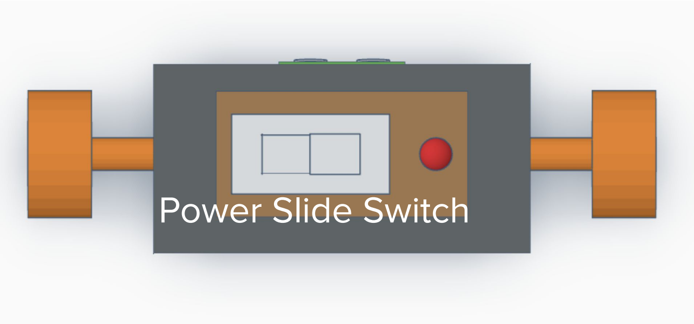
Cleaning Robot

By Xavier Loera Flores

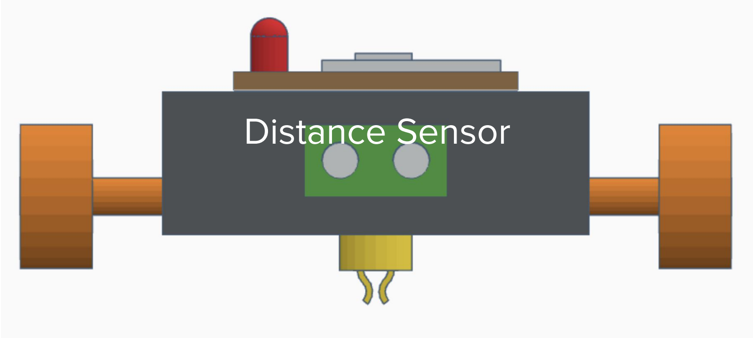
Mechanical Design

Cleaning Robot

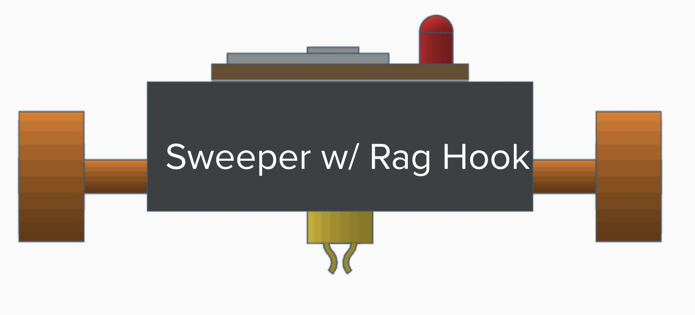
Top View



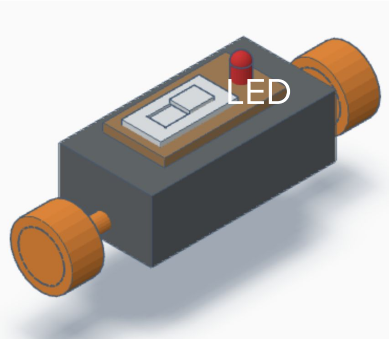
Back View



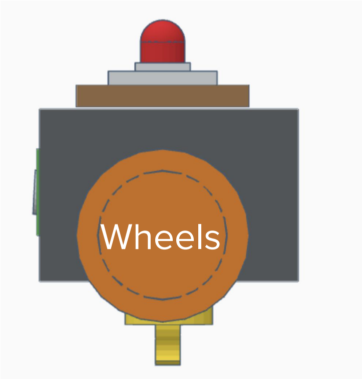
Front View



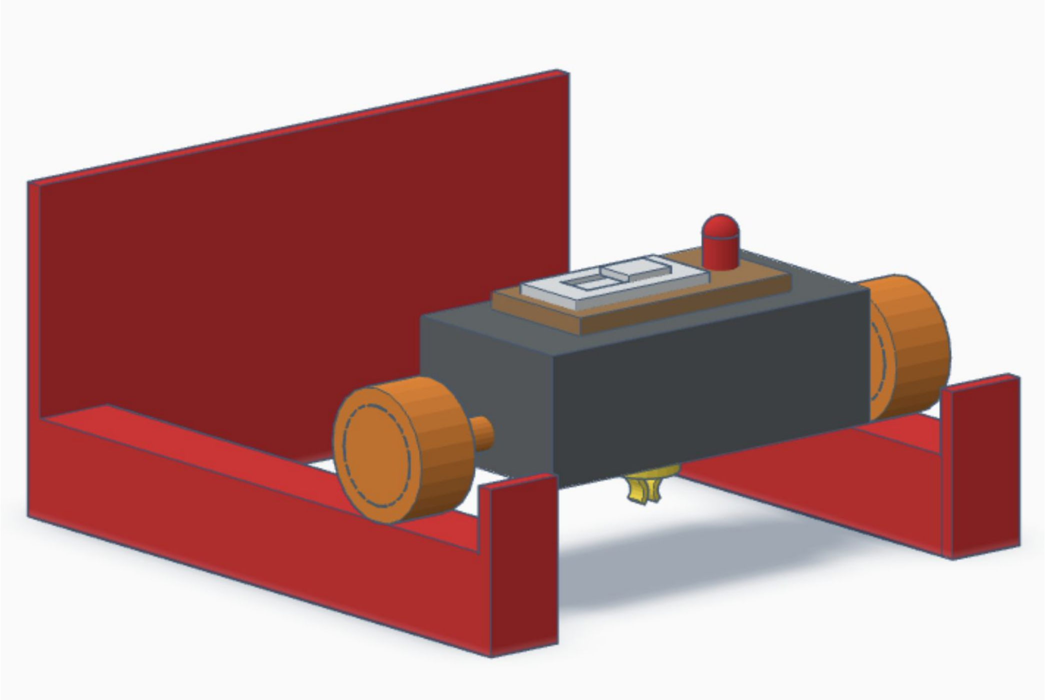
Isomorphic View



Side View



Cleaning Robot Steps



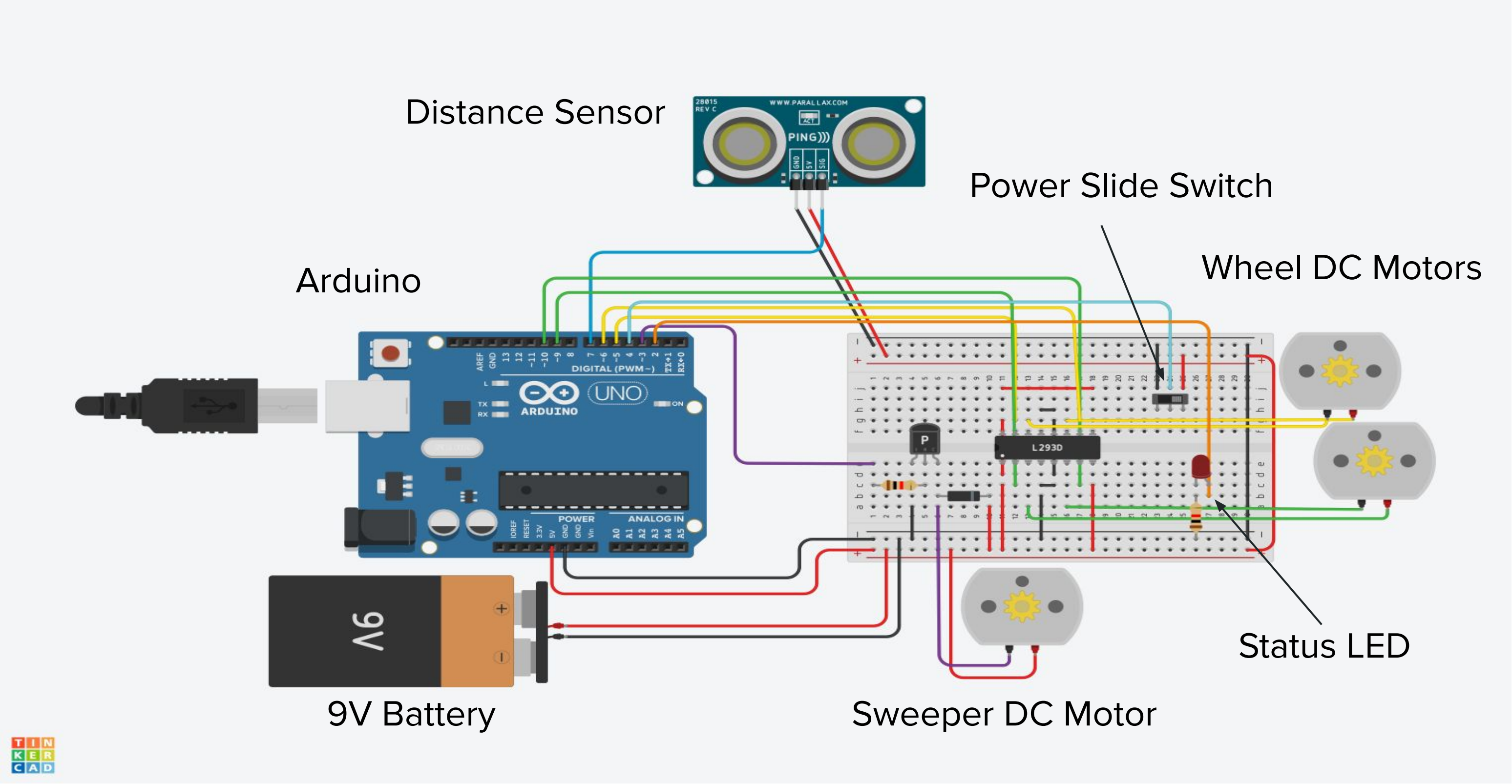
1. **Powered off:** The robot will remain in a permanent standby mode until the power switch has been toggled.
2. **Standby mode:** The robot will wait in standby for a set amount of time and then switch to cleaning mode. While on standby, the wheels, sweeper, & LED are turned off.
3. **Cleaning mode:** The robot will move forward until it surpass the safe working space distance and switches to docking mode. While cleaning, the wheels, sweeper, and LEDs are turned on.
4. **Docking mode:** The robot will move in reverse until it comes close to the wall and switches to standby mode. While docking, the wheels are reverse, the led flashes, and the sweeper shuts off.

Cleaning Robot

By Xavier Loera Flores

Circuit Design

Cleaning Robot: Circuit Design



Distance Sensor

Power Slide Switch

Wheel DC Motors

Arduino

Status LED

9V Battery

Sweeper DC Motor

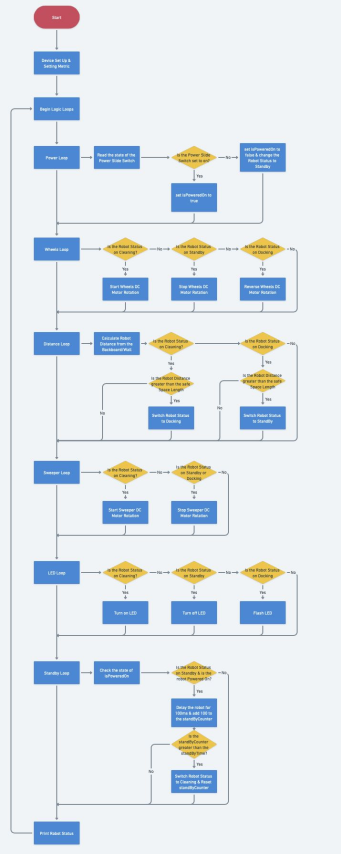
Cleaning Robot

By Xavier Loera Flores

Arduino Code

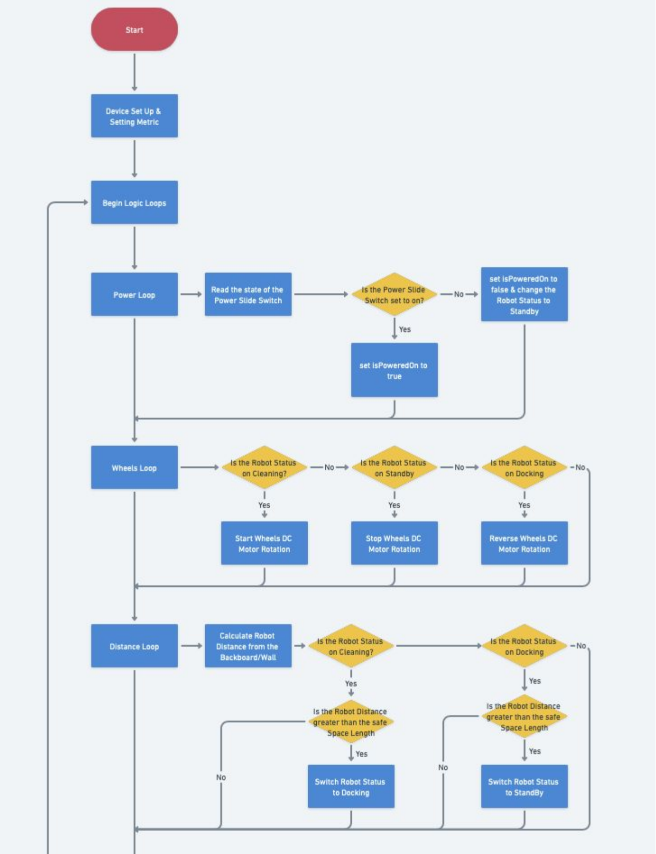
Cleaning Robot: Activity Diagram

Overview Map



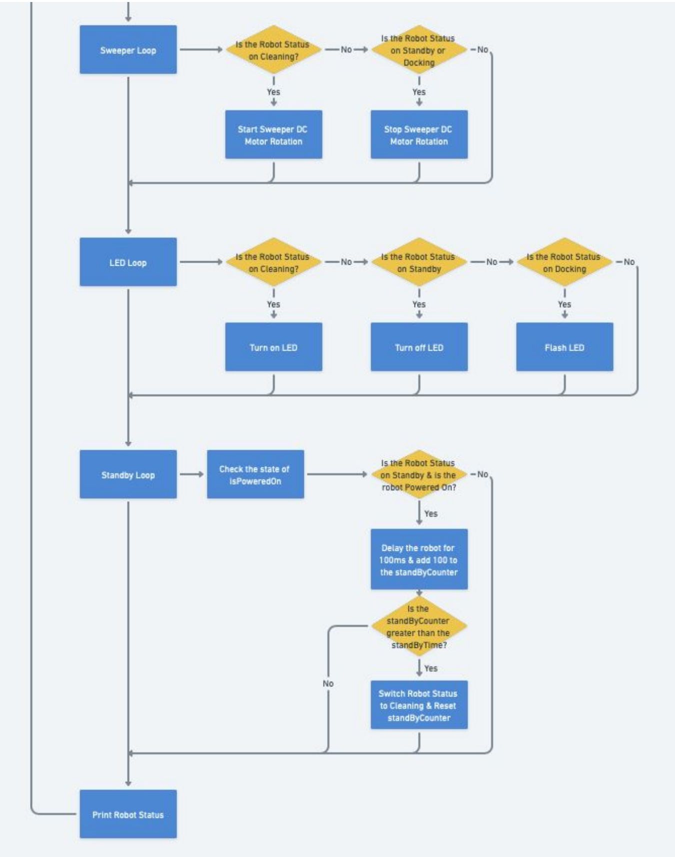
Start
Device Set Up & Setting Metric
Begin Logic Loops
Power Loop
Read the state of the Power Slide Switch
Is the Power Slide Switch set to on?
No
set isPoweredOn to false & change the Robot Status to Standby
Yes
set isPoweredOn to true
Wheels Loop
Is the Robot Status on Cleaning?
No
Is the Robot Status on Standby
No
Is the Robot Status on Docking
No
Yes
Start Wheels DC Motor Rotation
Yes
Stop Wheels DC Motor Rotation
Yes
Reverse Wheels DC Motor Rotation
Distance Loop
Calculate Robot Distance from the Backboard/Wall
Is the Robot Status on Cleaning?
Is the Robot Status on Docking
No
Yes
Is the Robot Distance greater than the safe Space Length
Yes
Is the Robot Distance greater than the safe Space Length
No
Yes
Switch Robot Status to Docking
No
Yes
Switch Robot Status to StandBy
Sweeper Loop
Is the Robot Status on Cleaning?
No
Is the Robot Status on Standby or Docking
No
Yes
Start Sweeper DC Motor Rotation
Yes
Stop Sweeper DC Motor Rotation
LED Loop
Is the Robot Status on Cleaning?
No
Is the Robot Status on Standby
No
Is the Robot Status on Docking
No
Yes
Turn on LED
Yes
Turn off LED
Yes
Flash LED
Standby Loop
Check the state of isPoweredOn
Is the Robot Status on Standby & is the robot Powered On?
No
Yes
Delay the robot for 100ms & add 100 to the standByCounter
Is the standByCounter greater than the standByTime?
No
Yes
Switch Robot Status to Cleaning & Reset standByCounter
Print Robot Status

Top Half



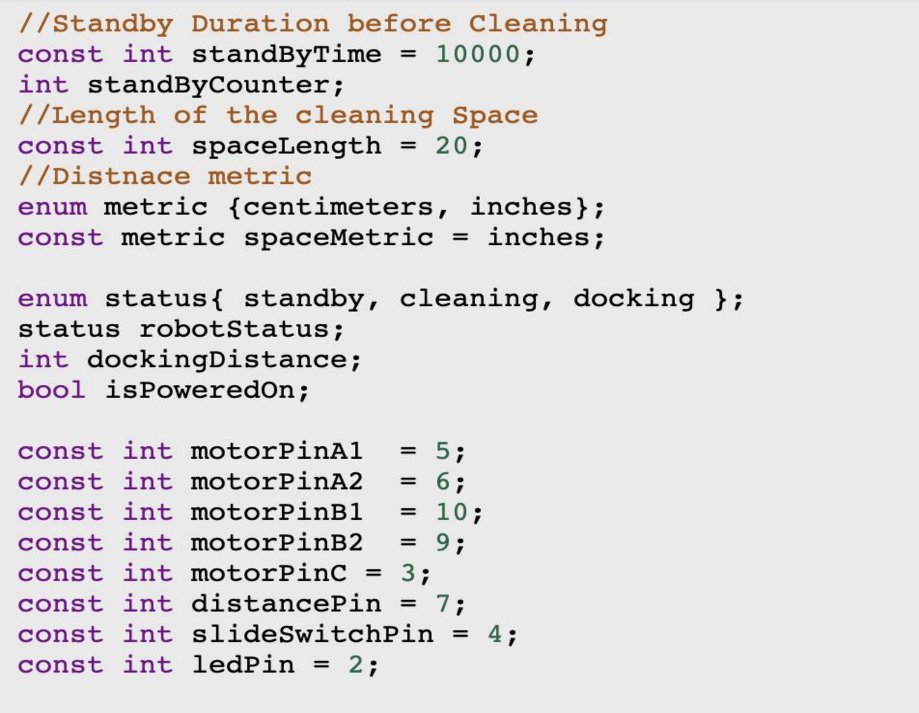
Start
Device Set Up & Setting Metric
Begin Logic Loops
Power Loop
Read the state of the Power Slide Switch
Is the Power Slide Switch set to on?
No
set isPoweredOn to false & change the Robot Status to Standby
Yes
set isPoweredOn to true
Wheels Loop
Is the Robot Status on Cleaning?
No
Is the Robot Status on Standby
No
Is the Robot Status on Docking
No
Yes
Start Wheels DC Motor Rotation
Yes
Stop Wheels DC Motor Rotation
Yes
Reverse Wheels DC Motor Rotation
Distance Loop
Calculate Robot Distance from the Backboard/Wall
Is the Robot Status on Cleaning?
Is the Robot Status on Docking
No
Yes
Is the Robot Distance greater than the safe Space Length
Yes
Is the Robot Distance greater than the safe Space Length
Yes
No
Switch Robot Status to Docking
No
Switch Robot Status to StandBy

Bottom Half

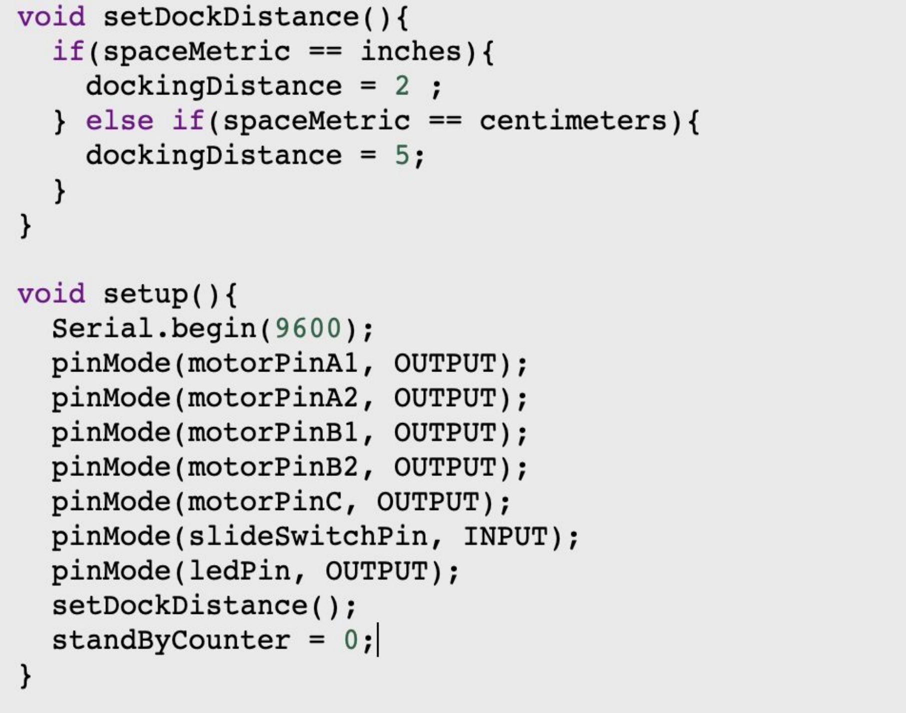


Sweeper Loop
Is the Robot Status on Cleaning?
No
Is the Robot Status on Standby or Docking
No
Yes
Start Sweeper DC Motor Rotation
Yes
Stop Sweeper DC Motor Rotation
LED Loop
Is the Robot Status on Cleaning?
No
Is the Robot Status on Standby
No
Is the Robot Status on Docking
No
Yes
Turn on LED
Yes
Turn off LED
Yes
Flash LED
Standby Loop
Check the state of isPoweredOn
Is the Robot Status on Standby & is the robot Powered On?
No
Yes
Delay the robot for 100ms & add 100 to the standByCounter
Is the standByCounter greater than the standByTime?
Yes
No
Switch Robot Status to Cleaning & Reset standByCounter
Print Robot Status

Cleaning Robot: Code Part 1: Constants, Metric, & Setup

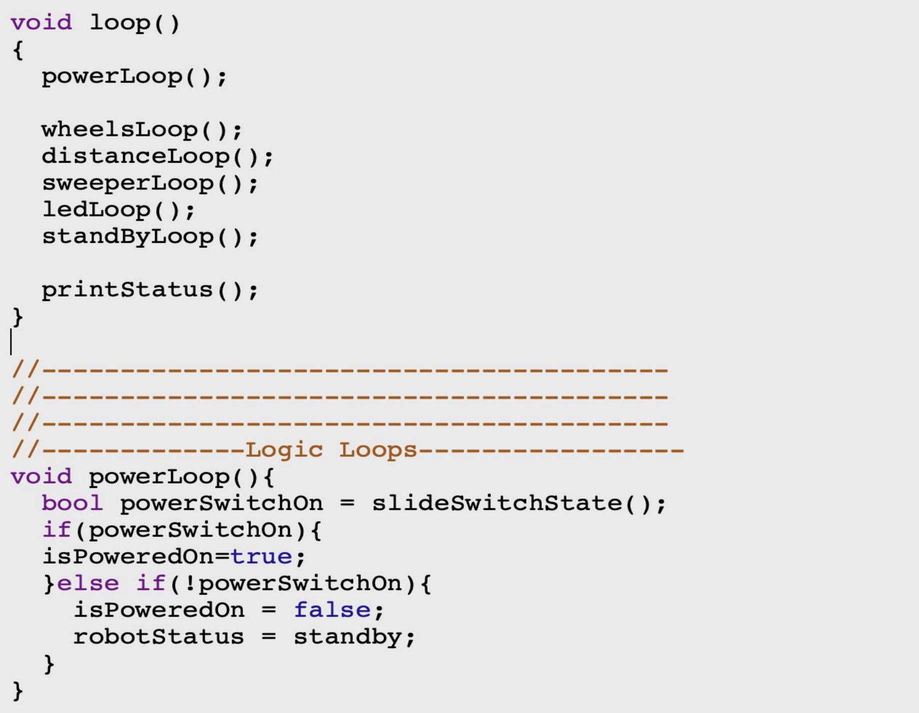


```
//Standby Duration before Cleaning
const int standByTime = 10000;
int standByCounter;
//Length of the cleaning Space
const int spaceLength = 20;
//Distnace metric
enum metric {centimeters, inches};
const metric spaceMetric = inches;

enum status{ standby, cleaning, docking };
status robotStatus;
int dockingDistance;
bool isPoweredOn;

const int motorPinA1  = 5;
const int motorPinA2  = 6;
const int motorPinB1  = 10;
const int motorPinB2  = 9;
const int motorPinC = 3;
const int distancePin = 7;
const int slideSwitchPin = 4;
const int ledPin = 2;
```

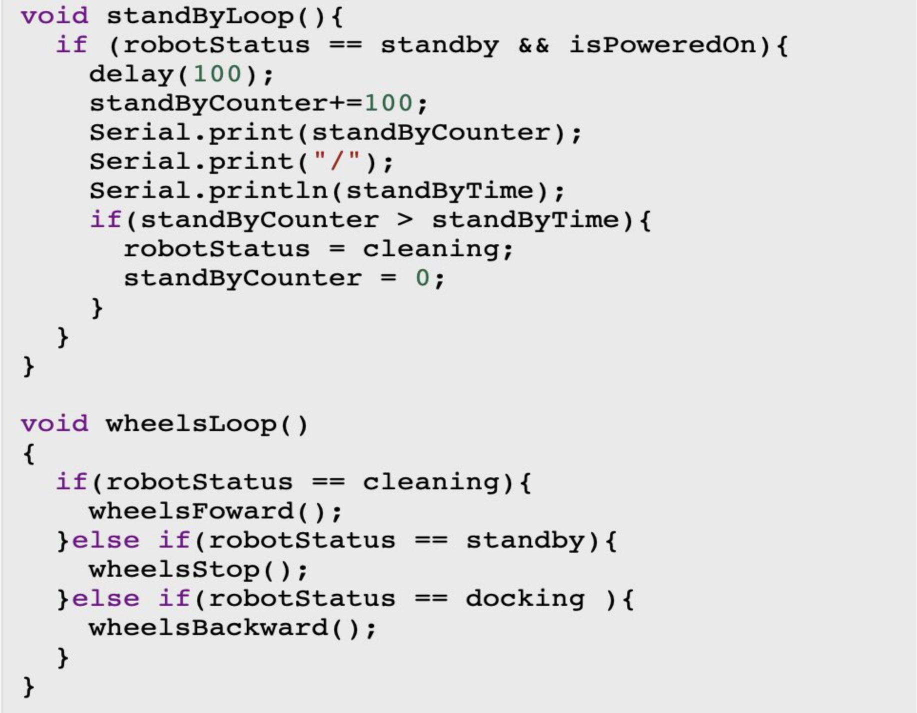


```
void setDockDistance(){
  if(spaceMetric == inches){
    dockingDistance = 2 ;
  } else if(spaceMetric == centimeters){
    dockingDistance = 5;
  }
}

void setup(){
  Serial.begin(9600);
  pinMode(motorPinA1, OUTPUT);
  pinMode(motorPinA2, OUTPUT);
  pinMode(motorPinB1, OUTPUT);
  pinMode(motorPinB2, OUTPUT);
  pinMode(motorPinC, OUTPUT);
  pinMode(slideSwitchPin, INPUT);
  pinMode(ledPin, OUTPUT);
  setDockDistance();
  standByCounter = 0;
}
```

Cleaning Robot: Code Part 2: Power, Standby, Wheels

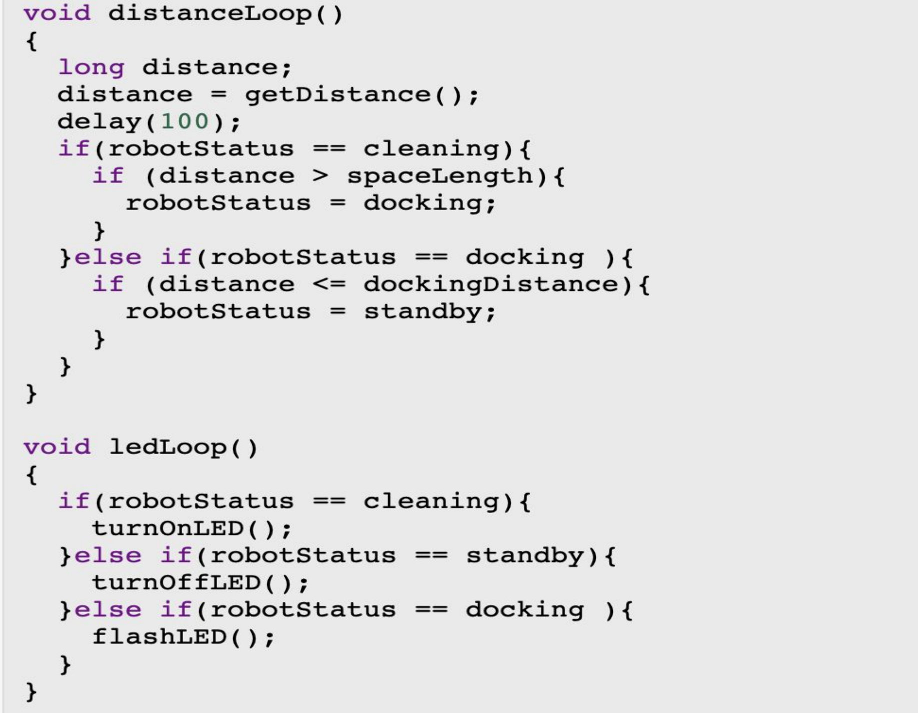


```
void loop()
{
  powerLoop();

  wheelsLoop();
  distanceLoop();
  sweeperLoop();
  ledLoop();
  standByLoop();

  printStatus();
}

//----------------------------------------
//----------------------------------------
//----------------------------------------
//-------------Logic Loops-----------------
void powerLoop(){
  bool powerSwitchOn = slideSwitchState();
  if(powerSwitchOn){
  isPoweredOn=true;
  }else if(!powerSwitchOn){
    isPoweredOn = false;
    robotStatus = standby;
  }
}
```

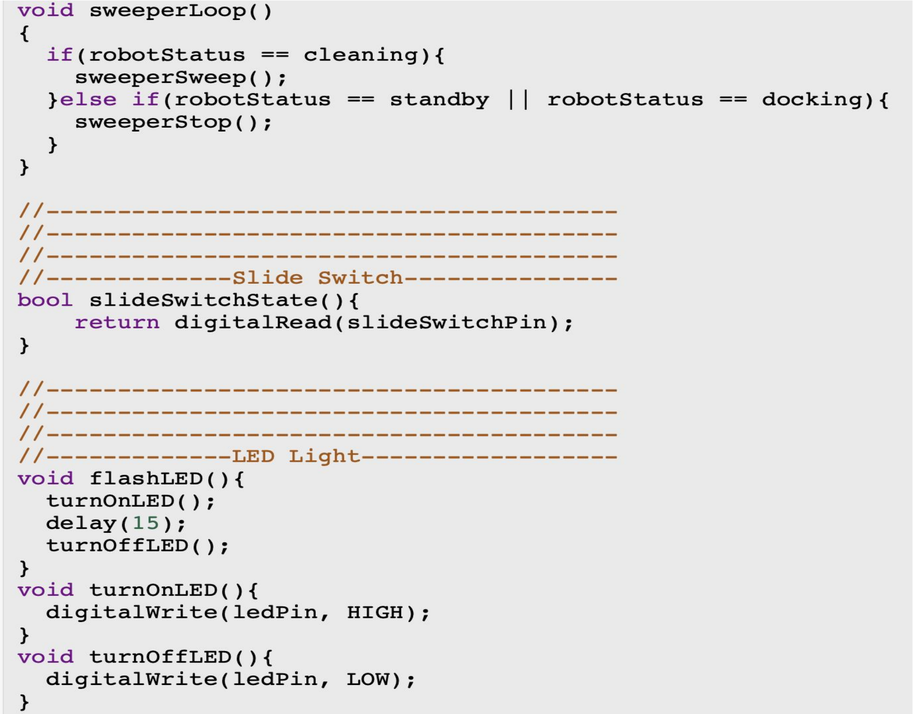


```
void standByLoop(){
  if (robotStatus == standby && isPoweredOn){
    delay(100);
    standByCounter+=100;
    Serial.print(standByCounter);
    Serial.print("/");
    Serial.println(standByTime);
    if(standByCounter > standByTime){
      robotStatus = cleaning;
      standByCounter = 0;
    }
  }
}

void wheelsLoop()
{
  if(robotStatus == cleaning){
    wheelsFoward();
  }else if(robotStatus == standby){
    wheelsStop();
  }else if(robotStatus == docking ){
    wheelsBackward();
  }
}
```

Cleaning Robot: Code Part 3: Distance, LED, Sweeper

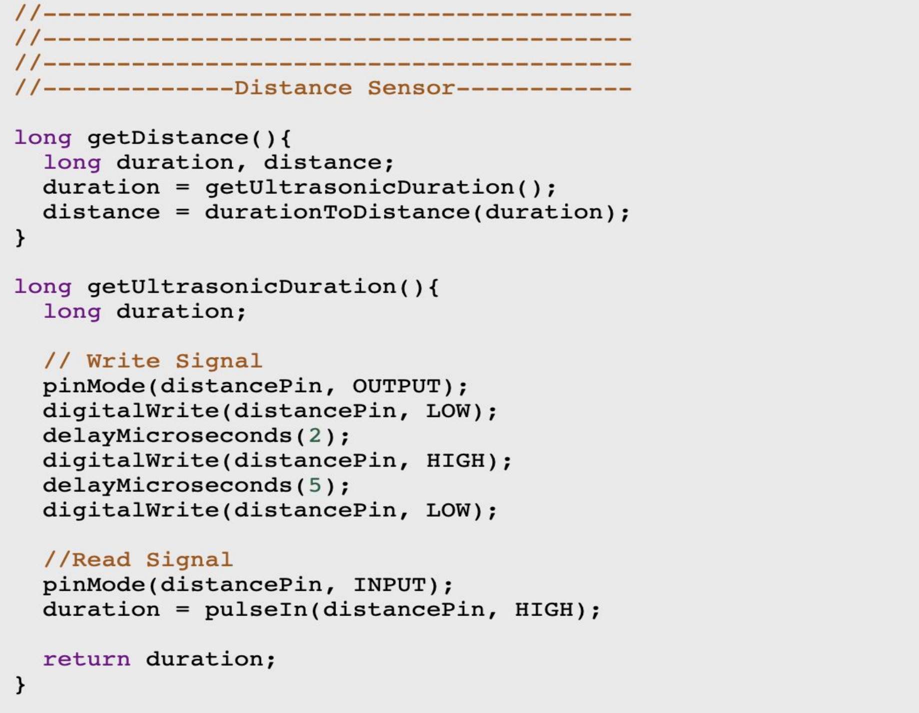


```
void distanceLoop()
{
  long distance;
  distance = getDistance();
  delay(100);
  if(robotStatus == cleaning){
    if (distance > spaceLength){
      robotStatus = docking;
    }
  }else if(robotStatus == docking ){
    if (distance <= dockingDistance){
      robotStatus = standby;
    }
  }
}

void ledLoop()
{
  if(robotStatus == cleaning){
    turnOnLED();
  }else if(robotStatus == standby){
    turnOffLED();
  }else if(robotStatus == docking ){
    flashLED();
  }
}
```

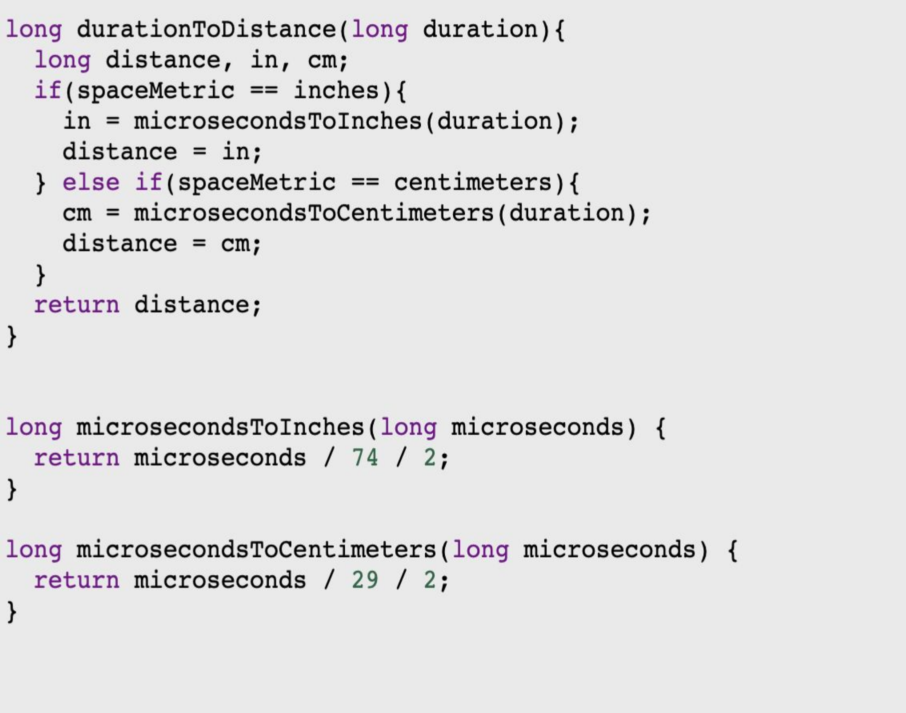


```
void sweeperLoop()
{
  if(robotStatus == cleaning){
    sweeperSweep();
  }else if(robotStatus == standby || robotStatus == docking){
    sweeperStop();
  }
}

//----------------------------------------
//----------------------------------------
//----------------------------------------
//-------------Slide Switch---------------
bool slideSwitchState(){
    return digitalRead(slideSwitchPin);
}

//----------------------------------------
//----------------------------------------
//----------------------------------------
//-------------LED Light------------------
void flashLED(){
  turnOnLED();
  delay(15);
  turnOffLED();
}
void turnOnLED(){
  digitalWrite(ledPin, HIGH);
}
void turnOffLED(){
  digitalWrite(ledPin, LOW);
}
```

Cleaning Robot: Code Part 4: Distance Sensor

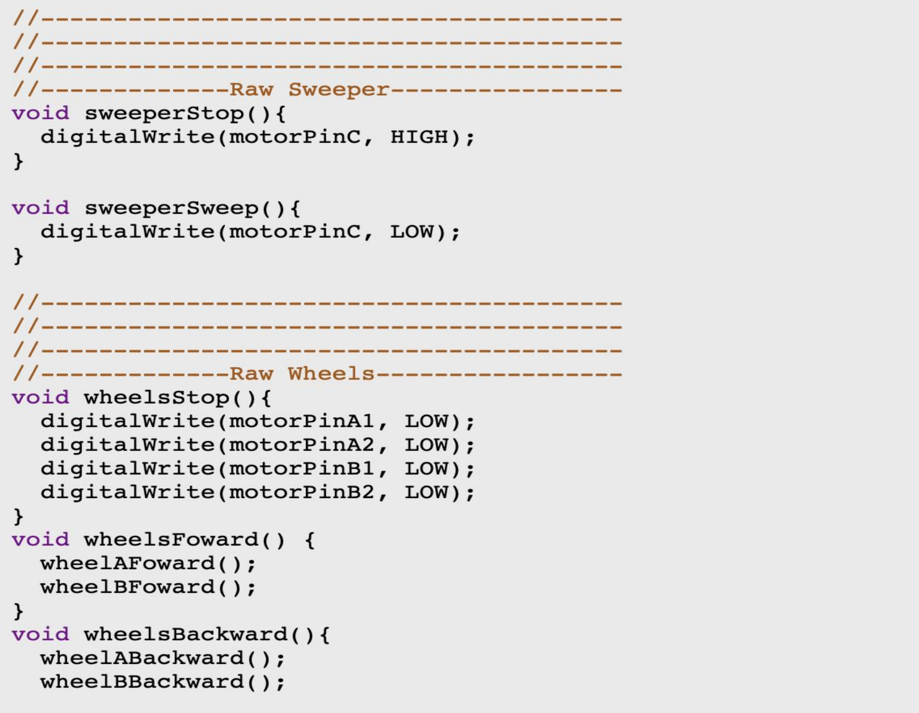


```
//----------------------------------------
//----------------------------------------
//----------------------------------------
//-------------Distance Sensor------------

long getDistance(){
  long duration, distance;
  duration = getUltrasonicDuration();
  distance = durationToDistance(duration);
}

long getUltrasonicDuration(){
  long duration;

  // Write Signal
  pinMode(distancePin, OUTPUT);
  digitalWrite(distancePin, LOW);
  delayMicroseconds(2);
  digitalWrite(distancePin, HIGH);
  delayMicroseconds(5);
  digitalWrite(distancePin, LOW);

  //Read Signal
  pinMode(distancePin, INPUT);
  duration = pulseIn(distancePin, HIGH);

  return duration;
}
```

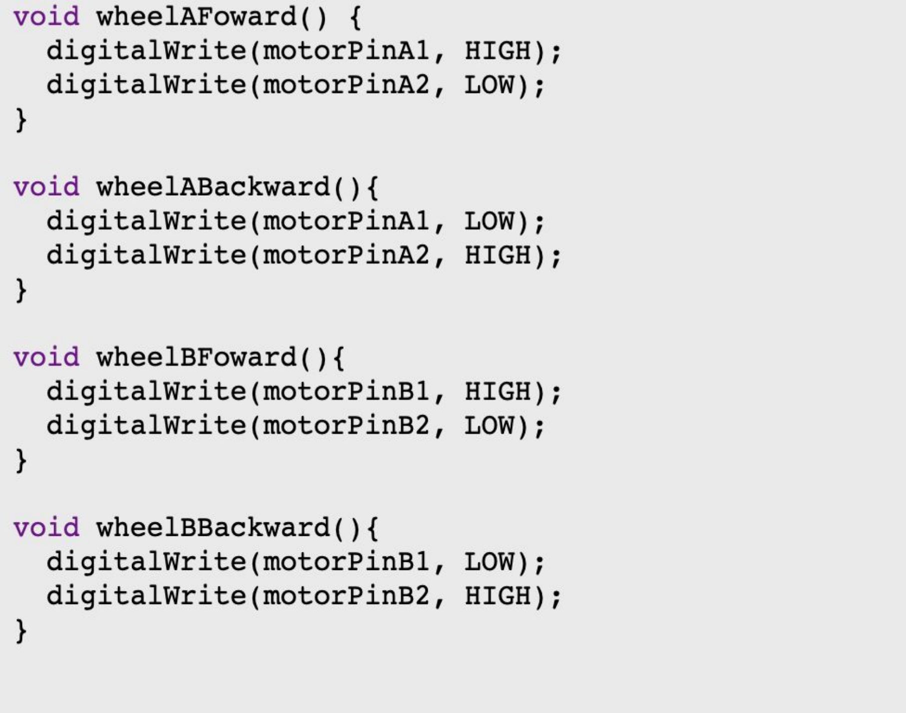


```
long durationToDistance(long duration){
  long distance, in, cm;
  if(spaceMetric == inches){
    in = microsecondsToInches(duration);
    distance = in;
  } else if(spaceMetric == centimeters){
    cm = microsecondsToCentimeters(duration);
    distance = cm;
  }
  return distance;
}


long microsecondsToInches(long microseconds) {
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```

Cleaning Robot: Code Part 5: Sweeper & Wheels

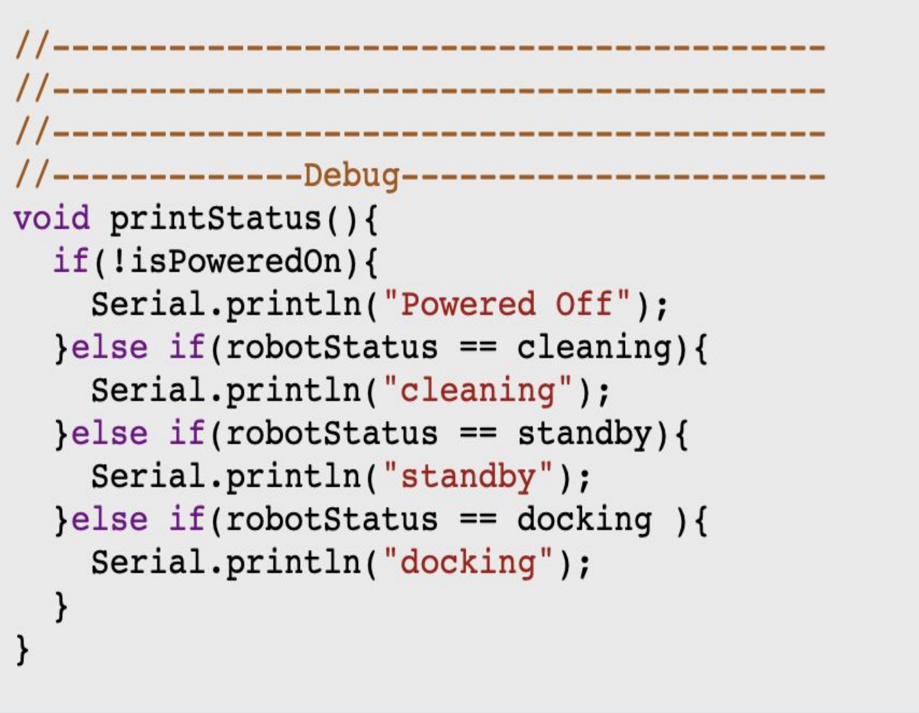


```
//----------------------------------------
//----------------------------------------
//----------------------------------------
//-------------Raw Sweeper----------------
void sweeperStop(){
  digitalWrite(motorPinC, HIGH);
}

void sweeperSweep(){
  digitalWrite(motorPinC, LOW);
}

//----------------------------------------
//----------------------------------------
//----------------------------------------
//-------------Raw Wheels-----------------
void wheelsStop(){
  digitalWrite(motorPinA1, LOW);
  digitalWrite(motorPinA2, LOW);
  digitalWrite(motorPinB1, LOW);
  digitalWrite(motorPinB2, LOW);
}
void wheelsFoward() {
  wheelAFoward();
  wheelBFoward();
}
void wheelsBackward(){
  wheelABackward();
  wheelBBackward();
```

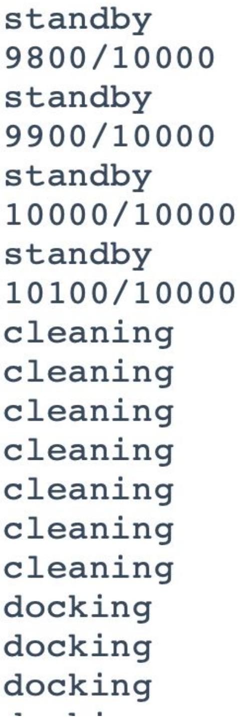


```
void wheelAFoward() {
  digitalWrite(motorPinA1, HIGH);
  digitalWrite(motorPinA2, LOW);
}

void wheelABackward(){
  digitalWrite(motorPinA1, LOW);
  digitalWrite(motorPinA2, HIGH);
}

void wheelBFoward(){
  digitalWrite(motorPinB1, HIGH);
  digitalWrite(motorPinB2, LOW);
}

void wheelBBackward(){
  digitalWrite(motorPinB1, LOW);
  digitalWrite(motorPinB2, HIGH);
}
```

Cleaning Robot: Code Part 6: Print Status



```
//----------------------------------------
//----------------------------------------
//----------------------------------------
//-------------Debug----------------------
void printStatus(){
  if(!isPoweredOn){
    Serial.println("Powered Off");
  }else if(robotStatus == cleaning){
    Serial.println("cleaning");
  }else if(robotStatus == standby){
    Serial.println("standby");
  }else if(robotStatus == docking ){
    Serial.println("docking");
  }
}
```



```
standby
9800/10000
standby
9900/10000
standby
10000/10000
standby
10100/10000
cleaning
cleaning
cleaning
cleaning
cleaning
cleaning
cleaning
docking
docking
docking
```