

Advanced Programming Techniques & Autonomous Strategy



By Height Differential



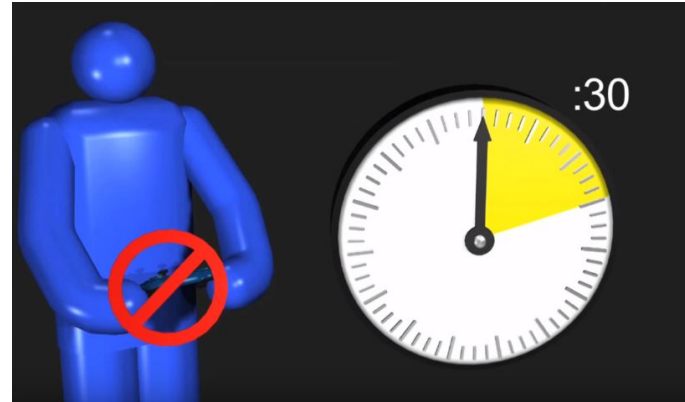
2016-2017 Kickoff

Brought to You By:



What is Autonomous?

- First 30 seconds of a match
- Pre-Programmed
- Robot moves without human interaction
- Many teams have little knowledge
- Underrated by many teams



2016-2017 Kickoff

Brought to You By:



Things to think about...

- Offense / Defense
- Scoring Points
- Positioning yourself for TeleOp
- Having pre-programmed options for many situations/strategies
- Complement your alliance partner
- Unique autonomous path
- Anticipate defense



FIRST[®] Tech Challenge

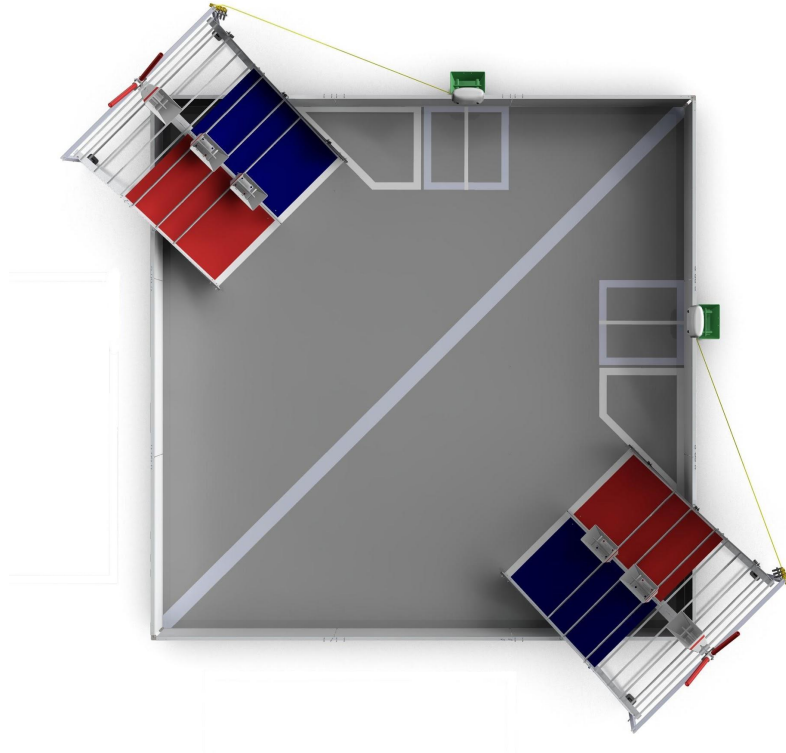
2016-2017 Kickoff

Brought to You By:



Use the Field!

- Walls
- Structures
- Beacons
- Colors
- Lines
- Game objects
- Use in teleop too



2016-2017 Kickoff

Brought to You By:



Sensors

- Modern Robotics
- Gyro
- Touch
- Color
- Optical Distance
- IMU (Inertial Measurement Unit)
- Ultrasonic (Range Sensor)
- Encoders



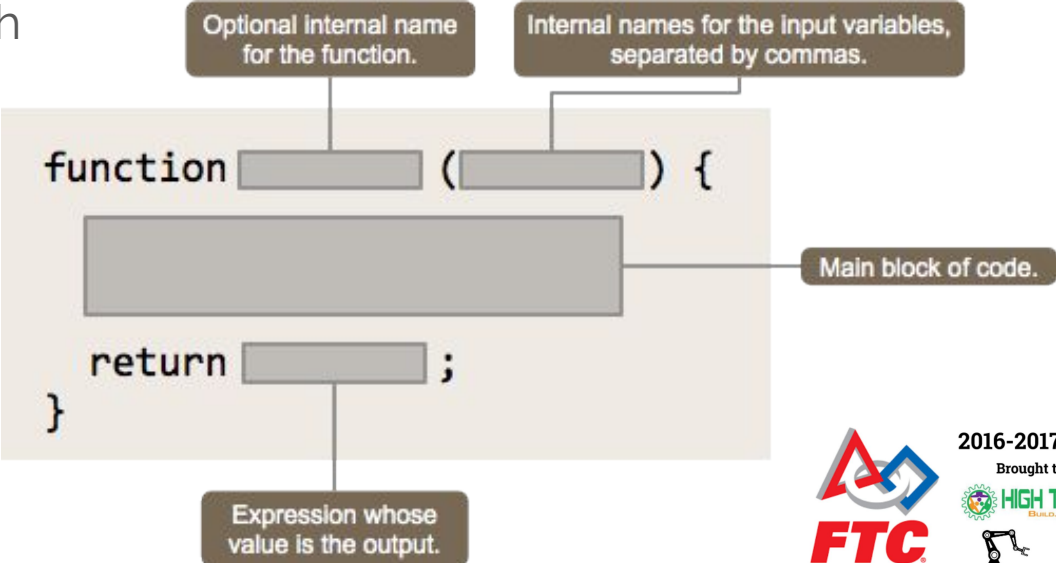
Using the Gyro

- Using the gyro allows you to maintain a straight course relative to your starting position. This is useful for if you run over blocks and get off course, the gyro may be able to correct it.



Functions

- Custom functions make your life easier!
- Simpler to repeat tasks
- Less chance of mistyping or mistakes
- Decreases program length
- Libraries
- Easier to understand



Functions Example

```

/**
 * Simple drive train power
 */
public void td4motor(double leftValue, double rightValue, double Sensitivity) {
    leftFront.setPower(leftValue / Sensitivity);
    rightFront.setPower(rightValue / Sensitivity);
    leftBack.setPower(leftValue / Sensitivity);
    rightBack.setPower(rightValue / Sensitivity);
}

/**
 * Stop Motors
 */
public void StopMotors() {
    frontRight.setPower(0);
    frontLeft.setPower(0);
    backLeft.setPower(0);
    backRight.setPower(0);
}

```



State Machines

- The state of a robot is what is happening at a given moment in time
- Allows you to do multiple operations at a time
- Pick between the “states” of the robot
- Faster autonomous
- Combined/coordinated movements



FIRST® Tech Challenge

2016-2017 Kickoff

Brought to You By:



State Machines Real Life Examples

Bake a pizza:

1. Take the frozen pizza out of the freezer.
2. Preheat the oven at 400 degrees
3. Wait for the oven to reach the preheat temperature.
4. Put the frozen pizza into the oven.
5. Set a timer for 15 minutes
6. Wait for the timer to go off.
7. Check if the pizza is thoroughly cooked. If not, put it back in the oven and add 5 more minutes to the timer.
8. Repeat step 5 until the pizza is done.
9. Take the pizza out.
10. Cut it and eat it.



Doing laundry:

1. Put clothes into the washing machine.
2. Put in laundry detergent and start the washing machine.
3. Wait for the washing machine to finish.
4. Transfer the clothes into the dryer, put in the fabric softener and start it.
5. Wait for the dryer to finish.
6. Take clothes out of the dry and fold them.
7. Put them away back into the closet or dresser.

State Machines/Switch Loop Programming Example



```
switch (mCurrState) {
  case STATE_Delay://Our current robot state: wait for
the gyro to calibrate
    if (GeneralTime.time() > delay) {
      mCurrState =
BlueClimbers.STATE_DriveForward1//Move to next robot state
    } else
      telemetry.addData("l: ", "Delay
Left(GyroDone) " + String.valueOf(delay -
GeneralTime.time()); //Display how much time is left of our delay
      break;
    case STATE_DriveForward1://Our current robot state:
drive forward from start
      telemetry.addData("2", mGyro.getHeading());
      if (AvgEncoder() < 20 * EncoderToInch) {
        VLF(1, 0);
      } else {
        StopMotors();
        mCurrState =
BlueClimbers.STATE_GyroTurn1//Move to next robot state
      }
      break;
    case STATE_GyroTurn1://Our current robot state: turn
parallel with the mountain
      if (goalReached == false)
        GyroTurn(45);
      else {
        mCurrState =
BlueClimbers.STATE_DriveForward2//Move to next robot state
        goalReached = false;
      }
      break;
```

```
case STATE_DriveForward2//Our current robot state: drive forward close
to the white line
      telemetry.addData("2", mGyro.getHeading());
      if (AvgEncoder() < 68 * EncoderToInch) {
        VLF(1, 45);
      } else {
        mCurrState =
BlueClimbers.STATE_LookForWhite//Move to next robot state
      }
      break;
    case STATE_LookForWhite://Our current robot state: drive
forward until we find the white line
      if (optical.getLightDetectedRaw() < 10) {
        VLF(1, 45);
      } else {
        if (ballconfig == BallConfigs.BallClear) {
          mCurrState =
BlueClimbers.STATE_ClearBallArc//Move to next robot state
        }
        else{
          GeneralTime.reset();
          mCurrState =
BlueClimbers.STATE_DriveBack1//Move to next robot state
        }
        StopMotors();
      }
      break;
```



FIRST® Tech Challenge

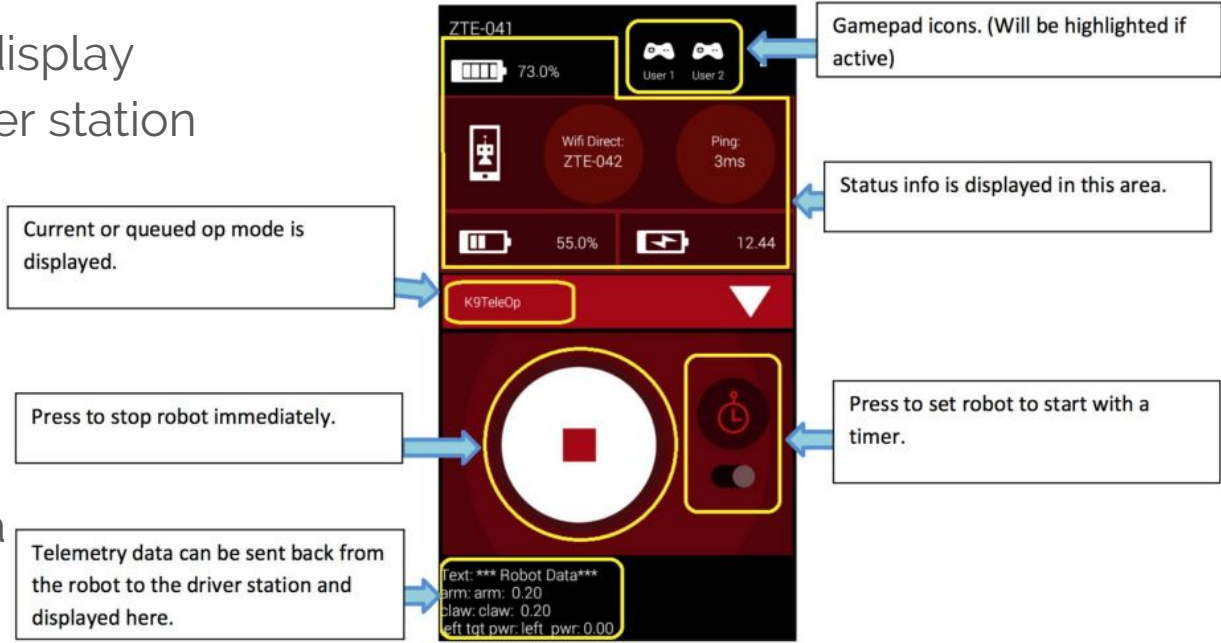
2016-2017 Kickoff

Brought to You By:



Telemetry

- Telemetry is a way to display information on the driver station
- Gives you feedback
 - Sensors
 - Motors
 - Servos
 - Battery
- Easier for debugging
- Good for in-match data
- Menu



Telemetry Example

```
/**
 * Display simple telemetry data for teleop
 */
public void TelemetrySub(){
  dashboard.displayPrintf(0, "Teleop");
  dashboard.displayPrintf(2, "TeleOp Started At : %s", startDate);
  dashboard.displayPrintf(3, "Current Run Time : %.2f", runtime.time());
  dashboard.displayPrintf(4, "Current WinchLock Status : %b", WinchLockUp);
  dashboard.displayPrintf(5, "Current WinchEnc Status : %d", WinchMotor.getCurrentPosition());
  dashboard.displayPrintf(6, "Current Roll: %d", Math.round(Math.toDegrees(roll)));
  dashboard.displayPrintf(7, "Current Tilt Mode: %s, Current Winch Pos: %f", TiltMode, CurrPosition);
  if (HoldActive == 1) {
    dashboard.displayPrintf(1, "Current Drive Speed: HOLDING FORCE");
  }
  else if (Speed == 2) {
    dashboard.displayPrintf(1, "Current Drive Speed: 50%");
  }
  else {
    dashboard.displayPrintf(1, "Current Drive Speed: 100%");
  }
}
```



Commenting

- Does not affect the code!
- Annoying but necessary
- `//` everything on the line will be a comment
- `/*` everything in between the asterisks will be comments, even if runs onto several lines like this `*/`
- Easier for someone to follow
- Faster to edit and troubleshoot programs
- Cleaner look
- Meaning of a variable
- Notes



Commenting Example

```
/**
 * Created by Height Differential on 2/2/2016.
 */
```

```
public class HDUtil {

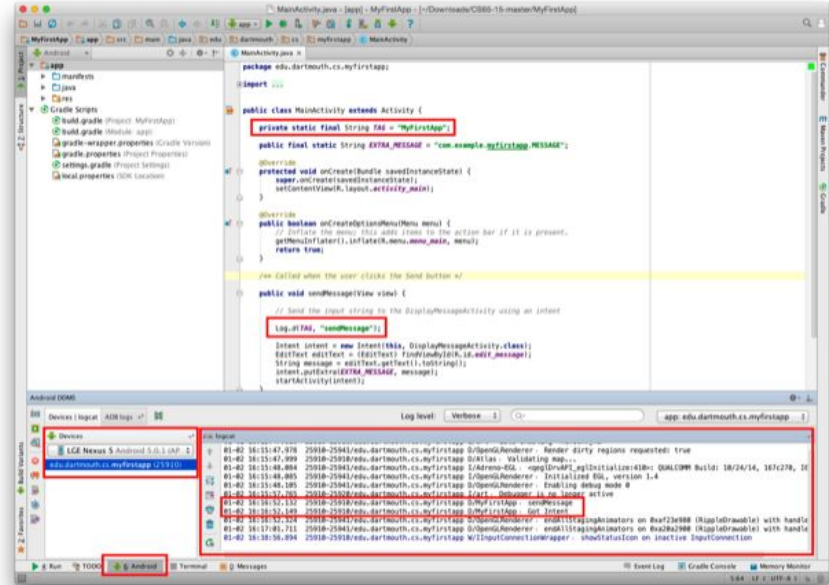
    public static void HDSleep(long sleepT){ //Sleep function which allows us to wait
        long TimeUntilDone = System.currentTimeMillis() + sleepT; //Add sleeptime to Systemtime to
        find when to stop waiting
        while(sleepT > 0){ //While sleep time is larger than 0, (Still needs to sleep)
            try {
                Thread.sleep(sleepT); //Try to sleep for sleep time.
            }
            catch (InterruptedException e){
            }
            sleepT = TimeUntilDone - System.currentTimeMillis(); //Calculates sleep time,
            SleepTime-Current time
        }
    }

    public static double round(double value, int Digits){ //Function that rounds decimal for easier
    to read telemetry.
        BigDecimal BD = new BigDecimal(value); //Create big decimal class
        BD = BD.setScale(Digits, BigDecimal.ROUND_HALF_UP); //Set class to round half up
        return BD.doubleValue(); //ReturnRoundedValue
    }
}
```



Logging

- Easily display information
- Types of logs:
 - `Log.e(String, String)` (error)
 - `Log.w(String, String)` (warning)
 - `Log.i(String, String)` (information)
 - `Log.d(String, String)` (debug)
 - `Log.v(String, String)` (verbose)



For more information - <https://developer.android.com/studio/debug/am-logcat.html>

Heartbeat

- A heartbeat allows you to confirm that everything is working
- Visual signal
- Mechanical or Electrical
- Connected and program is running
- Quickly start problem solving
- Saves match time
- Physical bar on servo
- LED



FIRST[®] Tech Challenge

2016-2017 Kickoff

Brought to You By:



Arrays

- A list of values
- Index of 0

Defining an array:

```
int[] myArray = {1,5,9,8,5,6,7,5};
```

Get the first item in the array:

```
int myNumber = myArray[0];
```

Get the fifth item in the array:

```
int myNumber = myArray[4];
```

Get the length of the array:

```
int myLength = myArray.length;
```



FIRST[®] Tech Challenge

2016-2017 Kickoff

Brought to You By:



Enumerators

- Specify a set of values
- Use the values like any other variable

Defining an enumerator:

```
enum DOW = {mon, tues, wed, thurs, fri};
```

Create a enum variable:

```
DOW currentDay = DOW.mon
```

Use it:

```
if(currentDay = DOW.tues){
//Do Stuff!
}
```



FIRST[®] Tech Challenge

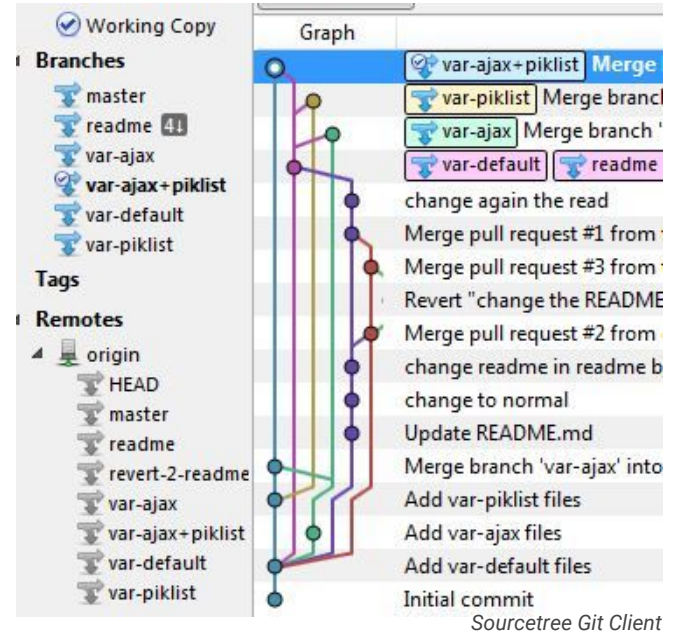
2016-2017 Kickoff

Brought to You By:



Source Code Control

- GIT
 - Allows you to revert to previous versions of code
 - Easily branch off and try something new
 - Merge or delete it depending on if it works
 - Main Git repository hosting services are Github and Bitbucket
 - Many Git clients available



Resources

- Mentors
- Your peers
- FTC Reddit (www.reddit.com/r/FTC)
- FTC Forums (ftcforum.usfirst.org)
- Stack Overflow (stackoverflow.com)
- Height Differential
(heightdifferential@gmail.com)
- Intelitek Training Material
(<http://www.intelitek.com/roborio/>)



FIRST® Tech Challenge

2016-2017 Kickoff

Brought to You By:



Thanks for listening!
Any questions?



2016-2017 Kickoff

Brought to You By:

