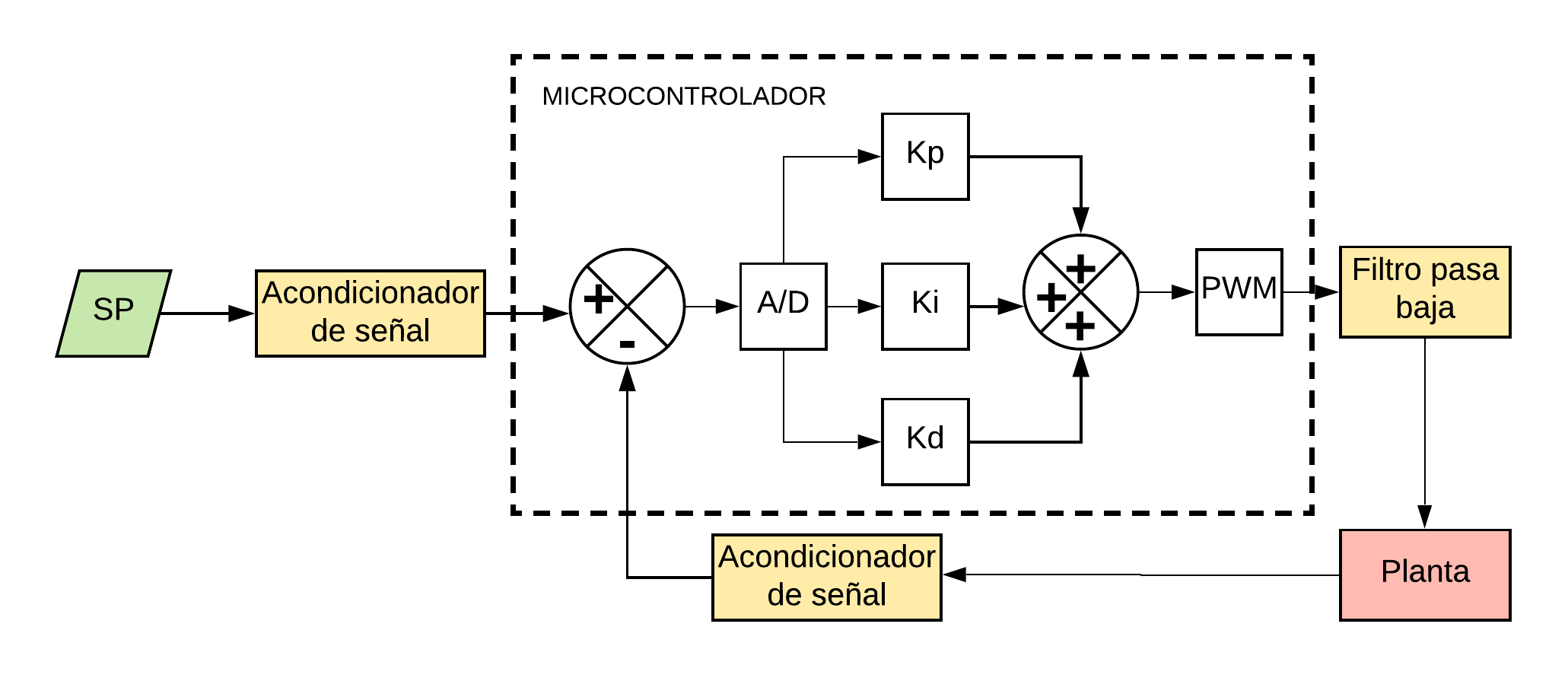
# Controlador PID digital con microcontrolador

El controlador PID se realizado basándonos en el siguiente diagrama de bloques:



## Acondicionamiento de señal (entradas)

Debido a que los voltajes de operación del microcontrolador están comprendidos entre 0 y 5V, es necesario acondicionar las señales provenientes de la planta y del generador de señales. Dichas señales requieren dos tratamientos: reducir su amplitud y adicionar un nivel de offset.

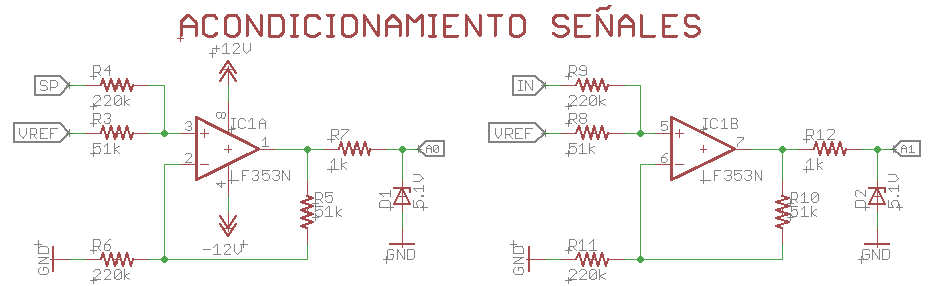
Reducción de amplitud: las señales a trabajar tienen una amplitud que varía desde -10.5V hasta +10.5V y es necesario reducirla a un nivel comprendido entre 0 y 5V para evitar dañar los puertos analógicos del microcontrolador.

Adicionar de offset: debido a que los microcontroladores solo puede trabajar con voltajes unipolares, es necesario añadir un nivel de offset a las señales análogas, con el fin de poder representar los valores de amplitud negativos. Para ello se usara un nivel de offset equivalente a VCC/2, 5V/2 para que el cero de la señal entrante sea representado como 2.5V. Ambos tratamientos se realizan en una única etapa realizada con amplificador operacional LF353N en configuración de sumador no inversor de dos entradas, donde una entrada será la señal de interés y la otra un nivel de voltaje DC con valor de 2.5V para obtener el offset necesario. La señal de entrada debe ser atenuada en factor de:

La ecuación de salida de esta configuración es la siguiente:

Donde debe elegir los valores de resistencias adecuados para obtener el nivel de atenuación deseado para V1. Esta ecuación solo es aplicable si R1 = R4 y R2 = R3, como R2/R3 = 1 y R1/R3 = 0.2381. Con un valor de R2 = 51K entonces:

EL valor comercial más cercano es de 220K y se procede a utilizar este valor. A continuación una imagen del circuito implementado:



Adicionalmente, se usaron diodos zener de 5.1V a la salida de cada operacional, con el fin de dar protección a las entradas análogas del microcontrolador en caso de que por algún motivo se cambiara el voltaje de alimentación de 12V a 15V. De esta forma el máximo voltaje que ingresara a los pines análogos será Vz.

## Microcontrolador

En esta etapa, se utilizara un Arduino mega, que utiliza un microcontrolador Atmega2560-AU de montaje superficial. Se eligió esta placa, debido a que ya se contaba con ella y se requería reducir costos en el desarrollo del proyecto, pero cabe destacar que esta sobredimensionada para las tareas que debe realizar. Las características de hardware de este microcontrolador son:

|  |  |
| --- | --- |
| Microcontrolador | ATmega2560 |
| Voltaje operativo | 5V |
| Voltaje de entrada (recomendado) | 7-12V |
| Voltaje de entrada (limite) | 6-20V |
| Pines Digitales E/S | 54 (15 proporcionan salida PWM) |
| Pines Digitales E/S PWM | 15 |
| Entradas Analógicas | 16 |
| Corriente DC por pin E/S | 20 mA |
| Corriente DC para el pin de 3.3V | 50 mA |
| Memoria Flash | 256 KB (ATmega2560) 8 KB bootloader |
| Memoria SRAM | 8 KB (ATmega328P) |
| Memoria EEPROM | 4 KB (ATmega328P) |
| Velocidad de reloj | 16 MHz |
| Pin LED incluido | 13 |
| Largo | 101.52 mm |
| Ancho | 53.3 mm |
| Peso | 37 g |

Esta placa se programó para que lea las entradas análogas correspondientes a la entrada (salida de la planta a controlar) y el setpoint (salida del generador de señales), donde posteriormente serán convertidas de análogo a digital por medio del ADC interno de 10 bits del MCU. Este ADC nos entregara valores binarios entre 0 y 1023 para lectura de voltaje comprendidas entre 0 y 5V. Cabe destacar que como las salidas de la parte analógica de la planta oscilan entre valores desde -10.5V hasta +10.5V, es necesario realizar una etapa de acondicionamiento de señal, que permita al Arduino Mega realizar las lecturas de forma correcta. Su segunda función es controlar una pantalla LCD1602 para visualizar los valores de las ganancias Kp, Ki y Kd del controlador que se esté implementando.

Con el acondicionamiento de señal de la etapa anterior, el MCU leerá en sus pines analógicos un 0 cuando la señal de entrada sea -10.5V y 1024 cuando la entrada sea de +10.5V. Con esto en mente, el primer paso en el código es realizar una conversión que nos entregue valores de coma flotante entre -10.5 y 10.5 que son los valores reales de la planta. Esto se realiza con la siguiente línea de código:

consigna = (analogRead(SETPOINT) \* 0.02053) - 10.42;

entrada = (analogRead(ENTRADA) \* 0.02053) - 10.42;

donde las variables consigna y entrada son de tipo double (decimales de mayor precisión que los float) y el valor de 0.02053 es el resultado de dividir el voltaje pico a pico de la señal entre la resolución máxima del conversor análogo a digital, que en este caso al ser de 10 bits, su valor es de 1023.

Posteriormente, hacemos uso de la librería Arduino PID para realizar los cálculos necesarios para la señal de control. Esta librería permite establecer límites de saturación en los valores de entrada y salida, y también recibir como parámetros las ganancias Kp, Ki y Kd. De esta forma podemos realizar el control digital con los cálculos realizados anteriormente para el controlador análogo y obtener resultados similares. La señal de control será del tipo PWM, con un valor promedio entre 0 y 5V proporcional al porcentaje del ciclo de trabajo, por ejemplo, para un ciclo de trabajo del 50%, la señal de salida tendrá un voltaje promedio de 2.5 Voltios. Como nuestras plantas requieren una señal de control continua que este comprendida entre -10.5V y +10.5V, es necesario acondicionarla para poder aplicarla a la entrada de la planta. En el código de programación se realiza esta acción con la siguiente línea:

sal\_pwm = OFFSET + int(salida \* 10.41);

Donde el valor de OFFSET es de 125 y la variable salida es un valor entero entre -12 y 12, dependiendo de los cálculos realizados por el PID. El valor de 10.41 (255/24) corresponde al factor de conversión para obtener una señal digital con resolución 8 bits, que son las señales de tipo PWM que puede manejar el Arduino mega. Esta acción se realiza con la siguiente línea de código:

analogWrite(U\_CONTROL, sal\_pwm);

Dependiendo del valor que tome la variable sal\_pwm, se obtendrá una señal periódica con un ciclo de trabajo variable entre 0 y 100%, obteniendo un voltaje promedio entre 0 y 5V.

## Acondicionamiento de señal (salida)

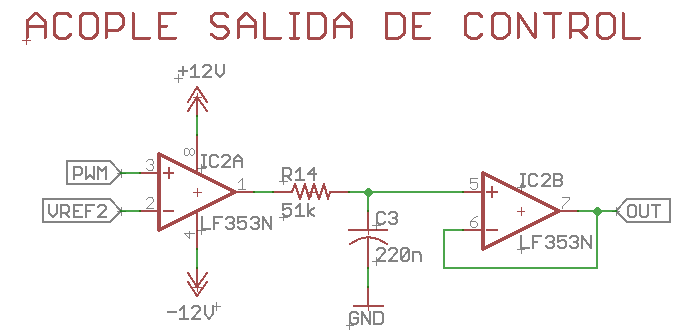
Para poder inyectar la señal de control a la planta física, es necesario realizar un acondicionamiento para convertir la señal cuadrada de 0 a 5V, en una señal continua que oscile entre -10.5V hasta +10.5V. esto se realizó mediante tres etapas, un comparador con amplificador operacional, un filtro pasivo pasa bajas y un amplificador seguidor de emisor, más conocido como buffer.

Comparador: esta etapa me permite obtener a su salida una señal cuadra con la misma frecuencia que la proporciona por el Arduino, pero con una amplitud que oscila entre -10.5V hasta +10.5V, que son los voltajes de saturación del LF353N cuando se alimenta con una fuente dual de +-12V.

Filtro pasa bajas: luego de acondicionar la amplitud de la señal, es necesario convertir a un nivel de DC que oscile entre -10.5V y +10.5V y que sea proporcional al ancho de pulso de la señal PWM entregada por la etapa del comparador. Con un filtro pasivo RC de primer orden podemos filtrar toda componente de frecuencia realizando la integral del área bajo la curva de la señal PWM, siempre que la frecuencia de corte del filtro sea por lo menos unas 10 veces menor a la de la señal PWM. Realizando una manipulación de los registros del Timer1 del Arduino mega, podemos obtener una frecuencia de señal PWM de 3.9 KHz. Con este valor y un capacitor de 220nF, procedemos a calcular un valor de resistencia que nos proporcione una frecuencia de corte adecuada:

Para mejorar el nivel de filtrado, se procede a usar una resistencia de 51K y de esta forma por der usar frecuencias más bajas en la señal de PWM en caso de ser necesario.

Buffer: esta etapa tiene la función de realizar un acople de impedancias entre la salida del filtro y la entrada a la planta, con el fin de obtener una respuesta lo más fiel posible a los cálculos realizados y evitar tener respuestas diferentes y cálculos erróneos en el algoritmo del PID. El tiempo de muestreo para el algoritmo del PID se calculó siguiendo la regla de usar un Tm menor a 10 veces el tiempo de estabilidad de la planta en lazo abierto. A continuación se muestra una imagen del circuito implementado en esta etapa:



El esquemático final y el diseño de la PCB se muestran a continuacion:

