## 完整型位置型PID程序

 (2012-01-01 19:03:05)

[IMG_256](http://blog.sina.com.cn/s/javascript:;)转载▼

|  |  |
| --- | --- |
| 标签： [杂谈](http://search.sina.com.cn/?c=blog&q=%D4%D3%CC%B8&by=tag" \t "http://blog.sina.com.cn/s/_blank) |  |

3. #include <reg51.h>
4. #include <absacc.h>
5. #include <string.h>
6. #include <stdio.h>
7. #include<math.h>
9. sbit ADC\_INT = P3^2;
10. code unsigned ****char**** Tab[10] ={0x0c0,0x0f9,0x0a4,0x0b0,0x99,0x92,0x82,0x0f8,0x80,0x90};
11. typedef struct   {
13. ****double****  SetPoint;
15. ****double****  Proportion;
16. ****double****  Integral;
17. ****double****  Derivative;
19. ****double****  LastError;
20. ****double****  PrevError;
21. ****double****  SumError;
23. } PID;
24. unsigned ****char**** sensor () ;
25. void actuator(unsigned ****char**** rDelta);
26. void Delay(unsigned ****char**** t);
27. void display(unsigned ****char**** rIn);


31. ****double**** PIDCalc( PID \*pp, ****double**** NextPoint )
32. {
33. ****double****  dError,Error;
35. Error = pp->SetPoint -  NextPoint;
36. pp->SumError += Error;
37. dError = pp->LastError - pp->PrevError;
38. pp->PrevError = pp->LastError;
39. pp->LastError = Error;
40. return(NextPoint+pp->Proportion \* Error + pp->Integral \* pp->SumError + pp->Derivative \* dError  );
41. }

44. void main(void)
45. {
46. PID                 sPID;
47. ****double****              rOut;
48. unsigned ****char****       rIn;
49. ****double****              x;
50. sPID.Proportion = 0.74;
51. sPID.Integral   = 0.70;
52. sPID.Derivative = 0.0;
53. sPID.SetPoint   = 2.0;

56. for (;;)
57. {
58. unsigned ****char****      sumout;
60. unsigned ****char**** i;
61. rIn = sensor ();
62. for(i=0;i<50;i++)
63. display(rIn);
64. x = 5.0 \* (****double****)rIn / 256.0;
65. rOut = PIDCalc ( &sPID,x );
66. sumout=rOut\*256/5;
67. actuator ( sumout );
68. }
69. }


73. void actuator(unsigned ****char**** rDelta)
74. {
76. XBYTE[0xA000] = rDelta;
77. Delay(10);
78. }


82. void Delay(unsigned ****char**** t)
83. {
84. TMOD &= 0xF0;
85. TMOD |= 0x01;
86. EA=1;
87. EX0=1;
88. do
89. {
90. TH0 = 0xee;
91. TL0 = 0x00;
92. TR0 = 1;
93. while ( !TF0 );
94. TR0 = 0;
95. TF0 = 0;
96. } while ( --t != 0 );
97. }

100. unsigned ****char**** sensor (void)
101. {
102. unsigned ****char**** v;
103. Delay(30);
104. XBYTE[0xB000] = 0xFF;
105. Delay(1);
106. while ( ADC\_INT );
107. v = XBYTE[0xB000];
108. return (v);
109. }
111. void display(unsigned ****char**** rIn)
112. {
113. ****float**** num;
114. ****int**** N;
115. num=5.0\*rIn/256;
116. N=(****int****)(num\*1000.0);
117. XBYTE[0x8000]=0x08;
118. XBYTE[0x9000]=Tab[N/1000]&0x7f;
119. Delay(1);
120. XBYTE[0x8000]=0x04;
121. XBYTE[0x9000]=Tab[N00/100];
122. Delay(1);
123. XBYTE[0x8000]=0x02;
124. XBYTE[0x9000]=Tab[N0/10];
125. Delay(1);
126. XBYTE[0x8000]=0x01;
127. XBYTE[0x9000]=Tab[N];
128. Delay(1);}