Ecuaciones Matemáticas Utilizadas en el Código para microcontroladores

Se emplean las siguientes ecuaciones matemáticas clave en la implementación del código del Arduino Mega 2560 para la conversión de sensores y el control PID:

Conversión de Lectura ADC a Temperatura (TC)

La lectura analógica de un sensor de temperatura se convierte a grados Celsius (TC) utilizando la siguiente relación lineal. Se asume un ADC de 10 bits (0-1023) y una referencia de voltaje de 5V, con un sensor que produce 10mV por cada grado Celsius.

Valor ADC=analogRead(pin de temperatura)

TC=Valor ADC×1024 bits5000 mV×10 mV1∘C

Simplificando, el factor de conversión es:

1024×105000=102405000≈0.48828125

Por lo tanto, la ecuación en el código es:

TC=analogRead(pin)×0.48828125

Conversión de Lectura ADC a Porcentaje de Nivel (%N)

La lectura analógica de un sensor de nivel se mapea a un porcentaje de llenado (%N) utilizando una función de mapeo lineal. El valor crudo del ADC (0-1023) se escala directamente a un rango de 0 a 100%.

Valor ADC=analogRead(pin de nivel)

La función map() en Arduino realiza internamente la siguiente operación:

%N=(EntradaMax−EntradaMinValor ADC−EntradaMin
)×(SalidaMax−SalidaMin)+SalidaMin

Para este código:

%N=(1023−0analogRead(pin)−0)×(100−0)+0

Simplificando:

%N=1023analogRead(pin)×100

Algoritmo de Control PID (Proporcional-Integral-Derivativo)

El controlador PID se utiliza para calcular la señal de salida (potencia del calentador, de 0 a 255 para PWM) basada en el error entre el setpoint y la temperatura actual.

1. **Cálculo del Tiempo Transcurrido (Δt)**: Es el tiempo en segundos desde la última iteración del PID.

   $$\Delta t = 1000 TiempoActual - TiempoAnterior$$

2. **Cálculo del Error (e(t))**: Es la diferencia entre el valor deseado (setpoint) y el valor medido (input).

   $$e(t) = SetPoint - Input$$

3. **Término Proporcional (Pout)**: Es proporcional al error actual.

   $$Pout = Kp \times e(t)$$

4. **Término Integral (Iout)**: Es proporcional a la suma acumulada de los errores a lo largo del tiempo. cumError representa la integral discreta del error.

   $$cumError = cumError + e(t) \times \Delta t Iout = Ki \times cumError$$

5. **Término Derivativo (Dout)**: Es proporcional a la tasa de cambio del error.

   $$rateError = \Delta te(t) - ErrorAnterior Dout = Kd \times rateError$$

6. **Salida del Controlador (OutputPID)**: La suma de los tres términos da la salida final del PID.

   $$OutputPID = Pout + Iout + Dout$$

   En el código, esto se expresa como:

   $$output = Kp \times error + Ki \times cumError + Kd \times rateError$$

7. **Restricción de la Salida**: La salida calculada se limita al rango permitido para la señal PWM (0-255).

   $$output = constrain(output, 0, 255)$$

Donde:

- Kp: Ganancia Proporcional
- Ki: Ganancia Integral
- Kd: Ganancia Derivativa
- SetPoint: Valor deseado (temperatura objetivo).
- Input: Valor actual medido (temperatura actual).
- ErrorAnterior: Error de la iteración anterior.
- cumError: Suma acumulada de los errores.
- Δt: Tiempo transcurrido entre las lecturas (en segundos).