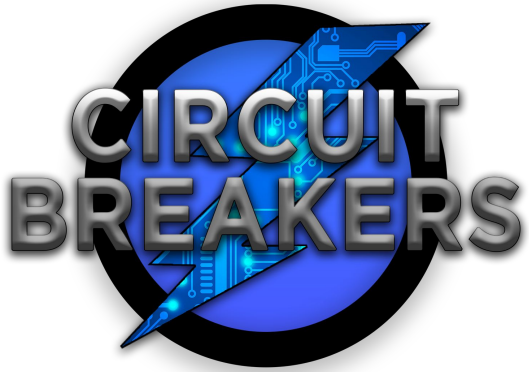
FTC Kickoff 2022

Getting Started with JAVA Programming

Android Studio setup and Github



10435 Circuit Breakers

Presenters

10435

Samanyu Duvvuri: 3rd year of FTC

Evin Jerry: 3rd year of FTC

Kalya Kothavale: 3rd year of FTC

Revanth Surabhi: 3rd year of FTC

Aditya Pethe: 2nd year of FTC

Start Here - New FTC docs site

https://ftc-docs.firstinspires.org/

“Getting Started” section has links for:

- Rookie Teams
- Veteran Teams

Click on the programming resources on the left-hand side. Both of the sections mentioned above link to the same page. Here you will find “Android Studio Tutorial”

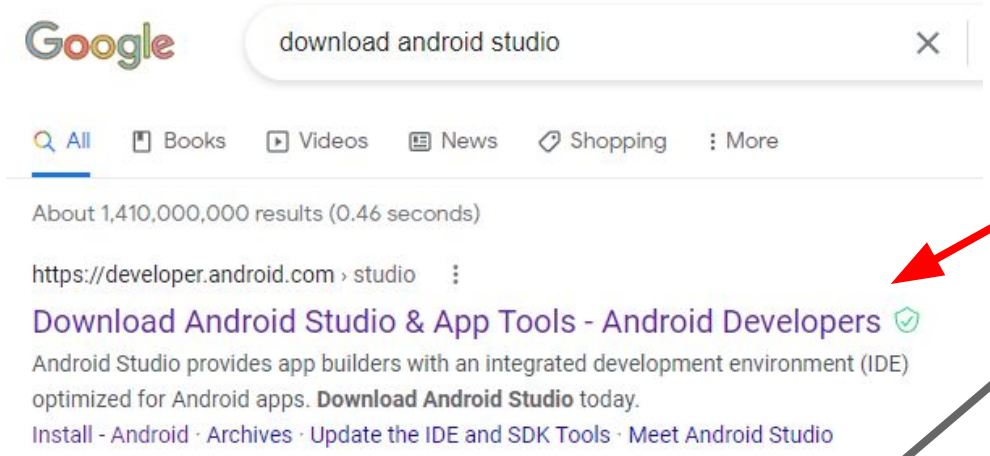
FTC 7.2 SDK Release Announcement

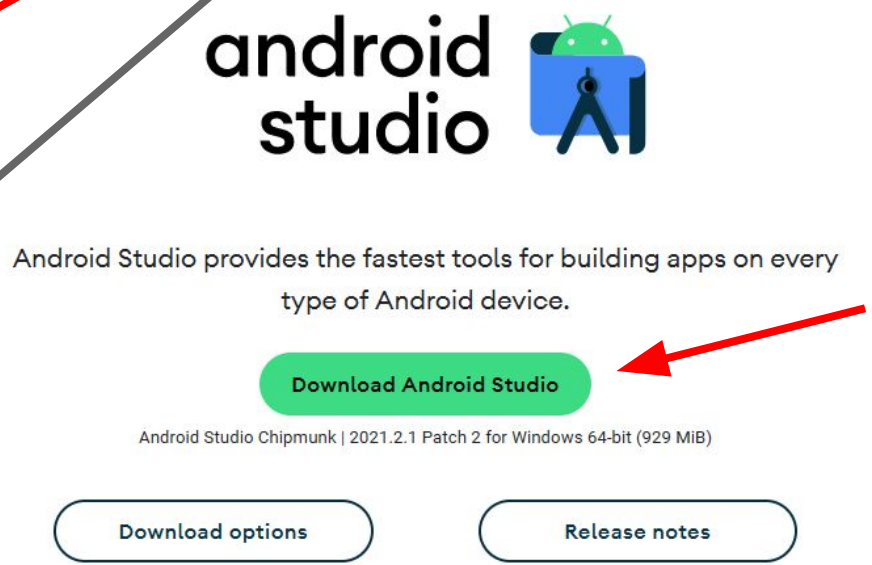
https://firsttechchallenge.blogspot.com/2022/08/first-tech-challenge-72-sdk-release.html

Highlights:

- Requires Android Studio “Chipmunk”
- Driver Station apps must be 7.2.
- Be sure to view the Readme.md in github
- Many improvements

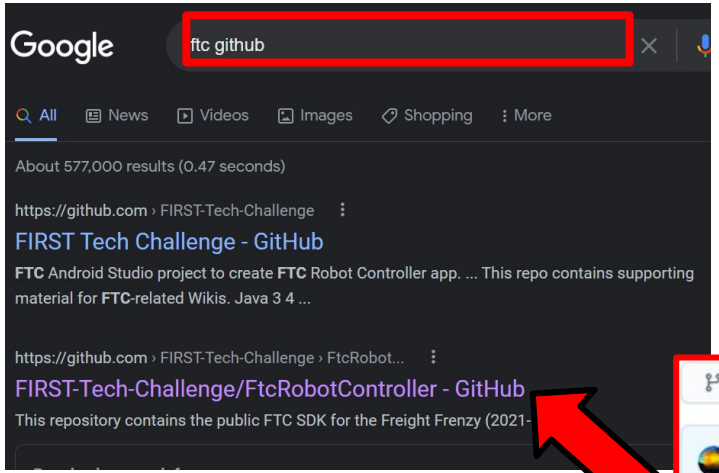
Setting Up Android Studio

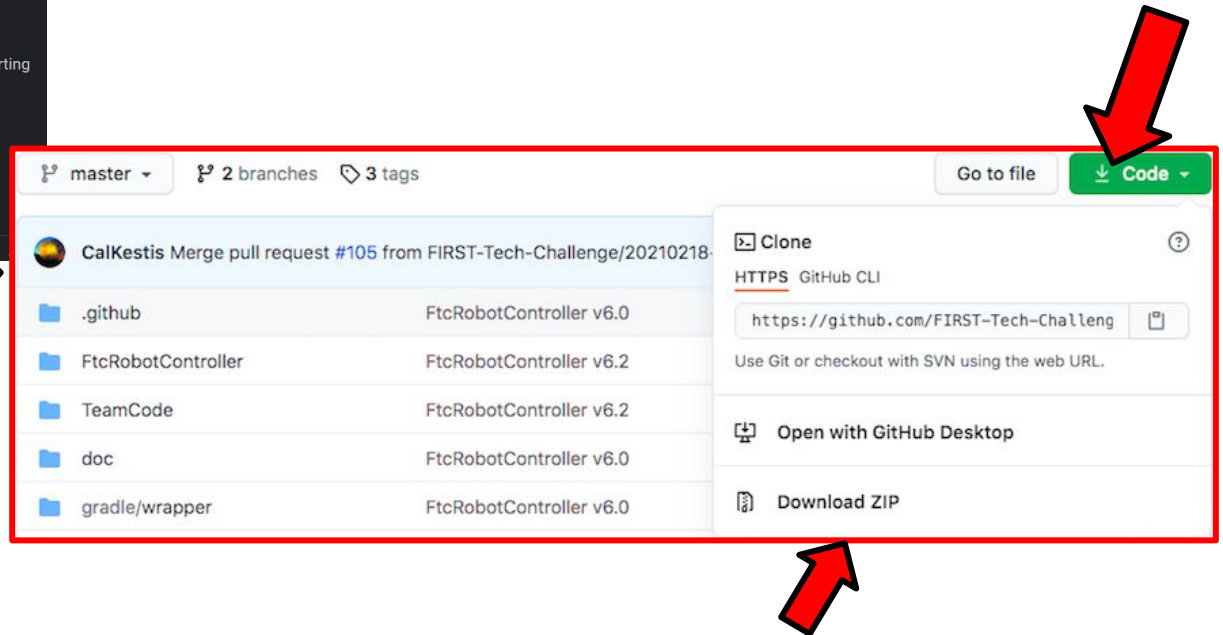




CIRCUIT BREAKERS 10435

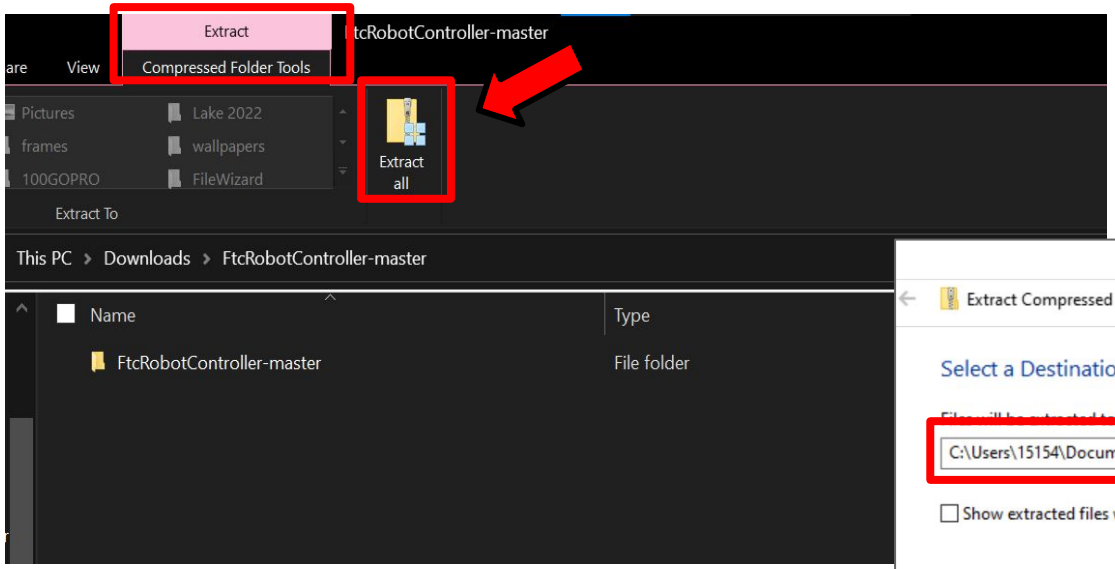
Installing the FtcRobotController SDK

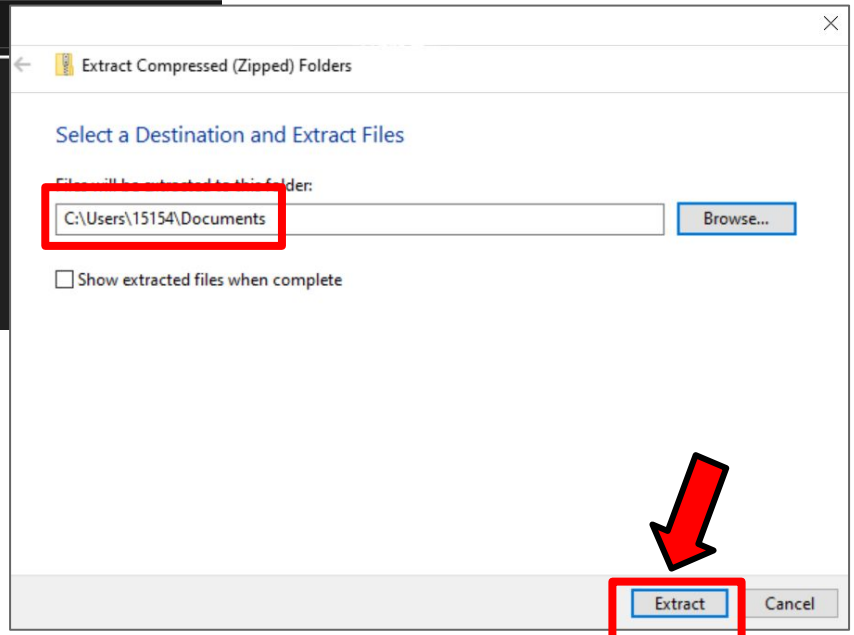




CIRCUIT BREAKERS 10435

Extracting the ZIP

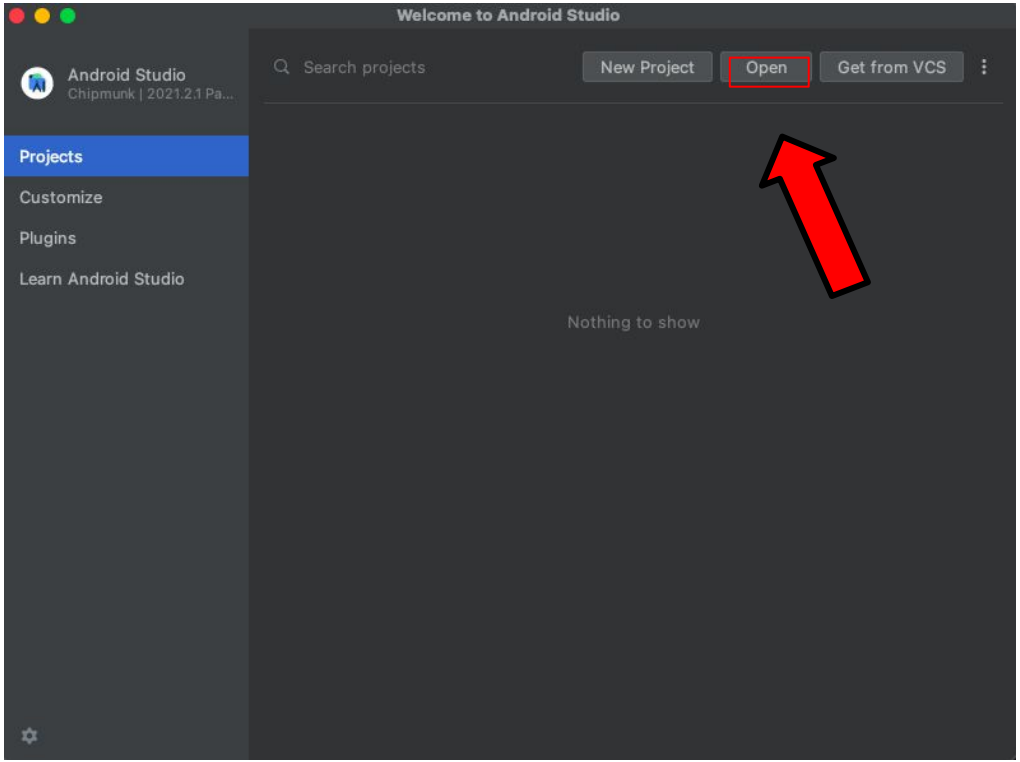


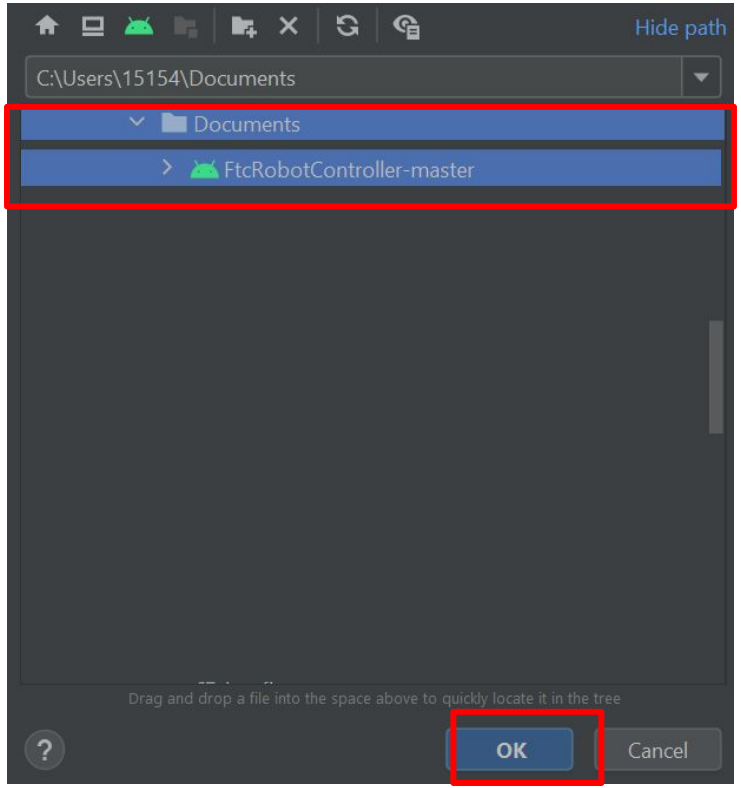


CIRCUIT BREAKERS 10435

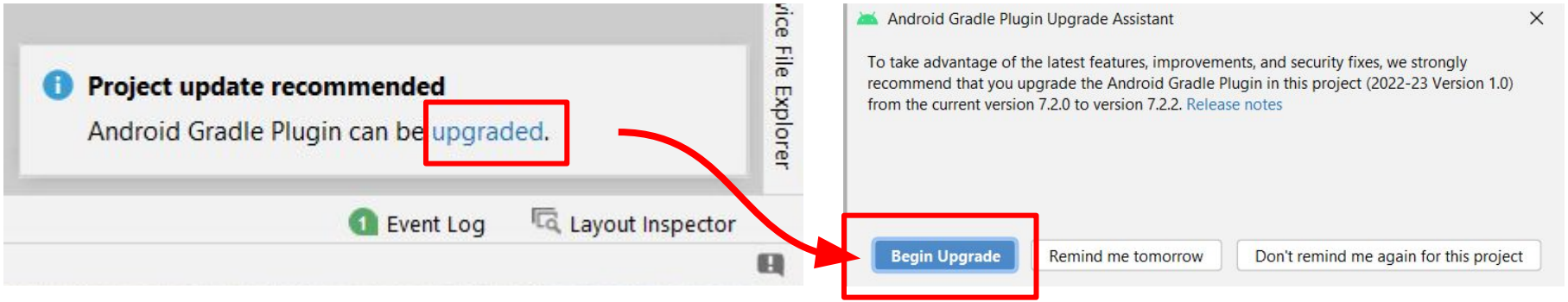
Opening the FTC SDK

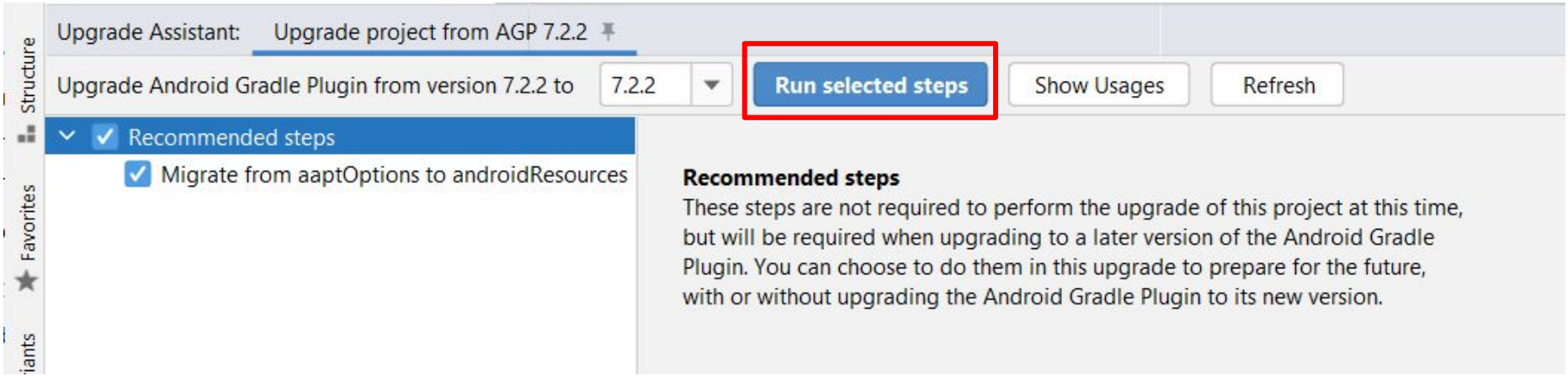
10435



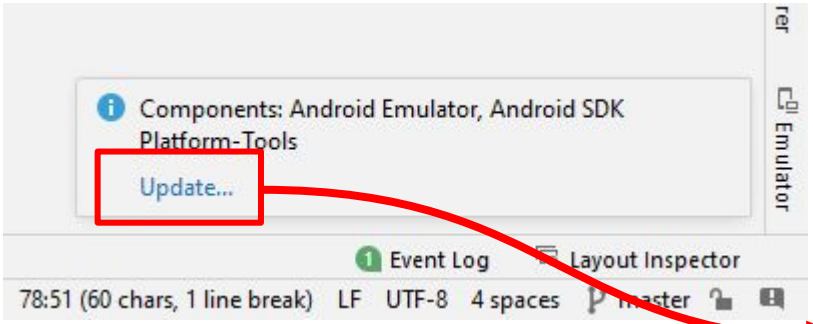


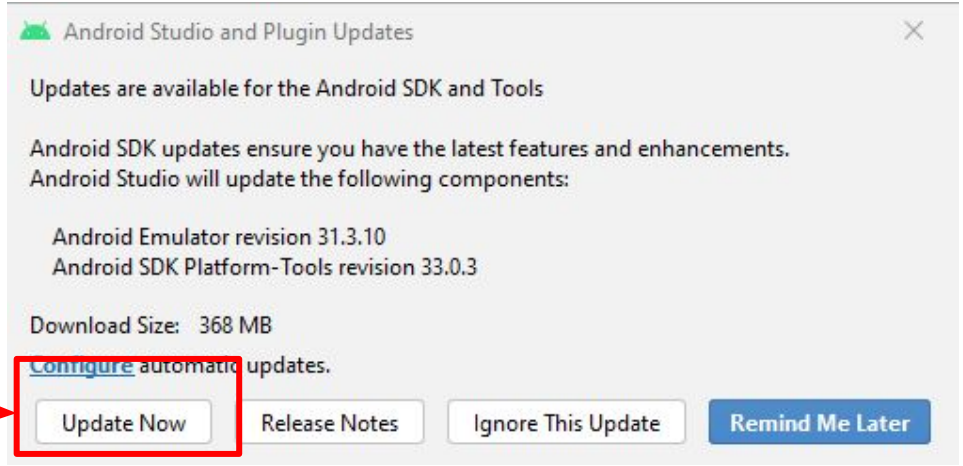
Update Messages



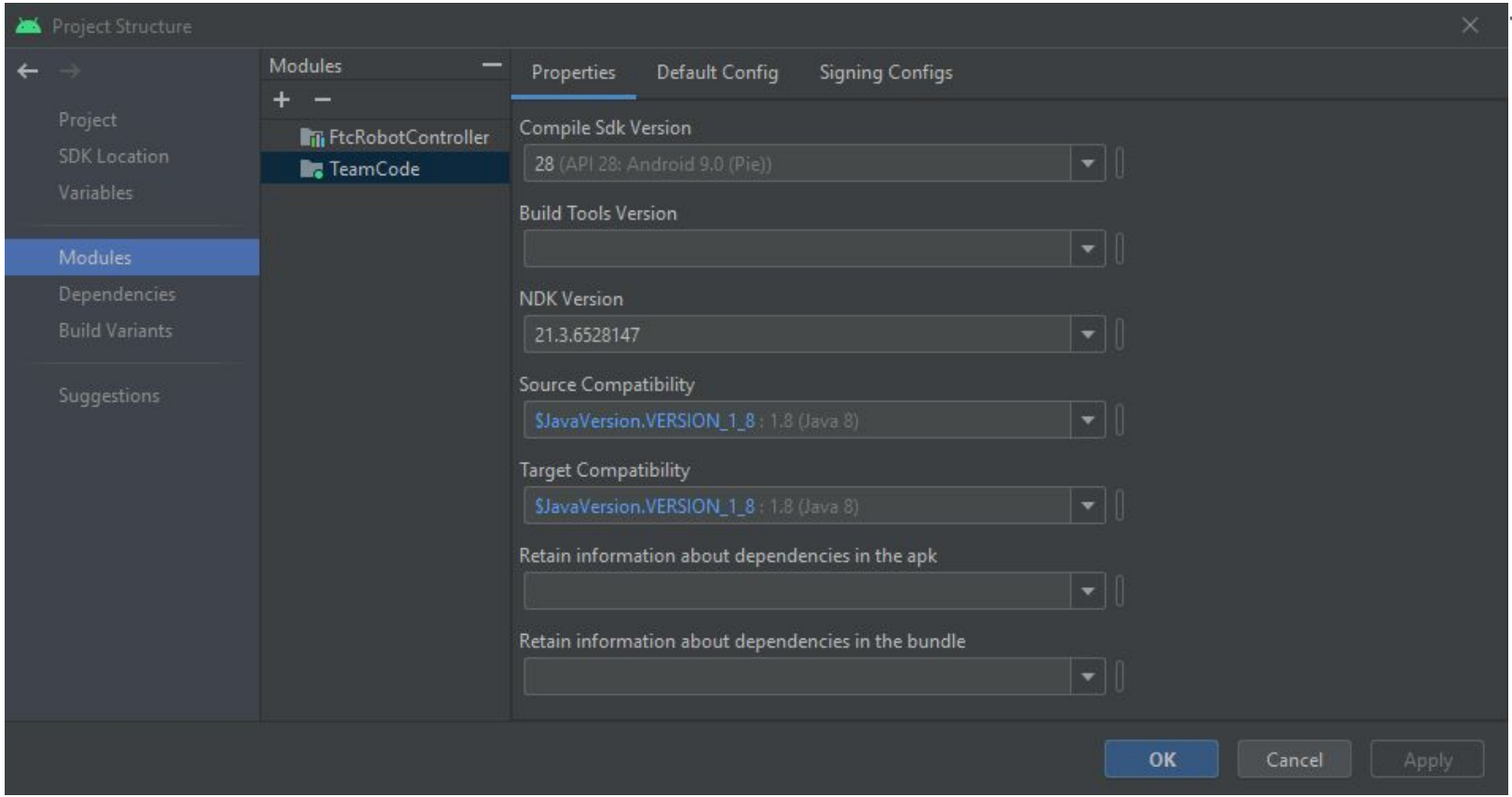


Update Messages



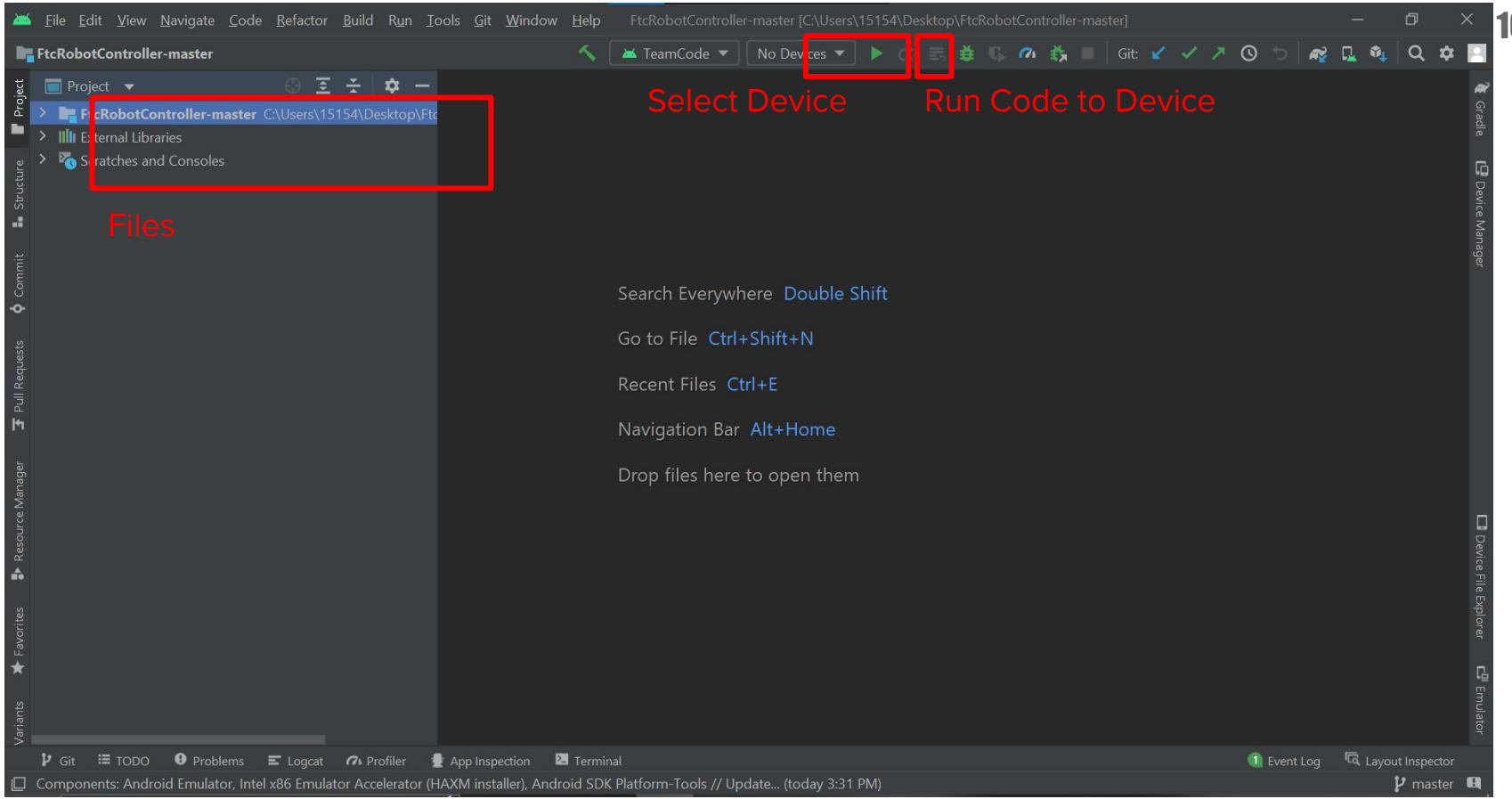


Compile SDK Version



CIRCUIT BREAKERS
10435

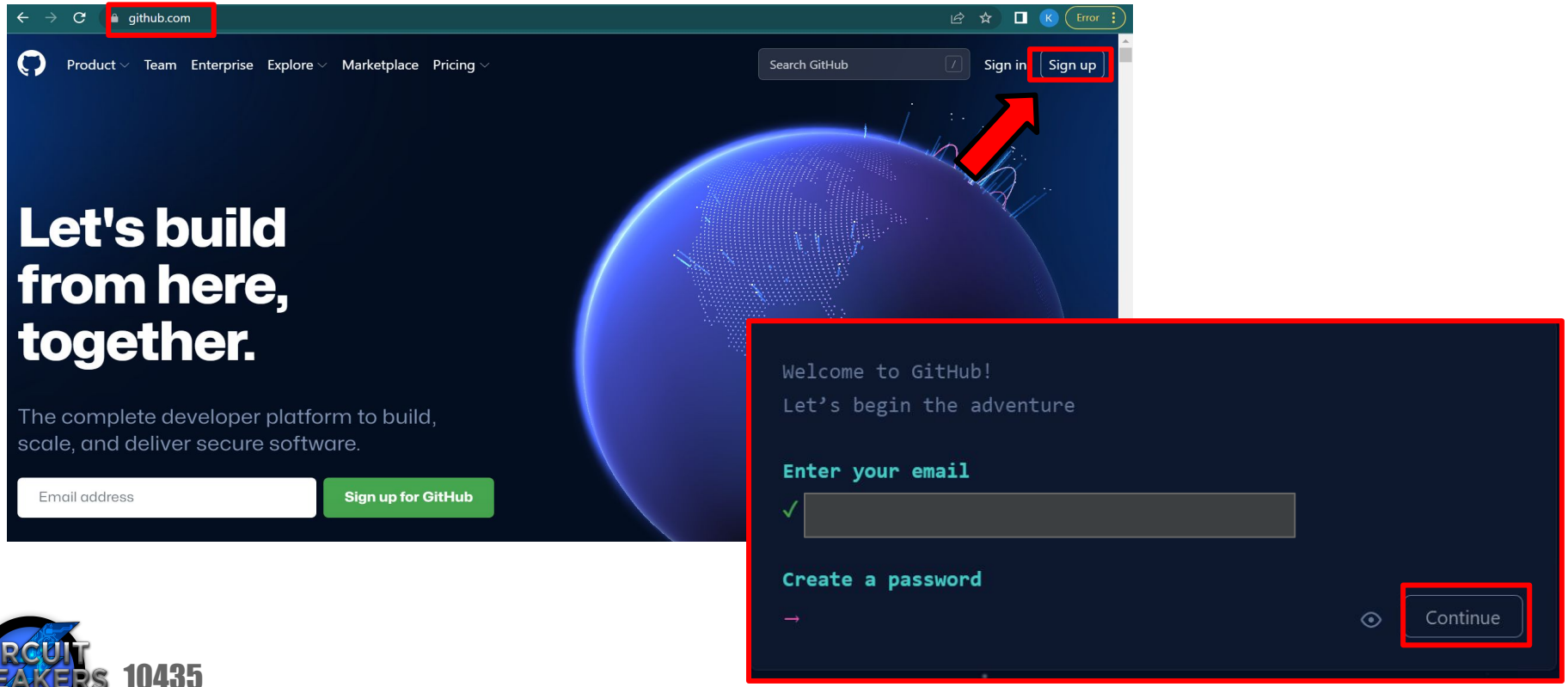
Walkthrough of Android Studio



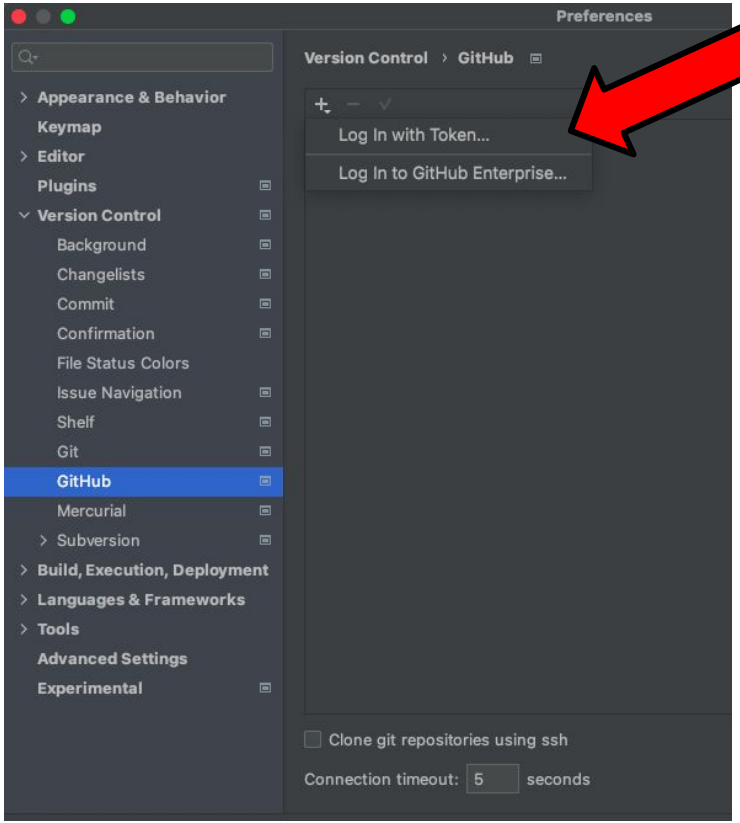
CIRCUIT BREAKERS

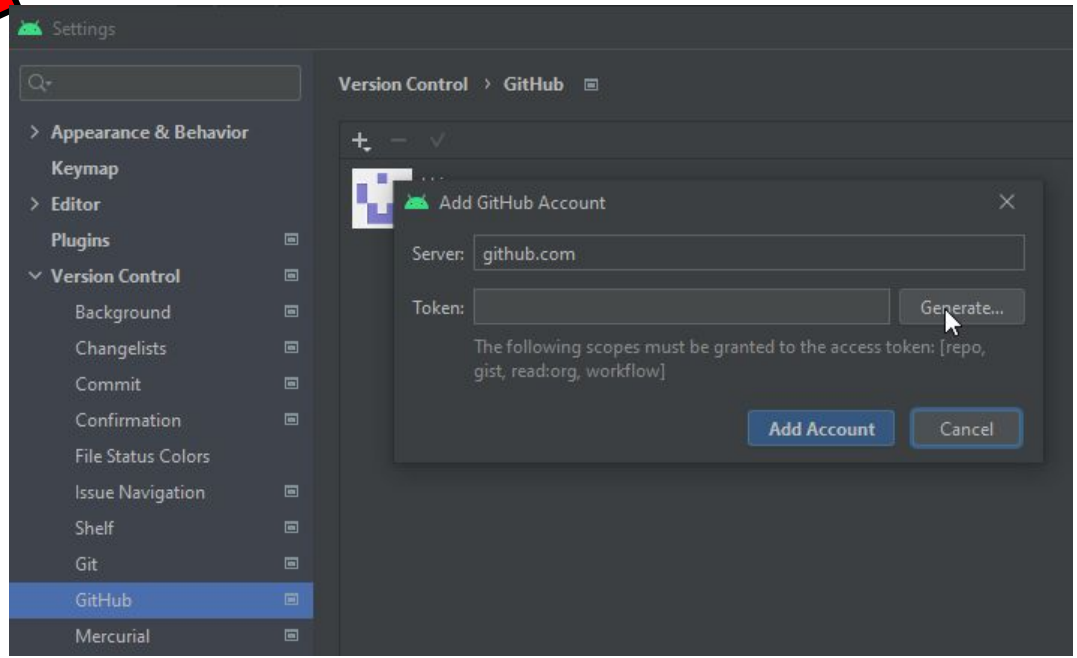
10435

Using GitHub



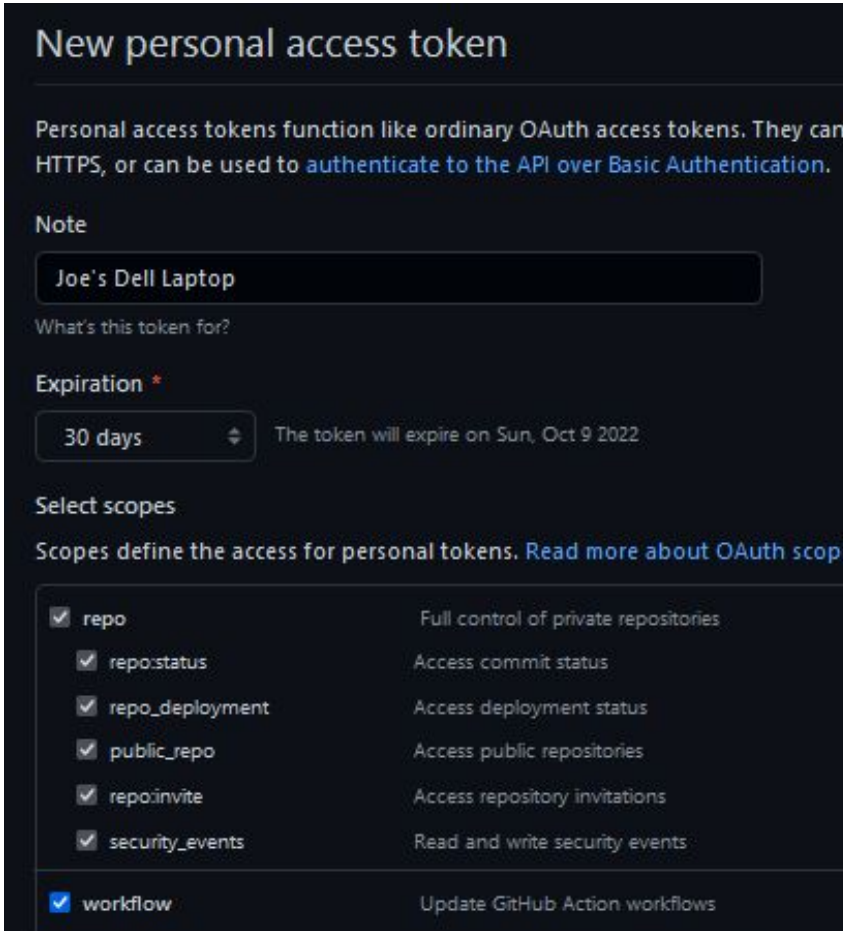
Enabling Github in Android Studio





Generate a token in your Github Account

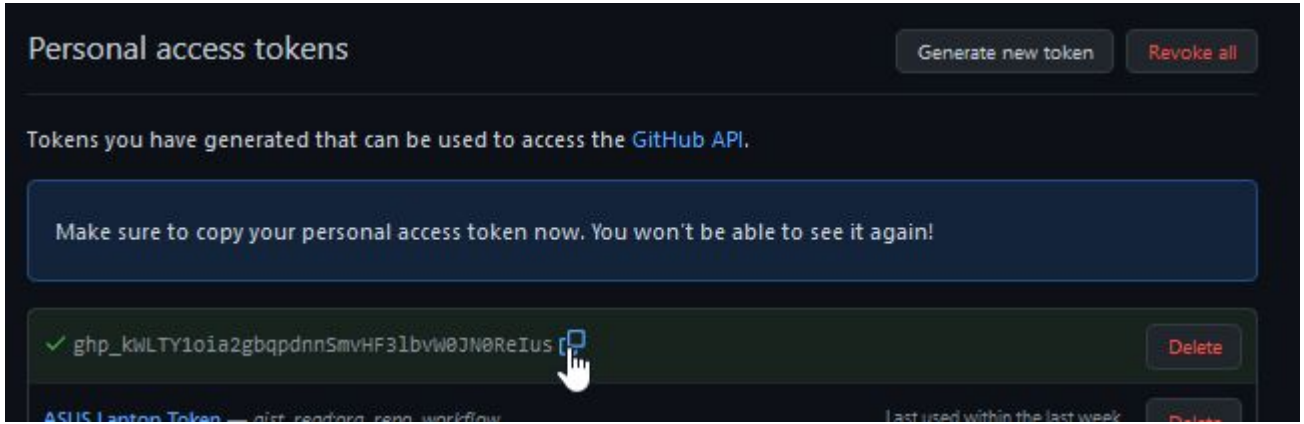
- Enter the name of your PC or laptop in the Note
- Make it “No expiration” (easiest but maybe not best)
- Make sure “repo”, “workflow”, “read:org”, and “gist” are selected and click “Generate”



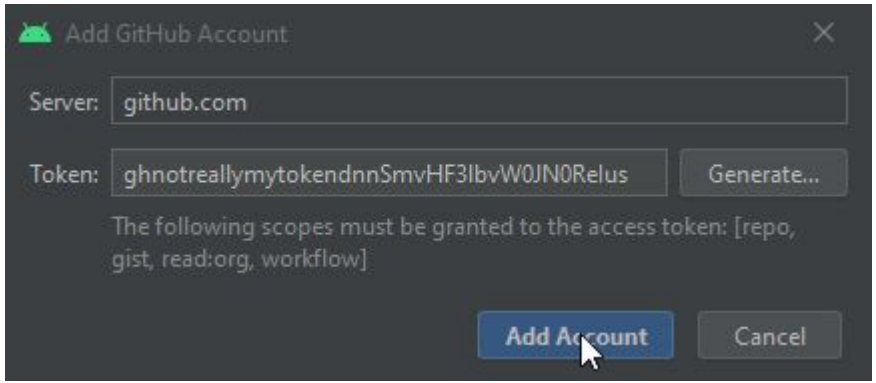
CIRCUIT BREAKERS 10435

Add your Github token to Android Studio

Copy to your clipboard



Back in Android Studio, paste it into the “Token:” field and click “Add Account”

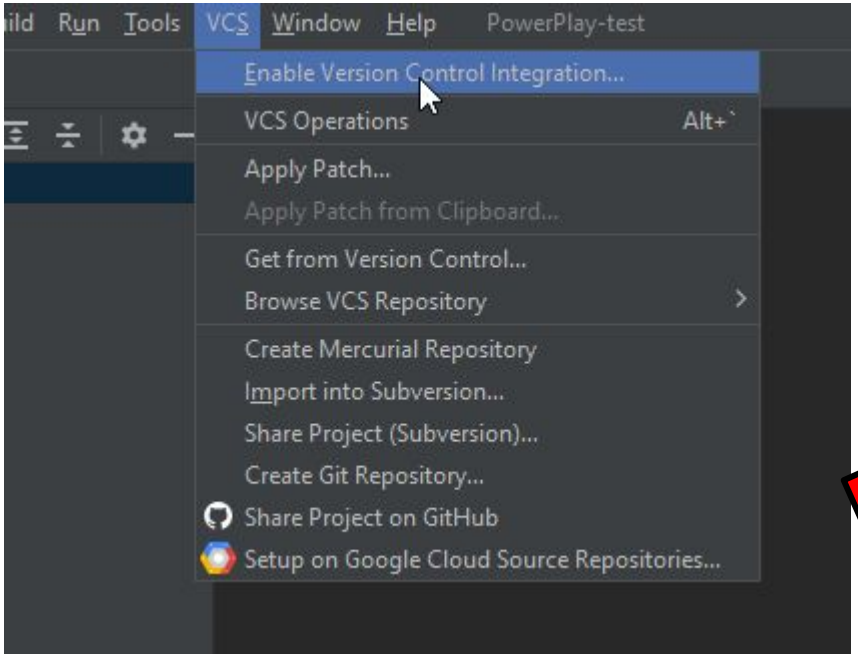


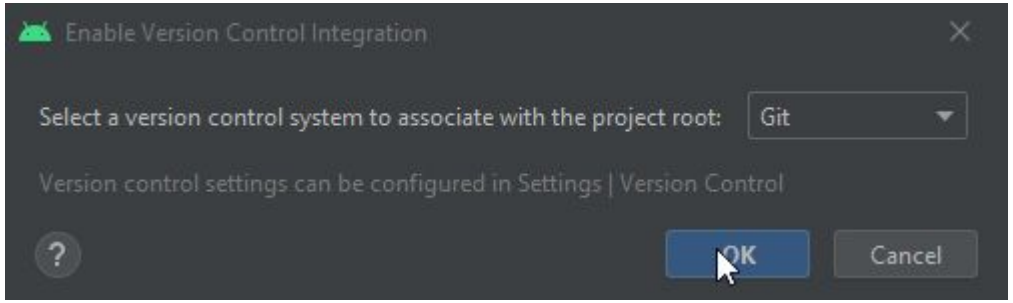
10435
CIRCUIT BREAKERS

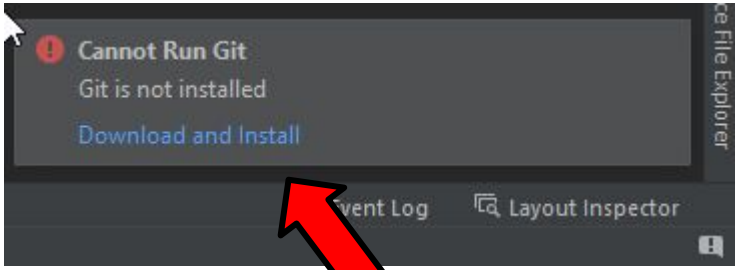
Installing Git

10435

If you have Git already installed, uninstall it using your operating system uninstall.

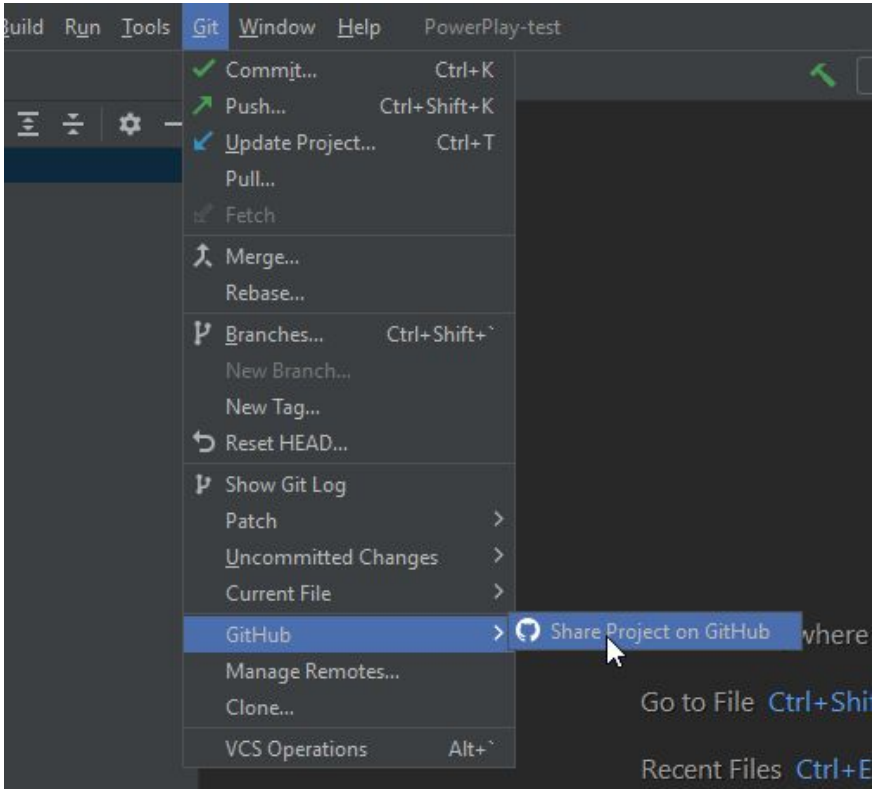


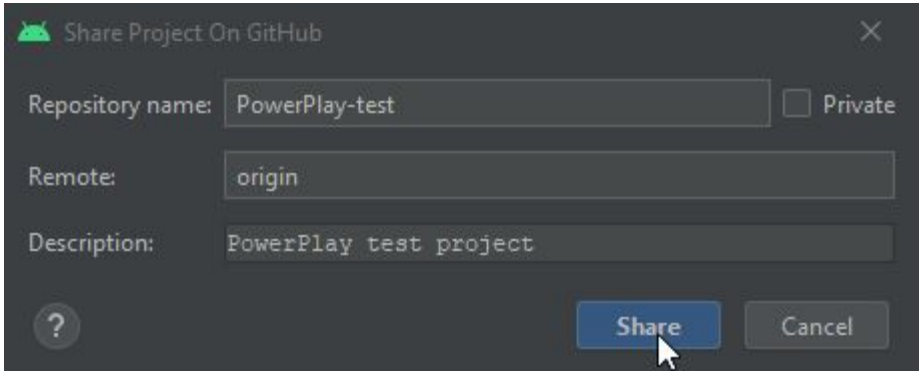




After clicking “Download and Install” do this again

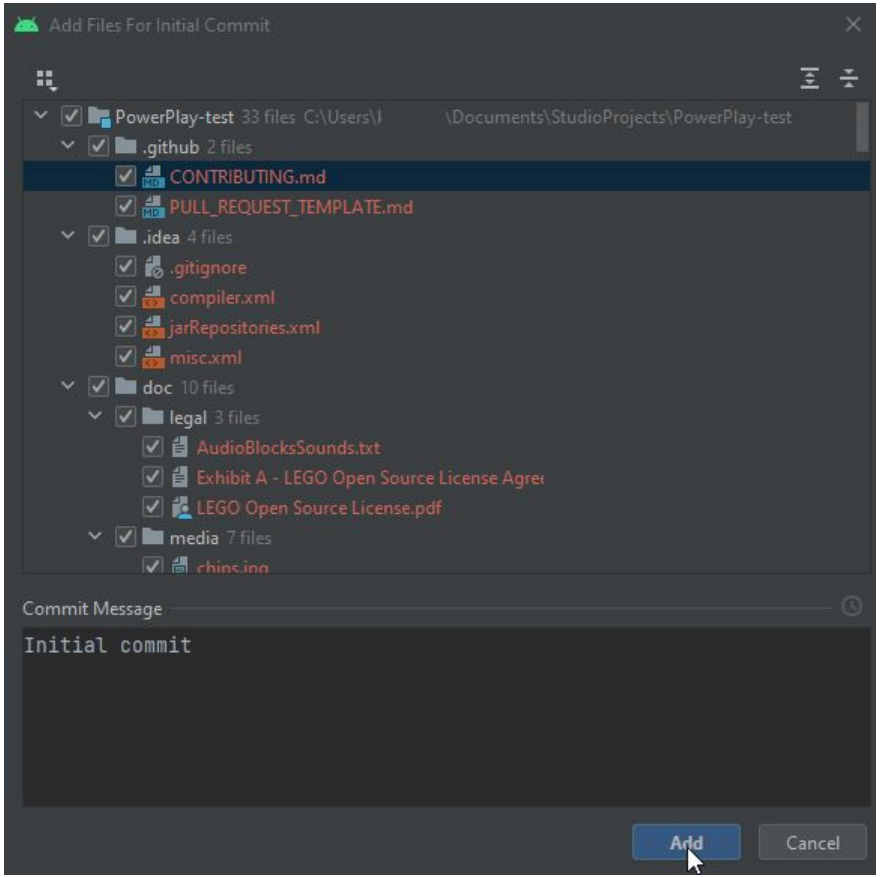
Creating Repository on GitHub

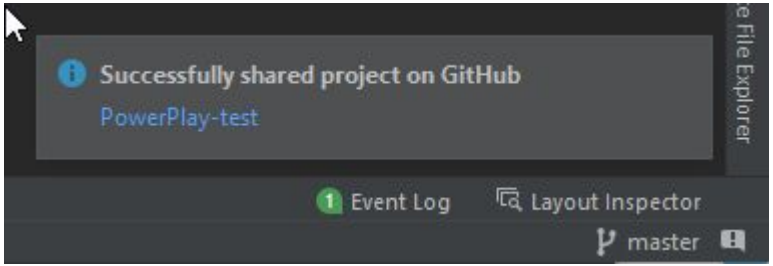




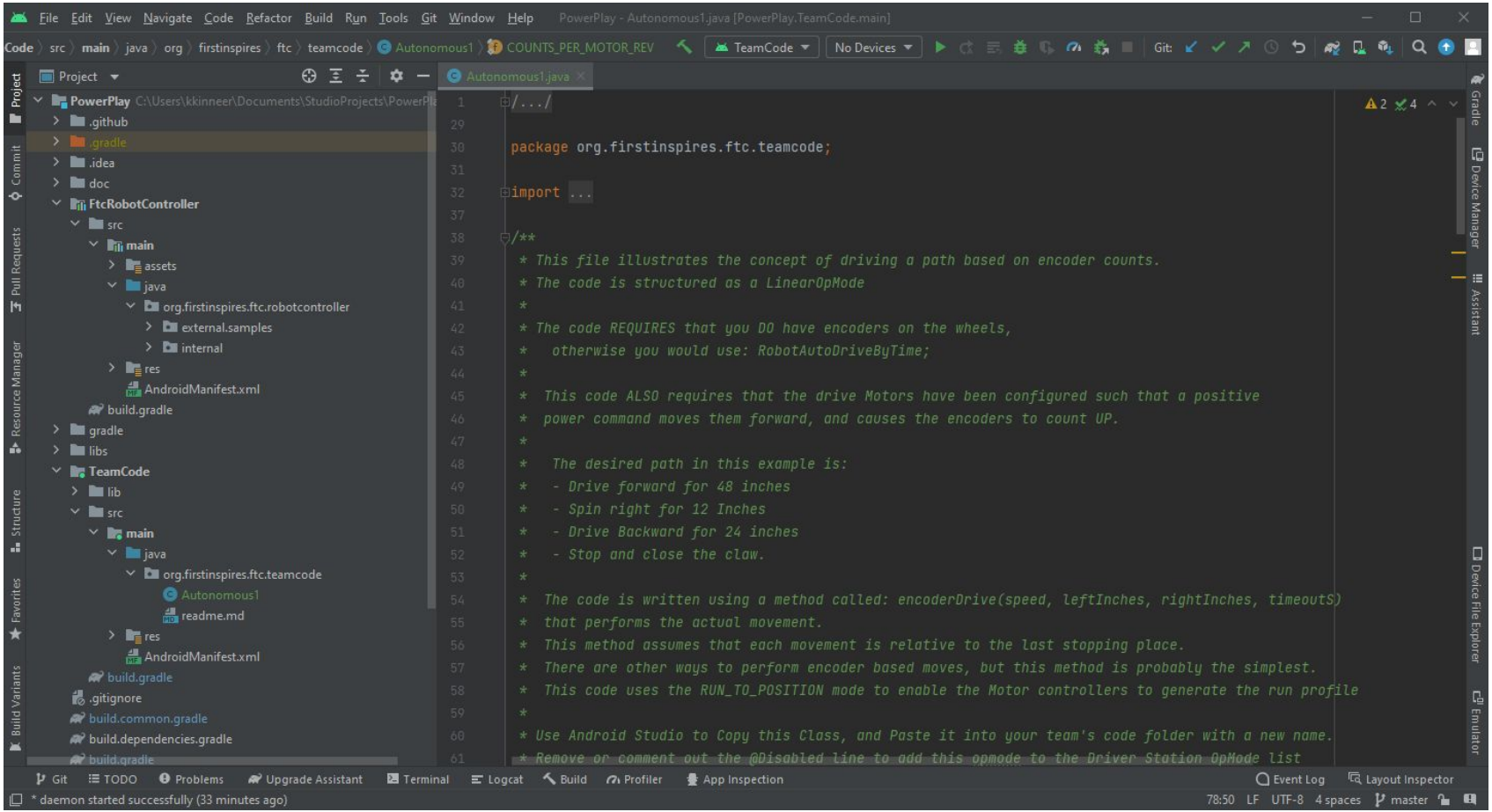
Creating Repository on GitHub

10435





TeleOp and Autonomous



10435

Programming Resources

From FTC:

- Android Studio Guide
- Javadoc.io RobotCore 7.1.0 API
- FTC Github - RobotController 7.2
- Java Video Tutorials

Learning Java

- Codecademy
- W3 Schools
- Learn Java in 14 Minutes

From Iowa FTC:

- FTC Programming with Java - Everything you need to know to get started (2020-08-22)
- Programming with the Circuit Breakers
- FTC Iowa 2020-2021
- FTC Iowa 2021-2022

Learning Github and Git

- How To Set Up A Github Repository For FTC
- How to Set Up Android Studio for FTC
- How to Use Github in Android Studio
- How to Use GitHub - Wizards.exe

Advanced Concepts

CIRCUIT BREAKERS
10435

1. Understanding Electrical/Mechanical Timing Delay
2. PID
    - Also - PD w/ Feedforward
3. Odometry / Location Algorithms
    - Pathing / Splines / Interpolation of Movement
    - Road Runner
    - Pathfinder II
    - FTC Dashboard
4. Vision/Recognition
    - Color
    - Shape/Object Recognition
    - Distance/Angle

Understanding Electrical/Mechanical Timing Delay

Electrical and Mechanical Timing Delay is a problem that affects every moving part in an FTC robot. These delays must be dealt with when coding your FTC robot, in order for your robot to perform or function precisely.

Two types of mechanical delay are caused by

- Momentum (overshooting your target after turning off a motor for instance)
- The normal time to reach position (Set servo to 90 degrees which is instant in code, but servo takes maybe 2 tenths of a second to get there).

Oversimplified - electrical delays are caused by

- Code cycle time. Code doesn't run instantly 1) Your Java code, 2) The SDK code, 3) Android code, and 4) other libraries.
- Embedded code that controls and reads devices

10435 CIRCUIT BREAKERS

PID

CIRCUIT BREAKERS
10435

PID, or Proportional Integral Derivative, is a control loop system where changes are continuously made based on the feedback it receives.

P - Present error- Start at 100% and gradually decrease the speed until there is no error in the system and you have reached the desired output.

I - Past error- Will increase/decrease the speed until it reaches zero error.

D - Prediction of future error- Measures how fast the error is growing or shrinking and increasing or decreasing the speed to compensate.

Why you need it: PID is helpful in correcting errors and controlling the speed of your robot, especially in sudden changes like acceleration at a fast pace.

Odometry and Location Algorithms

10435

Odometry refers to the use of rotational readings from wheels to figure out the robot's change in position and heading. This is usually done by reading the rotational values of two or three omni wheels (usually deadwheels) and performing calculations using a code library on the readings to determine the change in location.

Why do you need it:

If the code can know where the robot is on the field and control precisely where it will go, you can write amazing Autonomous routines and even some driver-assist functions in Tele-Op

See links at end for pathing (how to tell your robot where to go)

Vision/Recognition

Vision on an FTC robot can have many applications. Every one of those applications starts with a camera.

Vision plays an important role in nearly every FTC Season.

- Freight Frenzy - Detecting location of team shipping element
- Ultimate Goal - Determining how many rings are in the stack
- Skystone - Locating “Skystones” (a differently marked block)

Teams have also used vision to determine and control the robot’s location and angle relative to a target. Check out team 14481 Don’t Blink following a ring with Vuforia: https://www.youtube.com/watch?v=_Hxn4fzfN7k

Advanced Programming Resources

10435

Understanding PID

- Simple PID Demo as used for rotational angle
- PID Balance+Ball | full explanation & tuning
- Understanding PID Control, Part 1: What Is PID Control?
- PIDF Loops & Arm Control - KookyBotz

PID Programming

- PID Control in FTC Tutorial by Thermal Equilibrium
- PID Velocity Tutorial for FTC
- Using PID with Gyro in FTC Programming - LAFTC Team Wolfcorp 12525

Odometry

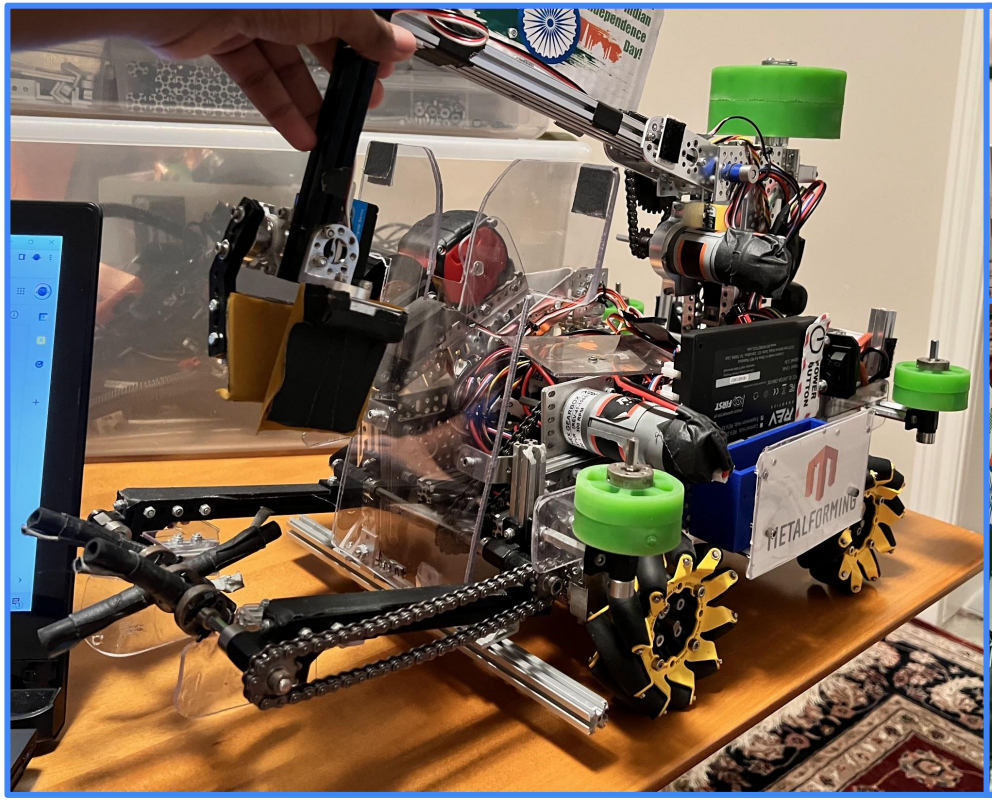
- The Math Behind Odometry - Rex Liu
- Odometry 101 for FIRST Tech Challenge Robots DrBatanga - Aperture Science
- Odometry in FTC by Uplift Robotics
- Pathfinder II
- Learn Road Runner
- Pure Pursuit tutorial by Gluten Free

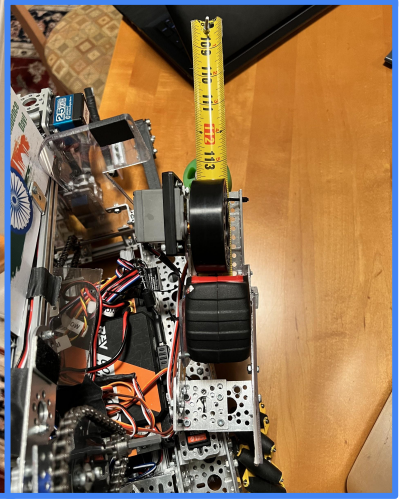
Vision

- Using Tensorflow - FTC github
- Vision overview from Game Manual Zero
- EasyOpenCV
- Basic Opmode with Vuforia

How these Concepts Relate to Build

CIRCUIT BREAKERS
10435





Concept Application:

PID- Wheels and Motors

Odometry- General Chassis

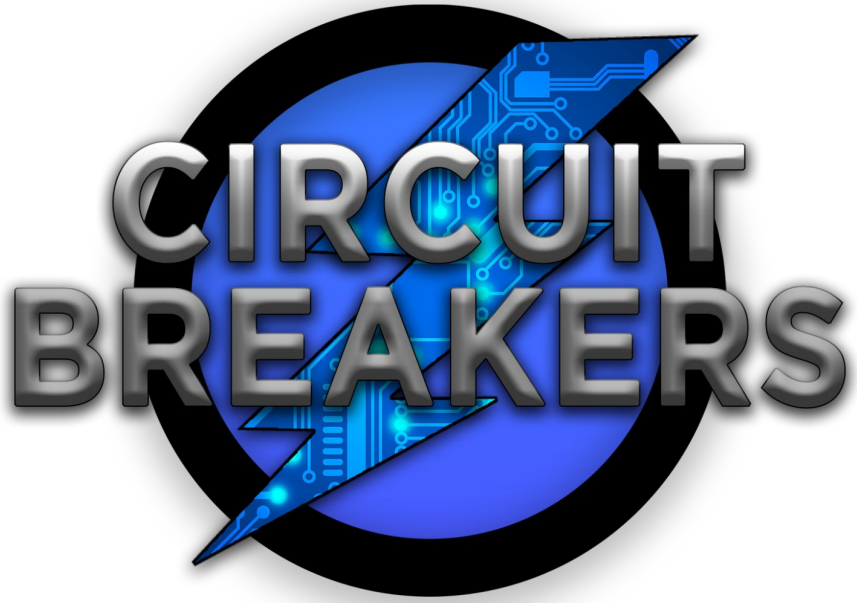
Vision/Recognition- Camera

Electrical/Mechanical Timing Delay- Full Robot

How to Integrate Team Roles into Planning

- Meetings must be planned with certain members of your team in order to work efficiently and successfully.
- Programmers will need to run a lot of testing, and trial and error with the robot.
- Builder(s) should be present while programmers are testing their code, this allows to fix the robot on the spot.
- Start making prototypes early on and split your robot into smaller, more specific parts
    - Individually tested alone by the programmer

CIRCUIT BREAKERS 10435



Thank you for listening!

10435 Circuit Breakers