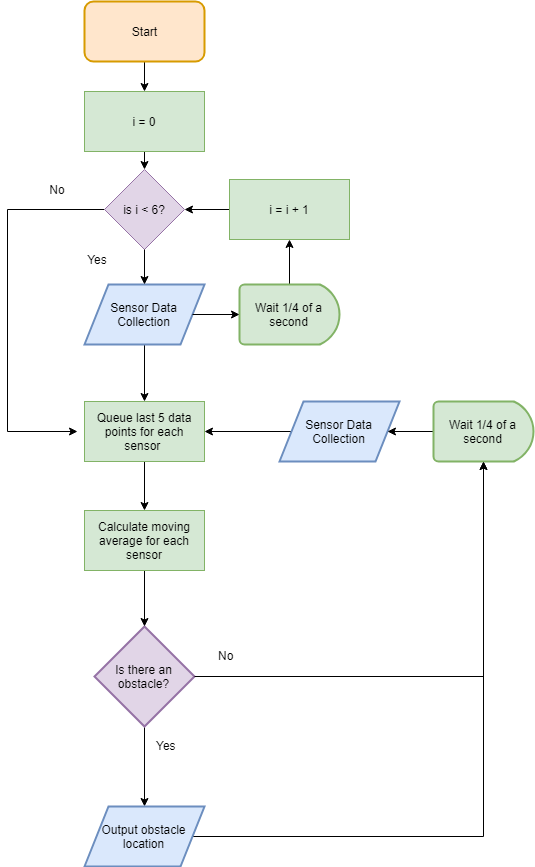
**Criterion B: Design**

**Diagrams**

My other level of detail will be more detail on how it detects an obstacle and what it outputs to the other computer. A possible expansion would involve factoring in the robot moving to determine if the obstacle is moving or if, due to the robot moving, the sensors only make it look like the obstacle is moving.

**Success Criteria**

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Nature of Test** | **Example** |
| System can detect when there is change in distance from the sensors | Check that change in distance is measured. | Moving hand in front of sensor alerts a change in console. |
| An algorithm can determine whether it is more likely that the sensor reading is a false-positive, a change in ground level, or an actual obstacle to avoid. | Moving UGV over a slope and into an object, and seeing if algorithm will give different responses | Moving UGV with system active over a slope, and then towards an object. The system should not alert the UGV the first time, but should the second. |
| System can determine approximately where the obstacle is located with respect to the robot. | Algorithm knows where each sensor is located on the robot to roughly determine where the obstacle it is detecting is. | Upon sensing an obstacle to the right of the robot, the Arduino alerts the UGV about its location |
| System can provide relevant information about what it detects so that the main computer on  the UGV can determine what actions to take. | The Arduino sends messages to the computer on the UGV. | Main computer will respond to created protocol to make sure communication is happening. |