Planning – How you and your teammate plan, work to solve the problem

**Arduino:**

Study:

* PID Library
* Motion controller
* Phillips Sensor Library
* Sorting algorithms
  + Study through all possible algorithms for sensor implementation

**Hardware:**

* Test Sensors:
  + Measure by 0.5cm difference
  + Capture readings
  + Draw graph
  + -> understand the sensor readings and plan sensor placements from readings
* Watch past semesters video:
  + understand other teams/groups sensor and hardware placements
  + understand infrastructure and how hardware affects overall movability (including weight distribution)
* Read Datasheet
  + Knowing hardware limitations
  + Understanding hardware usability and coding instructions

Implementation – Algorithms chosen and important code segment

**Motorshield:**

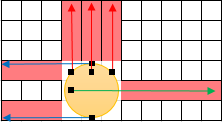
**PID:**

**Sensor:**

* Since algorithm is going the “follow/hugging left wall” concept

We..

* + Implement 3 front short sensors (left, mid, right)
    - Sense up to 3 grids (30cm 0,1,2,9) to prevent issues of phantom block
    - for calibration of robot (straightness against the wall & distance towards the wall)
  + Implement the right long sensor to sense up to 5 grids (50cm 0,1,2,3,4,9)
    - Sensor is able to sense half of the maze horizontally while following the wall
  + Implement 2 left sensor (front and back) [able to sense 3 grids 30cm 0,1,2,9] to for calibration purpose and to ensure robot to stay on-grid (by setting relevant displacement to the left side of the wall).



Problem Faced – Any difficulties encountered

* Sensor interference and positioning
  + Throughout the whole implementation process, realize that sensor can interfere each other when sensing. (eg. Front sensor sense forward, path towards front is obstruct with another sensor sensing the left side)
    - We alternate the placing by placing the front sensors on the top side of the mounting plate while the left sensors were placed on the bottom side
  + The positioning of the sensor greatly affects the reading
    - Due to the property of the sensor
* Not equal wheel spinning
  + Affects robot movability (forward movement too little, turning movement lacks accuracy
  + Robot not moving straight (adjust breaks and manually increasing tick count of faster wheel)
* Sensor reading inconsistent
  + Constant testing and read requires to ensure sensor is accurate in sensing the blocks
* Inconsistent movements
  + Due to power (battery) which affects the constant movement and accuracy of turning of the robot
* Consumption power issue
  + Powerbank & Rpi power usage affects the power given to the robot which affects actual movement
* Wiring and hardware building infrastructure
  + Wire stripped and etc.
* External library like PID and Philips sensor are not optimized
  + Philips sensor library uses Bubble Sort which is O(N2)
  + Need to study the flow of these external libraries and create a version for optimization