## 杭州电子科技大学创新实践实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学院 | 计算机学院 | 班级 |  | 学号 |  |
| 姓名 |  | 日期 | 2024.4.08 | 成绩 |  |
| 实验题目 | 小车位置判断 | | | | |
|  | | | | | |
| 实验目的 | 1、学习网站上模块6的视频部分，下载“实验”pdf，重点学习实验部分6.2-6.4（6.4.4）  2、完成maze代码中的Lab06\_GPIO（Program6-2函数）  3、补全函数Reflectance\_Center和Reflectance\_Position函数  购买电工黑胶布，在实验室地板上贴条黑色线出来，黑线宽度2厘米  实验最终结果要能判断小车的位置，是在线中心，偏左，偏右，线外  报告中要求记录不同状态下的position数值（开发环境截图） | | | | |
| 硬件原理 | 小车位置判断原理：  根据红外对管原理，小车底部的传感器八个管脚对黑色和白色的反馈值分别为1和0。我们可以将把八个二进制测量值组合成一个参数，表示机器人远离线路中心的数量。假设传感器 P7.0位于机器人的右侧，距离中线 28 毫米。另外假设传感器P7.7位于机器人的左侧，距离中线-28 毫米。另一个目标是使这个参数对角度不敏感。如果传感器运行正常，存储在 Data 中的 8 位二进制数分为四类：  1) 全部为0（偏离线路或白色平面）  2)左边为0，右边为1，例如 00000111 （偏左）  3)左右为0，中间为1，例如 00110000 (在线上)  4)左边为1，右边为0，例如 11110000 （偏右）  对于 Reflectance\_Read 函数返回的每个二进制位，将 bi 定义为 0（白色）或 1 （黑色）。 一种可能的传感器集成功能是计算 8 个传感器读数或二进制结果的加权平均值。假设至少有一个黑色读数，估计距离 Reflectance\_Position 为： | | | | |
| 关键代码及注释 | // ------------Reflectance\_Center------------  // Read the two center sensors  // Turn on the 8 IR LEDs  // Pulse the 8 sensors high for 10 us  // Make the sensor pins input  // wait t us  // Read sensors  // Turn off the 8 IR LEDs  // Input: time to wait in usec  // Output: 0 (off road), 1 off to left, 2 off to right, 3 on road  // (Left,Right) Sensors  // 1,1 both sensors on line  // 0,1 just right off to left  // 1,0 left left off to right  // 0,0 neither lost  // Assumes: Reflectance\_Init() has been called  uint8\_t **Reflectance\_Center**(uint32\_t time){  // write this as part of Lab 6  uint8\_t result;  P5->OUT |= BIT3; // Turn on IR light  P7->DIR |= 0xFF; // P7.0-7.7 output  P7->OUT |= 0xFF; // Set P7.0-7.7 high  Clock\_Delay1us(10);  P7->DIR &= ~0xFF; // P7.0-7.7 input  Clock\_Delay1us(time);  result = (P7->IN>>3)&0x03; // Only read the two center sensors  P5->OUT &= ~BIT3; // Turn off IR light  **return** result;  }  // Perform sensor integration  // Input: data is 8-bit result from line sensor  // Output: position in 0.1mm relative to center of line  // 281 Off the track  // -280 to 0 Off to the left  // 0 to 280 Off to the right  int32\_t **Reflectance\_Position**(uint8\_t data){  // write this as part of Lab 6  uint8\_t i=0;  int32\_t position=0, sum\_b=0, sum\_bw=0;  **const** int32\_t w[8] = {-280,-200,-120,-40,40,120,200,280};  int8\_t b[8] = {0,0,0,0,0,0,0,0};  int32\_t bw[8] = {0,0,0,0,0,0,0,0};  **if**(!data) **return** 281; // Off the track  **else**{  **for**(i=0;i<8;i++){  b[i] = (data>>i)&0x01;  sum\_b += b[i];  bw[i] = w[i]\*b[i];  sum\_bw += bw[i];  }  position = sum\_bw/sum\_b;  **return** position;  }  } | | | | |
| 实验步骤 | 将函数Reflectance\_Center和Reflectance\_Position函数补全，给小车接上电源，运行代码，手动放置小车，判断小车和黑色胶带的相对位置。 | | | | |
| 实验结果 | 偏左：00000011    偏右：10000000    在线上：00011000    偏离：00000000 | | | | |
| 思考与反馈 | 小车位置判断实验代码无误，能够正确判断小车位置，但是无法判断小车和路线的夹角，这是后面做小车巡航实验的一个重难点。 | | | | |