

Can 数据手算:

- Pos 计算: $pos_{out} = \frac{(pos_{in}+95.5) \times 65535}{191}$, 结果保留整数并将 10 进制转化为 16 进制。

说明: 其中 65535 是 16bit 无符号整数的最大值, 这一步将调整后的值放大到 [0, 65535] 的范围内; 95.5 是偏移量, 将-95.5 映射到 0, 此值是单片机内部程序定义的 pos 最大值; 191 是缩放因子, 用于限制输入范围, 由 95.5 乘 2 得来, 确保输入范围 [-95.5, 95.5] 映射到 [0, 65535]。

示例: $pos_{out} = \frac{(2+95.5) \times 65535}{191} = 33,453.73 = 33453 = 0x82AD$, 高 8 位存入 BYTE0, 低 8 位存入 BYTE1。

- Vel 计算: $vel_{out} = \frac{(vel_{in}+45) \times 4095}{90}$, 结果保留整数并将 10 进制转化为 16 进制。

说明: 其中 4095 是 12bit 无符号整数的最大值, 这一步将调整后的值放大到 [0, 4095] 的范围内; 45 是偏移量, 将-45 映射到 0, 此值是单片机内部程序定义的 vel 最大值; 90 是缩放因子, 用于限制输入范围, 由 45 乘 2 得来, 确保输入范围 [-45, 45] 映射到 [0, 4095]。

示例: $vel_{out} = \frac{(1+45) \times 4095}{90} = 2093 = 0x82D$, 高 8 位存入 BYTE2, 低 4 位存入 BYTE3 的高 4 位。

- Kp 计算: $kp_{out} = \frac{kp_{in} \times 4095}{500}$, 结果保留整数并将 10 进制转化为 16 进制。

说明: 其中 4095 是 12bit 无符号整数的最大值, 这一步将调整后的值放大到 [0, 4095] 的范围内; 500 是缩放因子, 确保输入范围 [0, 500] 映射到 [0, 4095]。

示例: $kp_{out} = \frac{50 \times 4095}{500} = 409.5 = 409 = 0x199$, 高 4 位存入 BYTE3 的低 4 位, 低 8 位存入 BYTE4。

- Kd 计算: $kd_{out} = \frac{kd_{in} \times 4095}{5}$, 结果保留整数并将 10 进制转化为 16 进制。

说明: 其中 4095 是 12bit 无符号整数的最大值, 这一步将调整后的值放大到 [0, 4095] 的范围内; 5 是缩放因子, 确保输入范围 [0, 5] 映射到 [0, 4095]。

示例: $kd_{out} = \frac{0.1 \times 4095}{5} = 81.9 = 819 = 0x333$, 高 8 位存入 BYTE5, 低 4 位存入 BYTE6 的高 4 位。

- T 计算: $t_{out} = \frac{(t_{in}+18 \times \text{扭矩常数} \times \text{减速比}) \times 4095}{36 \times \text{扭矩常数} \times \text{减速比}}$, 结果保留整数并将 10 进制转化为 16 进制。

说明: 其中 4095 是 12bit 无符号整数的最大值, 这一步将调整后的值放大到 [0, 4095] 的范围内; 18 是偏移量, 将-18 映射到 0, 此值是单片机内部程序定义的 t 最大值; 36 是缩放因子, 用于限制输入范围, 由 18 乘 2 得来, 确保输入范围 [-18, 18] 映射到 [0, 4095]。