【作業内容】

文字の担当を決める

決まった担当の文字をどのようにするか検討する

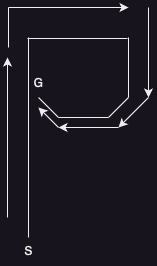
【作業項目】

1. 文字の担当を決める。

自分はPの文字になった。

1. Pの文字を攻略するための方法を検討する

Pは比較的簡単な文字だと考える。やり方としていくつか考えようかと思ったが、どう考えても１つしかやり方が浮かばないのでそのやり方でプログラムを変えてみる。



このように右回りで回っていく。前にラインが繋がっていなければ左のモーターの出力を強くする。このようにやれば右回りでコースを全て走破することができる。

1. このコースをクリアするプログラムを以下に示す。

|  |
| --- |
| 1. #include <stdio.h> 2. #include <stdlib.h> 3. #include <wiringPi.h> 4. #include <wiringPiI2C.h> 5. // PWMユニットのI2Cアドレス 6. #define PWMI2CADR 0x40 7. // PWM制御に使う。（１でブリッジ動作、0はブリッジオフ） 8. #define ENA\_PWM 8 9. // IN1とIN2は右車輪の回転方向を決める（後進：0, 1, 前進：1, 0）（0, 0と1, 1はブレーキ） 10. #define IN1\_PWM 9 11. #define IN2\_PWM 10 12. // 左側のモーター：パワーユニットのK3またはK4に接続 13. // ENBはPWM駆動に使う（1でブリッジ動作、0はブリッジオフ） 14. #define ENB\_PWM 13 15. // IN3とIN4は左車輪の回転方向を決める（後進：0, 1、前進：1, 0）（0, 0と1, 1はブレーキ） 16. #define IN3\_PWM 11 17. #define IN4\_PWM 12 18. // PWMモジュールのレジスタ番号 19. #define PWM\_MODE1 0 20. #define PWM\_MODE2 1 21. #define PWM\_SUBADR1 2 22. #define PWM\_SUBADR2 3 23. #define PWM\_SUBADR3 4 24. #define PWM\_ALLCALL 5 25. // PWM番号＊4＋PWM\_0\_??\_?でレジスタ番号は求まる 26. #define PWM\_0\_ON\_L 6 27. #define PWM\_0\_ON\_H 7 28. #define PWM\_0\_OFF\_L 8 29. #define PWM\_0\_OFF\_H 9 30. // PWM出力定数 31. #define PWMFULLON 16 32. #define PWMFULLOFF 0 33. // プリスケーラのレジスタ番号 34. // PWM周波数を決めるレジスタ番号、100Hzなら61をセット 35. #define PWM\_PRESCALE 254 36. // 光センサーのピン番号 37. #define GPIO\_L 5 38. #define GPIO\_ML 6 39. #define GPIO\_M 13 40. #define GPIO\_MR 19 41. #define GPIO\_R 26 42. // motor\_drive()から呼ばれる関数、PWMユニットへの書き込みをする。 43. int set\_pwm\_output(int fd, int pwmch, int outval) { 44. int ef = 0; 45. int regno; 46. if ((pwmch < 0) || (pwmch > 15)) ef = 1; // チャネルの指定違反チェック 47. if ((outval < 0) || (outval > 16)) ef = ef + 2; // 出力値の指定違反チェック 48. if (ef == 0) { 49. regno = PWM\_0\_ON\_L + pwmch \* 4; // 1chあたり4レジスタで16ch分あるので 50. if (outval == 16) { 51. wiringPiI2CWriteReg8(fd, regno + 3, 0); 52. wiringPiI2CWriteReg8(fd, regno + 1, 0x10); 53. } else { 54. wiringPiI2CWriteReg8(fd, regno + 1, 0); 55. wiringPiI2CWriteReg8(fd, regno + 3, outval); 56. } 57. } 58. return ef; // エラーがなければ０が返る 59. } 60. // モーターを制御するための関数。 61. // fdはI2C初期化時のファイルディスクリプタ（デバイス番号のようなもの） 62. // lmは左モーター、rmは右モーターの駆動数値で、-16~+16の範囲で指定 63. // 負の場合は後ろ方向に回転、生の場合は前方向に回転 64. // 全体値が大きいほど、パワーが大きくなる 65. // PWMユニット自体は12ビット制度だが、上位４ビット分を制御 66. int motor\_drive(int fd, int lm, int rm) { 67. set\_pwm\_output(fd, ENA\_PWM, 0); // 右のモーター無効化 68. set\_pwm\_output(fd, ENB\_PWM, 0); // 左のモーター有効化 69. // 右モーターの制御 70. if (rm < 0) { 71. set\_pwm\_output(fd, IN1\_PWM, 0); // OUT -> GND 72. set\_pwm\_output(fd, IN2\_PWM, rm); // OUT2 -> +Vs 73. rm = abs(rm); 74. } else { 75. set\_pwm\_output(fd, IN1\_PWM, rm); // OUT1 -> +Vs 76. set\_pwm\_output(fd, IN2\_PWM, 0); // OUT2 -> GND 77. } 78. // 左モーターの制御 79. if (lm < 0) { 80. set\_pwm\_output(fd, IN3\_PWM, 0); // OUT3 -> GND 81. set\_pwm\_output(fd, IN4\_PWM, lm); // OUT -> +Vs 82. lm = abs(lm); 83. } else { 84. set\_pwm\_output(fd, IN3\_PWM, lm); // OUT3 -> +Vs 85. set\_pwm\_output(fd, IN4\_PWM, 0); // OUT4 -> GND 86. } 87. if (lm > 16) lm = 16; 88. if (rm > 16) rm = 16; 89. set\_pwm\_output(fd, ENA\_PWM, rm); // 右モータースタート 90. set\_pwm\_output(fd, ENB\_PWM, lm); // 左モータースタート 91. return 0; 92. } 93. int main() { 94. int fd; 95. wiringPiSetupGpio(); 96. fd = wiringPiI2CSetup(PWMI2CADR); 97. if (fd < 0) { 98. printf("I2Cの初期化に失敗しました。終了します。\n"); 99. exit(EXIT\_FAILURE); 100. } 101. wiringPiI2CWriteReg8(fd, PWM\_PRESCALE, 61); 102. wiringPiI2CWriteReg8(fd, PWM\_MODE1, 0x10); 103. wiringPiI2CWriteReg8(fd, PWM\_MODE1, 0); 104. delay(1); 105. wiringPiI2CWriteReg8(fd, PWM\_MODE1, 0x80); 106. while(1){ 107. if(digitalRead(GPIO\_L) == LOW && digitalRead(GPIO\_R) == LOW) break; 108. } 109. int ms, ls, rs; 110. while (1) { 111. ms = 0; 112. ls = 0; 113. rs = 0; 114. if (digitalRead(GPIO\_L) == HIGH) { 115. printf("right\n"); 116. rs = 5; 117. } 118. else if (digitalRead(GPIO\_ML) == HIGH) { 119. printf("middle right\n"); 120. rs = 3; ms=2; 121. } 122. else if (digitalRead(GPIO\_M) == HIGH) { 123. printf("middle\n"); 124. ms = 6; 125. } 126. else if (digitalRead(GPIO\_MR) == HIGH) { 127. printf("middle left\n"); 128. ls = 3; ms=2; 129. } 130. else if (digitalRead(GPIO\_R) == HIGH) { 131. printf("left\n"); 132. ls = 5; 133. } 134. else { 135. printf("not\_read\n"); 136. ls=6; 137. } 138. motor\_drive(fd, ms+ls, ms+rs); 139. delay(50); 140. } 141. return 0; 142. } |

④ 改善点

このプログラムを実行すると、カーブ部分でコースから外れてしまった。次はパラメータの調整をしていきたいと思う。

【作業時間】

・作業時間：90分

・報告書作成時間：30分