The PID is an acronym for 'Proportional, integral, differential' controller. It goes kind of like this:

You measure your angle. Your target angle is '0' degrees, right? So, you take your measured angle and subtract your target angle, this gives you an angular error:

e = (a - a0) // a is measured angle, a0 is target angle, e is error

So, now you know how far off you are, you need to decide how much energy to put into the wheels. We'll start simple with just the 'P' or 'proportional part:

f = k \* e; // f is the force to apply to the wheels, e is the error, and k is some constant you make up.

If k is too big, it goes unstable. if too small, your control algorithm can't keep up and balance. You determine k empirically by trying different numbers until you get it about right.

So... in short you 'P' or proportial control algorithm will look like this:

float get\_motor\_power(float angle, float target\_angle, float k) {

float error = angle - target\_angle;

float result = k \* error;

return result;

}

That's it! I did it all in float, but you may be able to change to ints for the parameters,but float computations makes things much easier.

Once you get your proportional algorithm going, you can easily add the integral and differential parts (slightly more complex) to further improve stability and accuracy