# 电控题目

电控基础：30分 C语言基础：50分 信息检索：20分

电控基础题目（总分30分；选择：15分；简答15分）

简答：

1. 写出步兵机器人三种电机的型号及其对应的电调型号

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

2．用自己的话概括出编码器的作用

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

3.写出超级对抗赛（7V7）中所有兵种的名称

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

1. 写出三种通讯协议

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

1. 描述pid算法在机器人控制中的作用及优点

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

选择：

1. GPIO口输出的是（）
2. 模拟信号
3. 数字信号
4. 能量信号
5. 随机信号
6. 在步兵机器人中用于检测系统姿态的硬件是什么
7. 图传模块
8. Minipc
9. 中心板
10. 陀螺仪
11. GPIO口高电平输出\_\_\_\_低电平输出\_\_\_\_

A.0 1

B.1 0

C.255 0

D.0 1024

4.动态系统中在含有不确定信息的情况下，需要对下一步的走向做出有根据的预测，此时可以用到（）

A.PID算法控制

B.DWA算法

C.卡尔曼滤波

D.模糊控制

5.PWM，全称Pulse Width Modulation，翻译过来就是脉冲宽度调制，是种把模拟信号调制成脉波的技术。它已经是应用非常广泛的显示器/光源的亮度控制方案，通过调整占空比来调整显示器或光源的亮度。请问若想调低显示器亮度，此时应该（）占空比

A 调大 B 调小

C语言基础（总分：50分）（选择20分，程序补全30分）

选择（20分，每题2分）

1.下列不是C语言基本数据类型的是()

A.字符型

B.整型

C.浮点型

D.结构体

2. 哪个语句可用于选择多个代码块中的一个来执行？

A. default

B. break

C. when

D. switch

3. 若有定义：char c; int d; 程序运行时输入：c=1,d=2<回车>，

能把字符 1 输入给变量 c、整数 2 输入给变量 d 的输入语句是

A.scanf(“c=%c,d=%d”, &c,&d);

B.scanf(“c=%c d=%d”, &c,&d);

C.scanf(“c=%d,d=%d”, &c,&d);

D.scanf(“c=%d d=%d”, &c,&d);

4. 哪个关键字用于创建结构体？

A. str

B. structure

C. struct

D. structs

5. 在C语言中，下列哪个运算符用于获取变量的地址？

A. &

B. \*

C. %

D. #

6.

#include <stdio.h>

int main() {

int x = 5;

x = x + 3;

printf("%d\n", x);

return 0;

}

选择题1：以上程序的输出结果是什么？

A) 5

B) 8

C) 3

D) 无法确定

7.

int main() {

int x = 10;

int y = 20;

int z;

z = (x > y) ? x : y;

printf("z = %d\n", z);

return 0;

}

以上程序的输出结果是什么？

A) z = 10

B) z = 20

C) z = 30

D) z = 0

8.

#include <stdio.h>

int main() {

int num;

printf("请输入一个整数: ");

scanf("%d", &num);

if (num % 2 == 0) {

printf(“1\n");

} else {

printf("2\n");

}

return 0;

}

如果用户输入数字5，以上程序将输出什么？

A) 1。

B) 2。

C) 无法确定

D) 程序错误

9.

执行下列程序后 x,y,z的值分别是\_\_\_\_\_\_

int x=10,y=20,z=30;

if(x>y)

z=x;

x=y;

y=z;

A 10;20;30

B 20;30;30

C 20;30;10

D 20;30;20

10.

#include <stdio.h>

int main() {

int arr[] = {1, 2, 3, 4, 5};

int \*ptr = arr;

printf("%d", \*ptr + 4);

return 0;

}

A. 1

B. 2

C. 5

D. 9

程序补全（30分，写不出程序写出逻辑也会给分）

1.（5分）

#include <stdio.h>

int main() {

int x = 5;

int y = 10;

// 请在此处添加代码，将 x 和 y 的和存储在变量 sum 中

printf("Sum: %d\n", sum);

return 0;

}

2.（10分）

#include <stdio.h>

int main() {

int n;

printf("请输入一个整数：");

scanf("%d", &n);

// 请在此处添加代码，计算并输出 n 的阶乘

return 0;

}

3.（15分）

任务：编写一个PID控制器程序，控制一个温度过程。已经提供了基本的PID框架，需要补全其中的P、I和D参数以及控制逻辑。

以下是关于PID的解释以及计算提示：

PID控制器（比例-积分-微分控制器）是一种广泛用于工程控制的算法，它通过比较当前状态和目标状态的差异来调整控制输出。PID控制器包括三个部分：

比例（Proportional）： 比例项根据当前误差来调整控制输出。这个项决定了控制器的灵敏度。较大的比例系数会使系统更快地响应，但可能会导致过冲。

积分（Integral）： 积分项考虑了误差随时间的累积，以便处理静态误差。如果系统存在稳态误差，积分项将逐渐减小它。

微分（Derivative）： 微分项考虑了误差变化的速度，以防止系统的超调。微分项可以减小快速变化引起的振荡。

在PID控制中，控制输出计算如下：

Kp、Ki 和 Kd 分别是比例、积分和微分系数。

error 是当前误差（目标值与当前值之差）。

integral 是误差的累积。

derivative 是误差的变化率。

现在，您需要补全以下代码，填写合适的PID参数和控制逻辑，以使温度逐渐接近目标温度 setpoint。

#include <stdio.h>

double Kp = 2.0; // 比例系数

double Ki = 0.1; // 积分系数

double Kd = 0.5; // 微分系数

// 全局变量

double setpoint = 70.0; // 目标温度

double currentTemp = 60.0; // 当前温度

double errorSum = 0.0; // 误差累积

double prevError = 0.0; // 上一次误差

// 补全PID控制函数

double pidControl(double setpoint, double currentTemp) {

// 计算误差

double error = \_\_\_\_\_\_\_\_\_;

// 计算误差累积

errorSum += \_\_\_\_\_;

// 计算误差变化率

double errorChange = \_\_\_\_;

// 计算PID控制输出

double output = \_\_\_\_\_\_\_\_;

// 更新上一次误差

prevError = \_\_\_\_\_;

return output;

}

int main() {

double time = 0.0;

double dt = 0.1;

while (time < 10.0) {

// PID控制逻辑

double controlOutput = pidControl(setpoint, currentTemp);

// 模拟过程

currentTemp += controlOutput \* dt；

printf("Time: %.1f, Temperature: %.2f\n", time, currentTemp);

time += dt;

}

return 0;

}

请使用上面的提示来填写代码中的空白部分，使得温度逐渐接近目标温度 setpoint。