

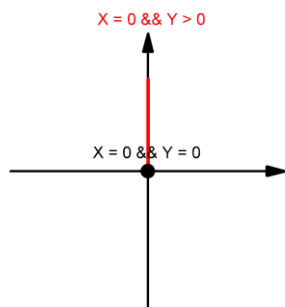
1. Position

Logic: $x == 0 \ \&\& \ y == 0$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: 0
- Right motor forward: 0
- Right motor backward: 0

```
42   if(PWMch[1]>-eps && PWMch[1]<eps && PWMch[2]>-eps &&
43   PWMch[2]<eps)
      Motors(0, 0, 0, 0);
```



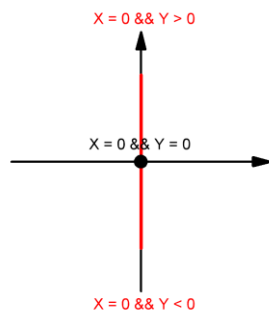
2. Position

Logic: $x == 0 \ \&\& \ y > 0$

PWM duty cycle values:

- Left motor forward: absolute value y
- Left motor backward: 0
- Right motor forward: absolute value y
- Right motor backward: 0

```
45   if(PWMch[1]<eps && PWMch[1]>-eps && PWMch[2]>eps)
46   Motors(PWMabs[2], 0, PWMabs[2], 0);
```



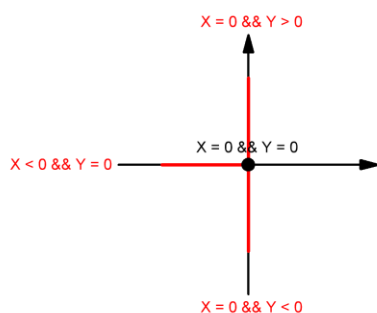
3. Position

Logic: $x == 0 \ \&\& \ y < 0$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: absolute value y
- Right motor forward: 0
- Right motor backward: absolute value y

```
48   if(PWMch[1]<eps && PWMch[1]>-eps && PWMch[2]<-eps)
49   Motors(0, PWMabs[2], 0, PWMabs[2]);
```



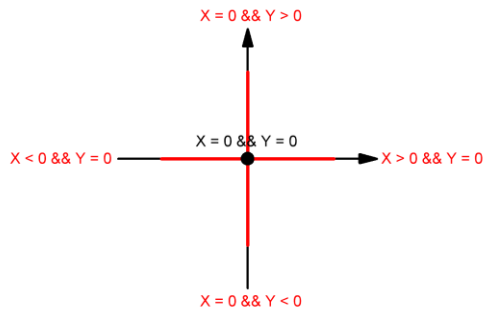
4. Position

Logic: $x < 0 \ \&\& \ y == 0$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: absolute value x
- Right motor forward: absolute value x
- Right motor backward: 0

```
51   if(PWMch[1]>eps && PWMch[2]<eps && PWMch[2]>-eps)
52   Motors(0, PWMabs[1], PWMabs[1], 0);
```



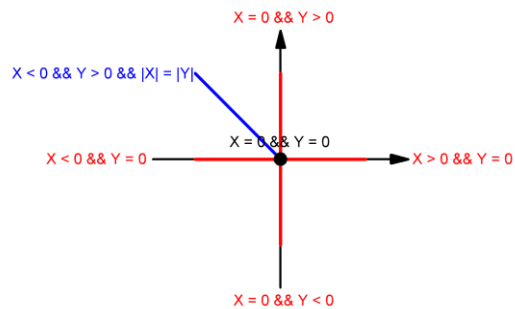
5. Position

Logic: $x > 0 \ \&\& \ Y == 0$

PWM duty cycle values:

- Left motor forward: absolute value x
- Left motor backward: 0
- Right motor forward: 0
- Right motor backward: absolute value x

```
54   if(PWMch[1]<-eps && PWMch[2]<eps && PWMch[2]>-eps)
55   Motors (PWMabs[1], 0, 0, PWMabs[1]);
```



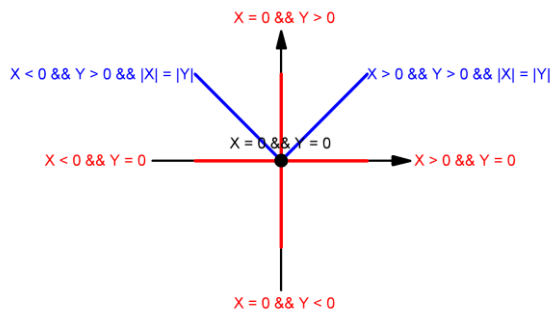
6. Position

Logic: $x < 0 \ \&\& \ y > 0 \ \&\& \ |x| == |y|$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: 0
- Right motor forward: absolute value y
- Right motor backward: 0

```
63   if(PWMch[1]>eps && PWMch[2]>eps && PWMabs[1]-PWMabs[2]<2*eps &&
64   PWMabs[2]-PWMabs[1]<2*eps)
    Motors(0, 0, PWMabs[2], 0);
```



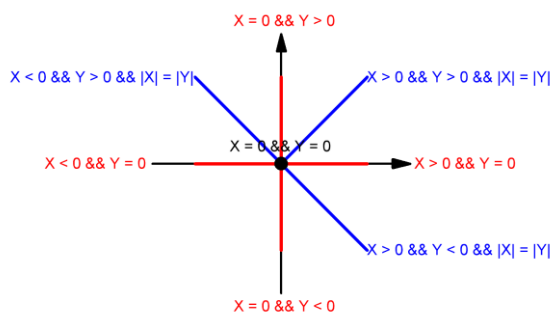
7. Position

Logic: $x > 0 \ \&\& \ y > 0 \ \&\& \ |x| == |y|$

PWM duty cycle values:

- Left motor forward: absolute value x
- Left motor backward: 0
- Right motor forward: 0
- Right motor backward: 0

```
72   if(PWMch[1]<-eps && PWMch[2]>eps && PWMabs[1]-PWMabs[2]<2*eps &&
73   PWMabs[2]-PWMabs[1]<2*eps)
    Motors(PWMabs[1], 0, 0, 0);
```



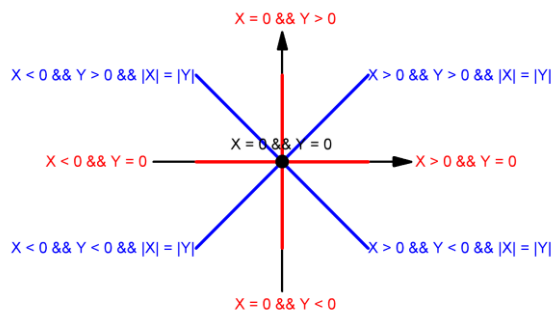
8. Position

Logic: $x > 0 \ \&\& \ y < 0 \ \&\& \ |x| == |y|$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: 0
- Right motor forward: 0
- Right motor backward: absolute value y

```
84   if(PWMch[1]>eps && PWMch[2]<-eps && PWMabs[1]-PWMabs[2]<2*eps &&
85   PWMabs[2]-PWMabs[1]<2*eps)
    Motors(0, 0, 0, PWMabs[2]);
```



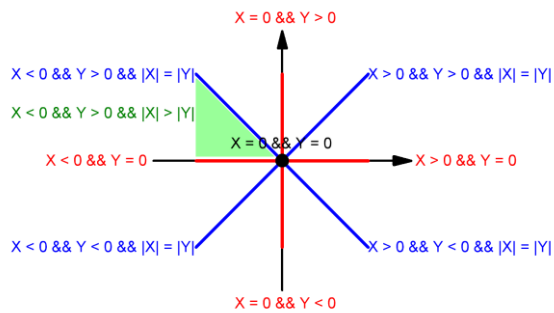
9. Position

Logic: $x < 0 \ \&\& \ y < 0 \ \&\& \ |x| == |y|$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: absolute value y
- Right motor forward: 0
- Right motor backward: 0

```
96   if (PWMch[1]<-eps && PWMch[2]<-eps && PWMabs[1]-PWMabs[2]<2*eps
97   && PWMabs[2]-PWMabs[1]<2*eps)
    Motors(0, PWMabs[2], 0, 0);
```



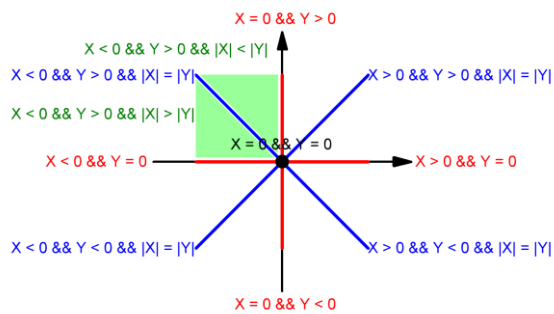
10. Position

Logic: $x < 0 \ \&\& \ y > 0 \ \&\& \ |x| > |y|$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: val $|x| - \text{val } |y|$
- Right motor forward: absolute value x
- Right motor backward: 0

```
57   if (PWMch[1]>eps && PWMch[2]>eps && PWMabs[1]>PWMabs[2]+eps)
58   Motors(0, PWMabs[1]-PWMabs[2], PWMabs[1], 0);
```



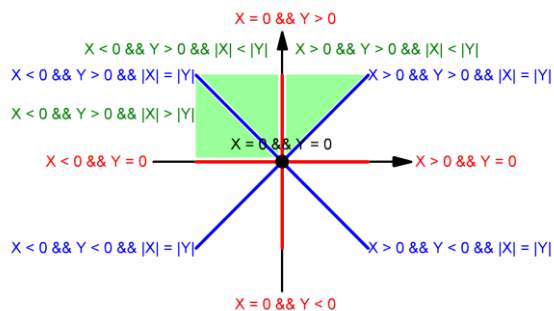
11. Position

Logic: $x < 0 \ \&\& \ y > 0 \ \&\& \ |x| < |y|$

PWM duty cycle values:

- Left motor forward: val $|y| - \text{val } |x|$
- Left motor backward: 0
- Right motor forward: absolute value y
- Right motor backward: 0

```
60   if (PWMch[1]>eps && PWMch[2]>eps && PWMabs[1]+eps<PWMabs[2])
61   Motors(PWMabs[2]-PWMabs[1], 0, PWMabs[2], 0);
```



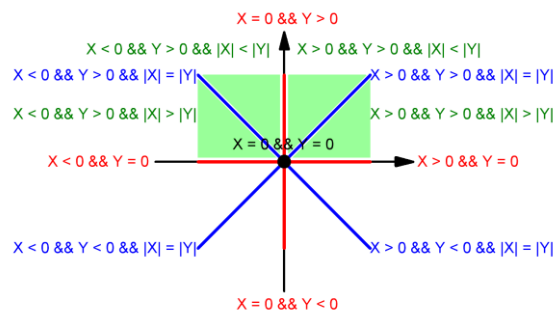
12. Position

Logic: $x > 0 \ \&\& \ y > 0 \ \&\& \ |x| < |y|$

PWM duty cycle values:

- Left motor forward: absolute value y
- Left motor backward: 0
- Right motor forward: val $|y| - \text{val } |x|$
- Right motor backward: 0

```
69   if (PWMch[1]<-eps && PWMch[2]>eps && PWMabs[1]+eps<PWMabs[2])
70   Motors(PWMabs[2], 0, PWMabs[2]-PWMabs[1], 0);
```



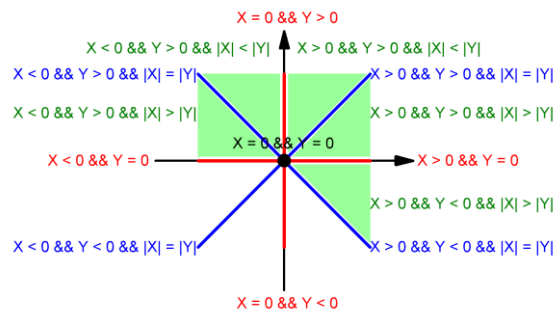
13. Position

Logic: $x > 0 \ \&\& \ y > 0 \ \&\& \ |x| > |y|$

PWM duty cycle values:

- Left motor forward: absolute value x
- Left motor backward: 0
- Right motor forward: 0
- Right motor backward: $\text{val } |x| - \text{val } |y|$

```
66 if (PWMch[1]<-eps && PWMch[2]>eps && PWMabs[1]>PWMabs[2]+eps)
67 Motors(PWMabs[1], 0, 0, PWMabs[1]-PWMabs[2]);
```



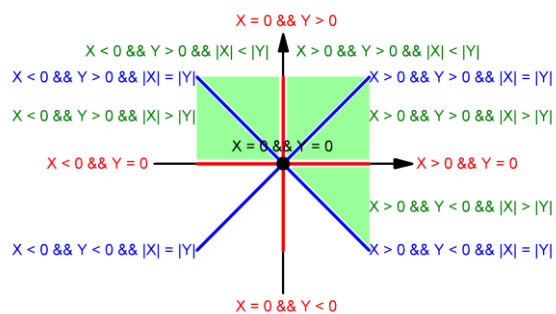
14. Position

Logic: $x > 0 \ \&\& \ y < 0 \ \&\& \ |x| > |y| \ \&\& \ |y| < 85$

PWM duty cycle values:

- Left motor forward: $(\text{val } |x| - \text{val } |y| - 133)*2$
- Left motor backward: 0
- Right motor forward: 0
- Right motor backward: $(\text{val } |x| - \text{val } |y| - 133)*2$

```
87 if (PWMch[1]<-eps && PWMch[2]<-eps && PWMabs[1]>PWMabs[2]+eps &&
88 PWMabs[2]<85)
Motors(backward_A, 0, 0, backward_A);
```



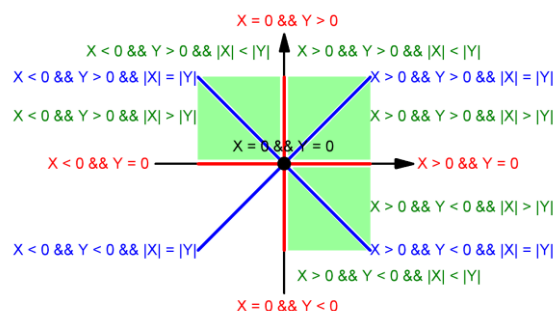
15. Position

Logic: $x > 0 \ \&\& \ y < 0 \ \&\& \ |x| > |y| \ \&\& \ |y| > 170$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: $(\text{val } |x| - \text{val } |y|)*2$
- Right motor forward: $(\text{val } |x| - \text{val } |y|)*2$
- Right motor backward: 0

```
90 if (PWMch[1]<-eps && PWMch[2]<-eps && PWMabs[1]>PWMabs[2]+eps &&
91 PWMabs[2]>170)
Motors(0, backward_B, backward_B, 0);
```



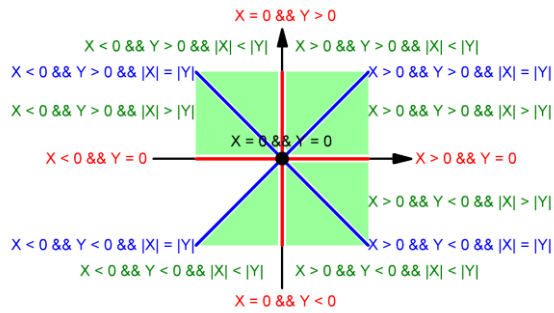
16. Position

Logic: $x > 0 \ \&\& \ y < 0 \ \&\& \ |x| < |y|$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: absolute value y
- Right motor forward: 0
- Right motor backward: absolute value x

```
93 if (PWMch[1]<-eps && PWMch[2]<-eps && PWMabs[1]+eps<PWMabs[2])
94 Motors(0, PWMabs[2], 0, PWMabs[1]);
```



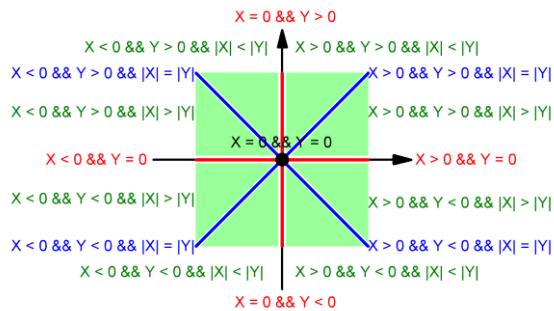
17. Position

Logic: $x < 0 \ \&\& \ y < 0 \ \&\& \ |x| < |y|$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: absolute value x
- Right motor forward: 0
- Right motor backward: absolute value y

```
81 if (PWMch[1]>eps && PWMch[2]<=-eps && PWMabs[1]+eps<PWMabs[2])
82 Motors(0, PWMabs[1], 0, PWMabs[2]);
```



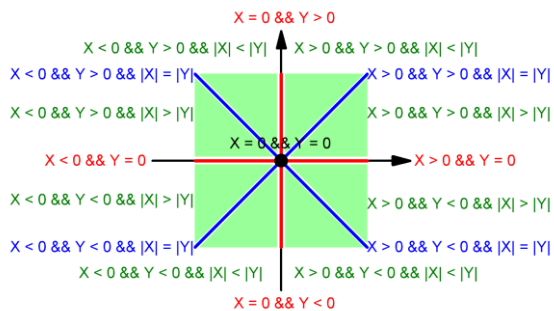
18. Position

Logic: $x < 0 \ \&\& \ y < 0 \ \&\& \ |x| > |y| \ \&\& \ |y| < 85$

PWM duty cycle values:

- Left motor forward: 0
- Left motor backward: $(\text{val } x - \text{val } |y| - 133) * 2$
- Right motor forward: $(\text{val } x - \text{val } |y| - 133) * 2$
- Right motor backward: 0

```
75 if (PWMch[1]>eps && PWMch[2]<=-eps && PWMabs[1]>PWMabs[2]+eps &&
76 PWMabs[2]<85)
Motors(0, backward_A, backward_A, 0);
```



19. Position

Logic: $x < 0 \ \&\& \ y < 0 \ \&\& \ |x| > |y| \ \&\& \ |y| > 170$

PWM duty cycle values:

- Left motor forward: $(\text{val } x - \text{val } |y|) * 2$
- Left motor backward: 0
- Right motor forward: 0
- Right motor backward: $(\text{val } x - \text{val } |y|) * 2$

```
78 if (PWMch[1]>eps && PWMch[2]<=-eps && PWMabs[1]>PWMabs[2]+eps &&
79 PWMabs[2]>170)
Motors(backward_B, 0, 0, backward_B);
```