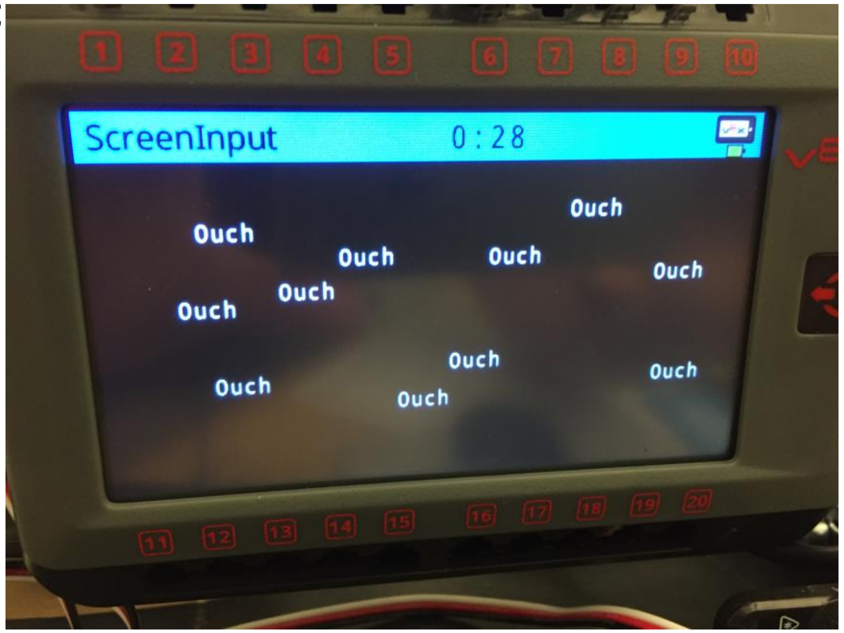
Save, Download Run & Observe!

- Save your program (YourNameGraphics) , download it to the robot, and observe the behavior.
- Feel free to change the values in your program to observe how those changes affect the output.
    - Remember, any time you make changes to your program, you must download them to the robot for it to take effect.
- To stop the program execution, you can press the physical button on the Robot Brain, followed by the Stop button on the screen.



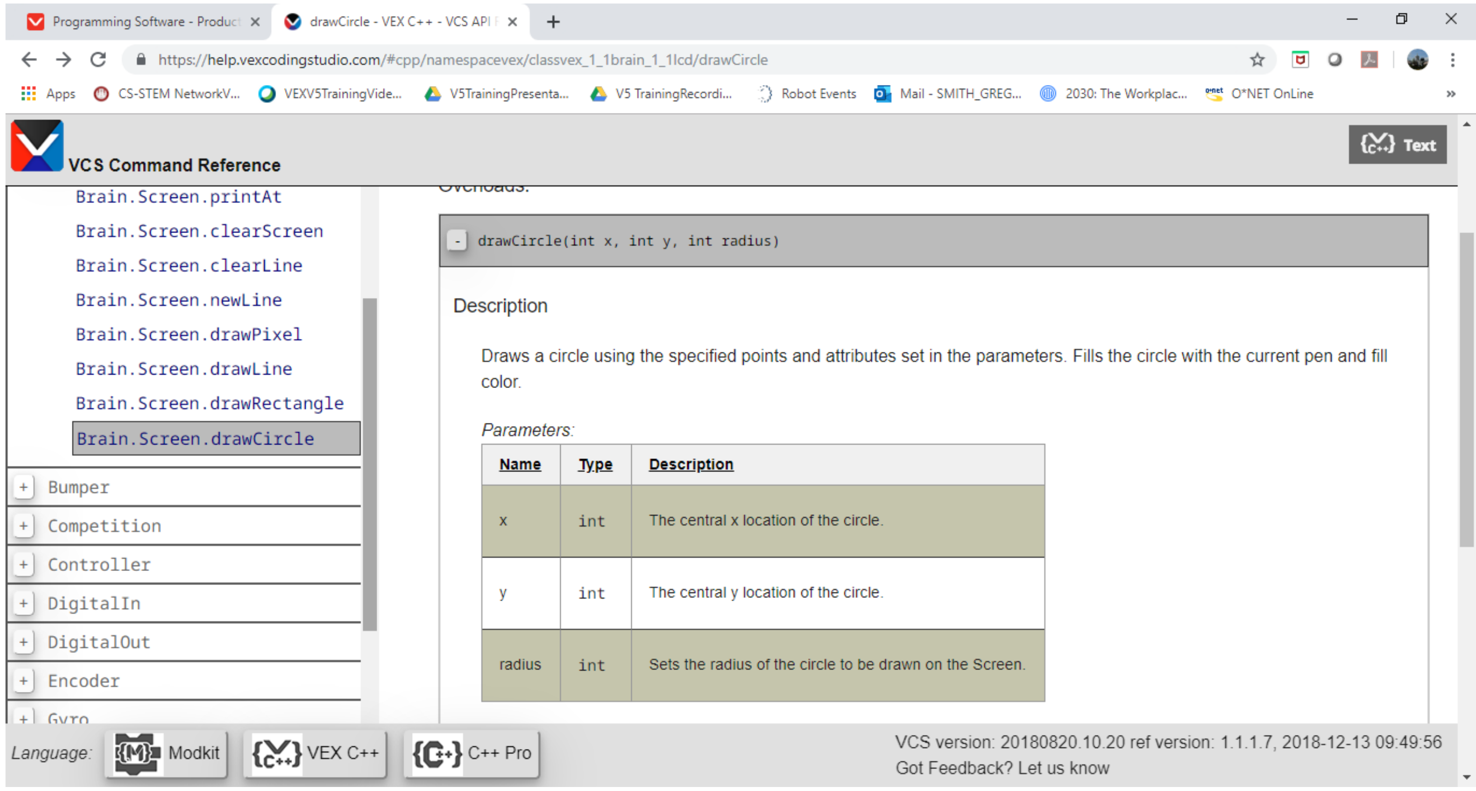
Getting input from the touch screen

```
#include "robot-config.h"
int main() {
    while (true)
    {
        while (!Brain.Screen.pressing())  //While NOT (!) the screen is being pressed
        {};  //Do nothing
        //Say 'Ouch' where the screen was pressed
        Brain.Screen.printAt(Brain.Screen.xPosition(),Brain.Screen.yPosition(),"Ouch");
    } //Go back to  while (true) to repeat forever
    Brain.Screen.print("Good bye");//Why will this line of code never be reached?
}
```



Resources/notes

- *Help.vexcodingstudio.com*
  - Vex Coding Studio Command Reference
  - Gives a description of how the commands work, often with sample code.



```
drawCircle(int x, int y, int radius)
```

Description

Draws a circle using the specified points and attributes set in the parameters. Fills the circle with the current pen and fill color.

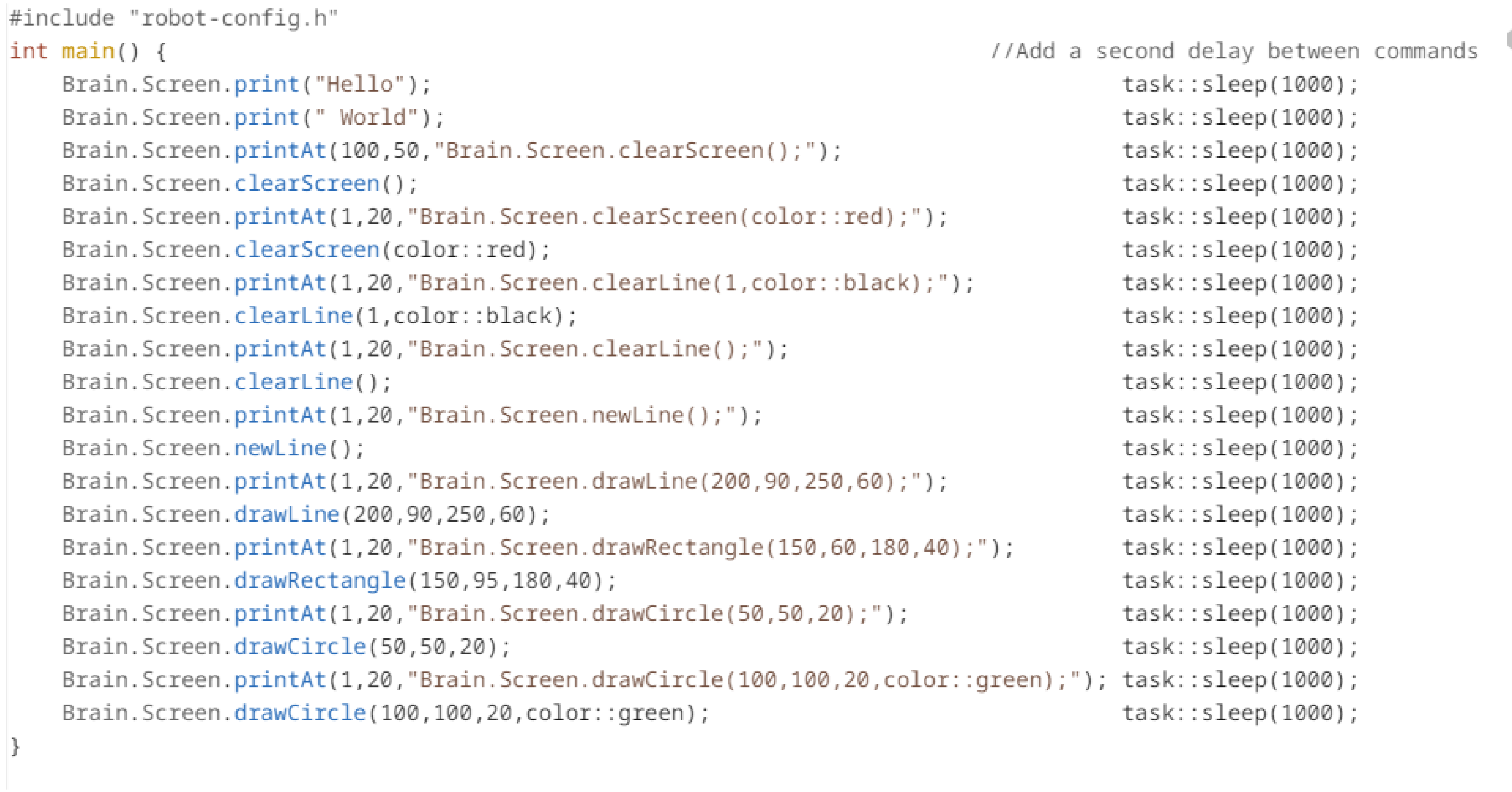
Parameters:

Name	Type	Description
x	int	The central x location of the circle.
y	int	The central y location of the circle.
radius	int	Sets the radius of the circle to be drawn on the Screen.

VCS version: 20180820.10.20 ref version: 1.1.1.7, 2018-12-13 09:49:56
Got Feedback? Let us know

Brain Programming Activity

- Use what you have learned to draw a Robot Face.
- <u>**Face Option**</u>
  - Be as creative as you would like but it must have…
  - At least 2 eyes, 1 mouth, 1 head and 3 colors
- <u>**Robot Option**</u>
  - Draw a robot
    - At least 2 Wheels
    - One figure for the Drive train
    - One Tower/arm
    - Three Colors
- Hints:
  - Draw it out first. (480 by 240)
  - Use the panel on the left to look up other Brain.Screen. Options.
- Extensions:
  - Get input from the touch screen to interact with your image
    - Shows until touched
    - Adds a new part of the drawing each time the screen is touched
    - Draws an image at each location touched
    - …



```
#include "robot-config.h"
int main() {                                                                      //Add a second delay between commands
    Brain.Screen.print("Hello");                                                      task::sleep(1000);
    Brain.Screen.print(" World");                                                     task::sleep(1000);
    Brain.Screen.printAt(100,50,"Brain.Screen.clearScreen();");                       task::sleep(1000);
    Brain.Screen.clearScreen();                                                       task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.clearScreen(color::red);");               task::sleep(1000);
    Brain.Screen.clearScreen(color::red);                                             task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.clearLine(1,color::black);");             task::sleep(1000);
    Brain.Screen.clearLine(1,color::black);                                           task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.clearLine();");                           task::sleep(1000);
    Brain.Screen.clearLine();                                                         task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.newLine();");                             task::sleep(1000);
    Brain.Screen.newLine();                                                           task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.drawLine(200,90,250,60);");               task::sleep(1000);
    Brain.Screen.drawLine(200,90,250,60);                                             task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.drawRectangle(150,60,180,40);");          task::sleep(1000);
    Brain.Screen.drawRectangle(150,95,180,40);                                        task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.drawCircle(50,50,20);");                  task::sleep(1000);
    Brain.Screen.drawCircle(50,50,20);                                                task::sleep(1000);
    Brain.Screen.printAt(1,20,"Brain.Screen.drawCircle(100,100,20,color::green);"); task::sleep(1000);
    Brain.Screen.drawCircle(100,100,20,color::green);                                 task::sleep(1000);
}
```

RobotC to Vex Coding Studio

- Website to help transition from RobotC to VEX Coding Studio
  - https://education.vex.com/eduvex/parent-wrapper.php?id=robotc-vcs
  - Gives sample code written in RobotC and VEX Coding Studio
- Webinar for transitioning from RobotC to VEX Coding Studio
  - https://www.cmu.edu/roboticsacademy/Training/Online/Webinar-rc-vcs.html