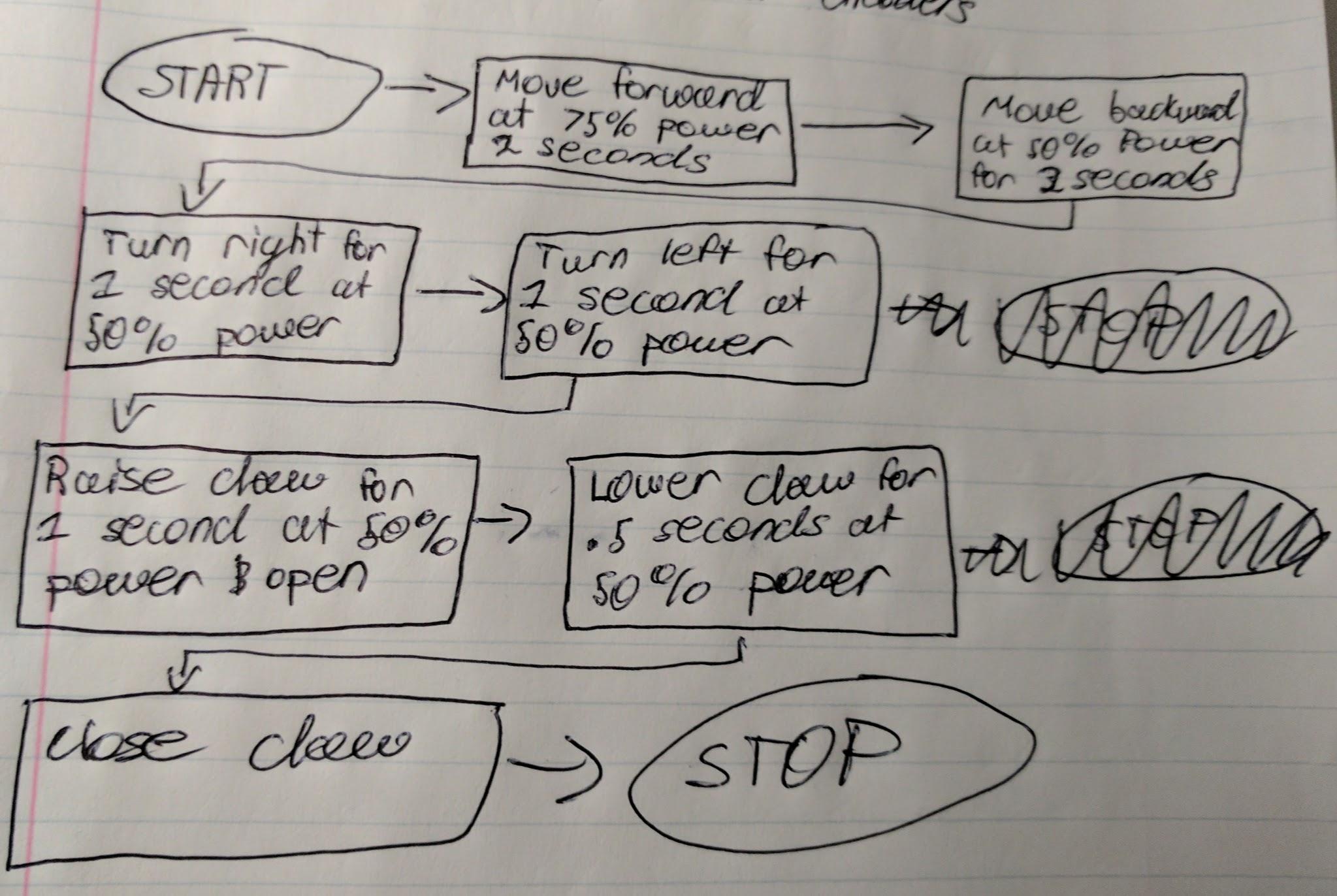
Summer Project Journal - Luna Sensors

Raashi Dewan

* 6/30/17
  + Collected parts
  + List of tasks
    - Touch sensors
      * Limit switch for arm to move up and down
        + In the front and back of the robot to prevent it from being driven too far when the arm touches the base
    - Gryo
      * Driving in and square and reversing (autonomously)
  + Next steps
    - Mount sensors and CID module - **7/1/27**
    - Set up correctly jdk - **7/2/17**
    - Configure robot - **7/3/17**
    - Gyro sensor - **7/5/17**
      * How it works
      * Potential uses
      * Advantages and limitations
    - IR seeker - **7/1/17**
      * How it works
      * Potential uses
      * Advantages and limitations
    - Infrared sensor - **7/6/17**
      * How it works
      * Potential uses
      * Advantages and limitations
    - Light sensor - **7/7/17**
      * How it works
      * Potential uses
      * Advantages and limitations
    - Color sensor - **7/7/17**
      * How it works
      * Potential uses
      * Advantages and limitations
    - Ultrasonic sensor - **TBD**
    - Read data from sensors as user input for autonomous - **TBD**
    - Program touch sensors - **TBD**
    - Program gyro sensor - **TBD**
* 7/1/17
  + Started to mount sensors, but the cables are not long enough to reach the cid
    - Do we have available connectors?
  + Gyro pseudocode
    - Drive in a straight line for 2 ft - (needs encoders, could I use another sensor for this?)
      * When heading shifts from (0 to 180) correct by adding power to right motor
      * When heading shifts from 0 to (180 to 356] correct by adding power to left motor
    - Turn right 90 degrees
      * Power to left motors until heading is 90
    - Drive straight
    - Turn right until heading is 180
    - Drive straight
    - Turn right until heading is 270
    - Drive straight
    - Turn right until heading is 0
* 7/2/17
  + Disregarding previous schedule outline from 6/30/17, I created a new schedule for the project, with some more guidelines and planning, because I realized that I was jumping into the project without really thinking it through.
    - **Interacting with hardware (motor control)** - **July 8th**
      * Forward, backward, and turning
      * Move at different powers
      * Drive by time
    - **Gyro - July 15th**
      * Receiving and displaying data from the gyro
      * Turning to specific angles
      * Driving and turning in a square autonomously
    - **Limit switches - July 22nd**
      * Displaying data from the sensor
      * For the arm when it reaches the base of the robot in the front and back to prevent the motor from continuing to drive
        + Drive the motor until sensor is triggered, then set motor power to 0
    - **ODS sensor - July 29th**
      * Displaying data from the sensor
      * Drive until a line is seen
    - **String the programs together - July 31st**
      * Drive to a line and turn 90 degrees
      * Raise and lower arm until reaching the sensor
      * Stretch - Drive forwards and turn until reaching another line
  + Stretch goals
    - Range sensor
      * Display data from the sensor
      * Drive to a specific distance from the wall and stop
        + Then drive back from the wall and stop
  + What I did today
    - Interacting with hardware flowchart



* + - Using the available pushbot drive by time class in the external samples of the FTC robot controller in the FTC sdk I was able to create a basic program to move the robot. I took the existing class to use as a base and wrote my code, based on my flowchart above, to replace the steps there.
  + Next Steps
    - Testing the code on the robot
* 7/5/17 - 7/7/17
  + Whenever I initialized my hardware interaction test I recieved the error, IllegalArgumentException - Unable to find a hardware device with the name “left\_drive”. I don’t have a device named “left\_drive” in my code, so I checked the HardwarePhushbot class which I was using for my configuration, and I realized that I had overridden some of the code with my own.
  + I decided instead of using the PushbotHarware class,I would create one specifically for Luna. This way I was able to get a more thorough understanding of the software - hardware interaction, and I would stop writing conflicting code.
  + I also decided to write a quick telop program that we can use for outreach events. I noticed at our last event that many of the kids found it difficult to drive the pushbot with the current program, especially driving forwards and backwards using both joysticks, so I wrote a program where if you push either joystick forward, backwards, left, or right the robot will move in that direction. Additionally, I decreased the power applied to the wheels to make it slower and easier to control. One more thing that I noticed was that kids wanted to use the triggers, so I kept the bumpers for opening and closing the claw and used the triggers to raise and lower the arm.
  + I was still having trouble with connecting to the robot hardware, so I decided to check my code line by line against the code that I wrote last year for Luna and the PushbotAutoDriveByTime\_Linear. When checking my code against the old code for Luna, I thought that I was missing the init() method for the hardware map, but I realized that I simply had it inside the runOpMode() method instead of before. Unfortunately I did not realize this until after I tried to edit the code to extend the OpMode class instead of LinearOpMode in order to place the init() method before.
  + After going back to my code and checking it again, it occurred to me that I could have an error in my hardware class that I made for Luna, so I checked it against the PushbotHardware class, and I found no discrepancies in the two programs, other than the addition of the gyro sensor to LunaHardware.
* 7/10/17 - 7/12/17
  + I wanted to check if the software was not recognizing any hardware or just the left\_drive, so I wrote a program to connect to the gyro sensor and read the angle value and display it on the phone. There are multiple bugs with the program. If I initialize it, and wait for the phone to display that the gyro is finished calibrating it automatically stops the program. If I press play before the gyro is finished calibrating, the program runs, but the gyro value is constantly 0, instead of displaying the integrated z value.
    - I realized that I deleted my waitForStart() statement, so I added that back in and my test worked well outputting the angle of the robot to the telemetry.
  + I decided to run the DriveByTime class with HardwarePushbot instead of LunaHardware for the robot hardware to see if this helps. Unfortunately, I received the same error message as before.
    - Jyoti Aunty also suggested running some of the old code for Luna that I know worked to see if it is an issue with my code or a robot configuration issue.
    - Since I was unable to test my teleop program with Luna, as I forgot to pick up a controller, I tested it today to see if the bug is in my autonomous program, or in the hardware connection itself.
    - Both my teleop program and the old code worked just fine on the robot, so I know that there is an issue with my autonomous code.
    - I decided to talk to Sarthak to see if he ever encountered this issue before. He hadn’t but suggested defining the hardware from the hardware map in my autonomous code instead of in a separate class to see if that worked.
      * My code is no longer throws the error of being unable to find “left\_drive”, and it executes properly. However, none of my other code using the same hardware class had any issues. Because this change was necessary to make my autonomous code work, but not my teleop, my guess is that there is something in the autonomous opMode register that the hardware object disagrees with.
* 7/17/17 - 7/19/17
  + Additional task for future, record sensor data to a file
  + Gyro Sensor Program
    - Resource- <http://www.modernroboticsinc.com/Content/Images/uploaded/Sensors/Modern_Robotics_Gryo_Sensor-Steering_Tutorial.pdf>
    - I finally ran my gyro program, but my code was throwing a NullPointerException, so I defined the hardware variables internally. This was the same issue I ran into before with my DriveByTime program, but I wanted to try one more time to use a hardware object for my autonomous program.
    - Then it moved forward and stopped throwing an IllegalFormatConversionException.
      * Sarthak suggested changing the following

while (!isStopRequested() && gyro.isCalibrating()) {

sleep(50);

idle();

}

* To the following

while(gyro.isCalibrating()){

//do nothing

}

* 7/20/17
  + Touch sensor test program
    - I was sucessfully able to read the value of the touch sensor. Next I will write a program to use the touch sensors as limit switches for the arm to prevent over driving it. This would be especially useful for outreach events, where kids tend to keep driving the arm and putting stress on the motor and servo wires.
  + When I added the second touch, the code did not read the second value. However, my limit switch program works perfectly for the touch sensor that worked in my sensor test. I just have to get both working at the same time, and it should work.
    - To avoid changing the code until I rule out a hardware issue, I am going to switch the ports of the touch sensors to see if the same sensor malfunctions. However, when I switched the ports both sensors functioned normally.
* 7/26/17
  + My ods sensor test works, but I still don’t really understand what the values mean.
  + Next steps: look up what the numbers represent and how to use them.
  + Program pseudocode
    - *//check color for blue, if not blue, keep moving forward/right until blue*
      * *//return telemetry of "blue detected"*
    - *//check color for red, if not red, keep moving forward/right until red*
      * *//return telemetry of "red detected"*
  + Ods value for blue -
  + Ods value for red -
* At some point when Luna was being used to test blocky the cpd module stopped working. I believe the fuse may have blown, but I can't be sure. I have moved the sensor that I have not finished working with, or need for the programs that I am working on to the holonomic drivetrain.
* 8/10/17
  + To incorporate touch sensors, ods sensors, and data logging I wrote a program to log the ods values when a touch sensor is pushed.
* Re-organizing the code to make it more modular
  + I want to make a separate class with all my drivetrain methods. This is a good practice, so that I can make changes in one class that will effect all my programs without making changes in multiple locations. This way I can use a drivetrain object to move my robot, without rewriting my code in every class.
    - Things to consider
      * My hardware class didn't work in my autonomous programs for Luna, so I don’t know if the drivetrain class will be effected in the same way.
  + 8/13/17
    - I wrote a drivetrain class with methods for each state.
  + When I went to test my code I realized that the driver station was on robot wifi protocol version v110 and the robot controller was on v11. First I had no idea what this meant so I checked the forums and found out that I had to re-download the newest sdk from gitHub and transfer all my code to a new project.
  + Another issue, when I was configuring the drivetrain, one of the motor controllers was not showing up. I tried switching out the USB cable with no effect. After talking to Ishaan I checked the module with core device discovery I realized it was broken, and we have to switch it out.
  + The phones were connecting yesterday, but today when I tried to reconnecting the phones from the settings in the driver station the robot controller won't appear (in the driver station app). I checked the wifi direct connections in the phone settings and they are connected to each other.
    - I cleared the wifi direct groups and restarted and reconnected the phones
  + The teleop program using my drivetrain class keeps throwing a null pointer exception even when defining the hardware in various different places, so I wrote a basic teleop program to test first.
    - It turns out that I just forgot to define my object before the runOpMode() method, so I changed that and the code runs as it was supposed to.
  + I keep getting a nullPointerException in my gyro program, which I didn't get when I wrote gyro code on Luna, where I didn't have a drivetrain class and I used the hardware object in other programs, but not that one. I was planning to try writing a program without using the drivetrain or hardware classes and see if I get the same error.