Trabajo Integrador
Electrónica General y Aplicada

Anemómetro Programable

Grupo 2:

- *Aspee Juan Manuel - 13058 - Industrial*
- *Babolene Ignacio - 13242 - Mecatrónica*
- *Bonada Ulises - 12873 - Mecatrónica*
- *Corazza Luis Agustin - 12909 - Mecatrónica*
- *Gaviño Matías - 13019 - Mecatrónica*

Descripción del proyecto realizado:

El trabajo consiste en realizar un anemómetro de copas, que en presencia del viento produce pulsos que son enviados al microcontrolador, en este caso a un Arduino UNO. Por otra parte, el anemómetro cuenta con un sensor de posición angular absoluto, que permite determinar la dirección (sentido) del viento. Además, a través del puerto serie, un supervisor podrá comunicarse con el microcontrolador y establecer o consultar la velocidad máxima permitida del viento. En caso que la velocidad del viento supere la velocidad máxima permitida sonará una alarma.

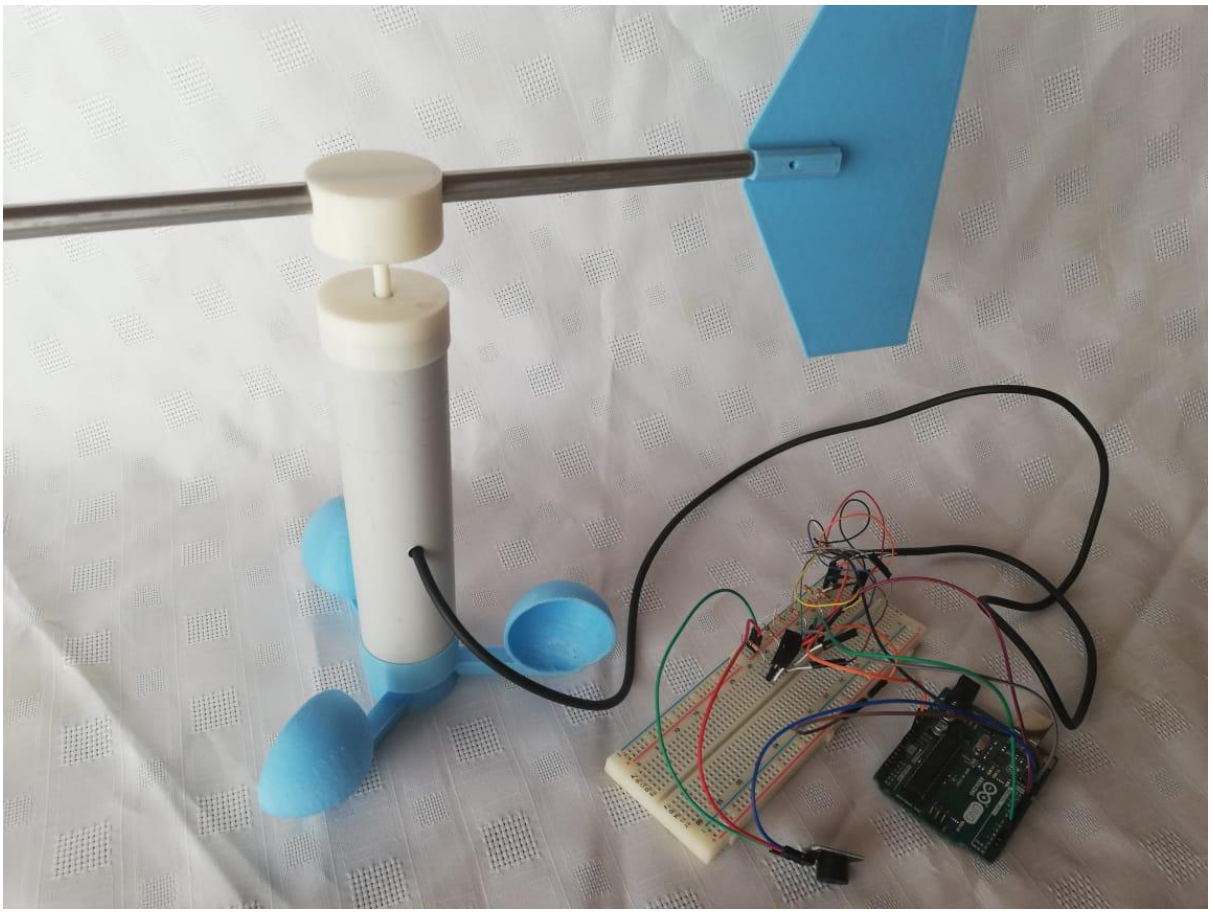


Figura 1: Anemómetro realizado

Para poder medir la velocidad del viento hemos utilizado un optoacoplador ranurado (TCST 2103) mientras que para medir la dirección del viento hemos utilizado 3 sensores infrarrojos (CNY70).

Figura 2: Optoacoplador Ranurado TCST 2103

Figura 3: Sensor Infrarrojo CNY70

Por otro lado, para realizar la alarma hemos utilizado un buzzer de 5V, el cual estará conectado a una salida digital de Arduino que pueda funcionar con PWM, en este caso hemos conectado el buzzer al pin digital 9.

Figura 4: Buzzer 5V

A continuación se muestra el circuito esquemático del proyecto completo:

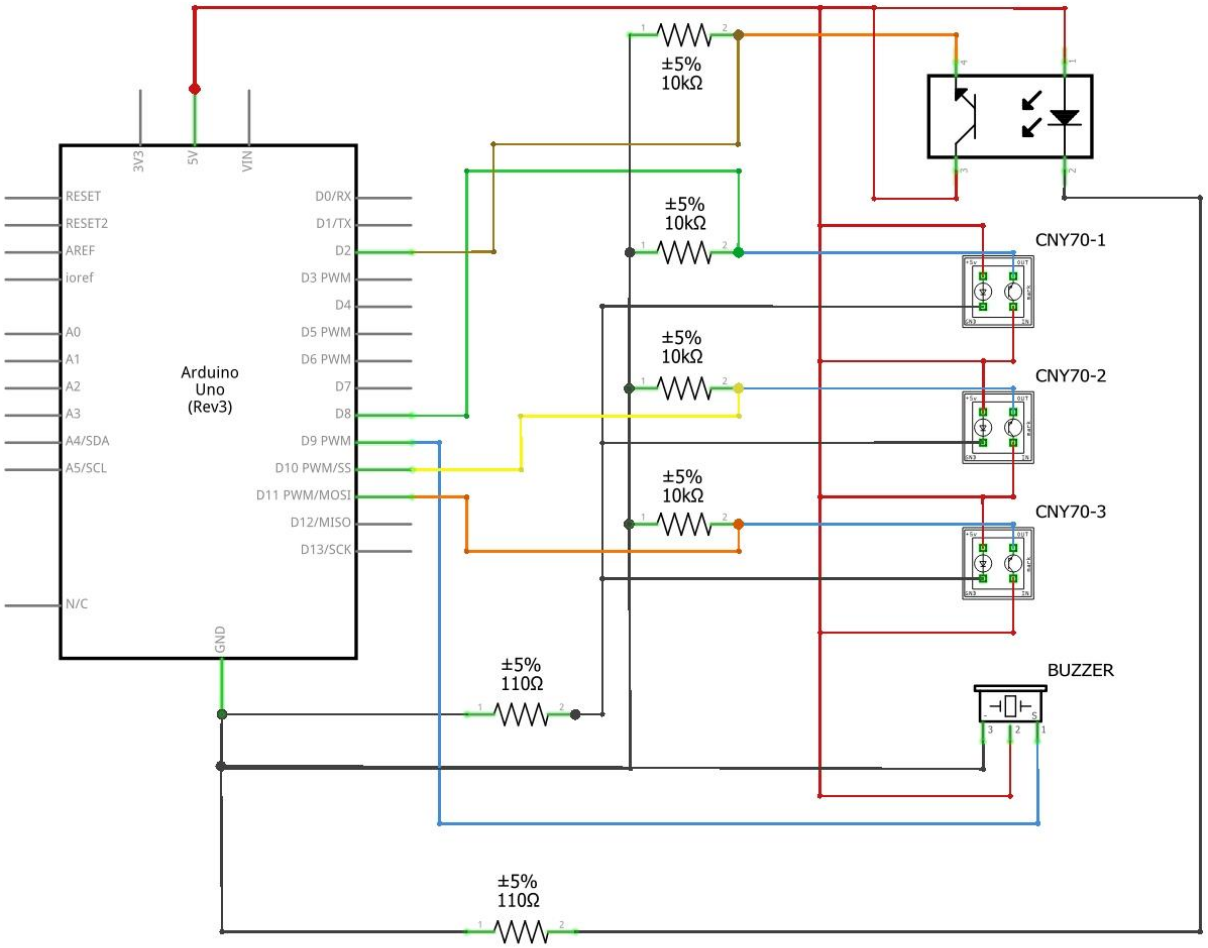


Figura 5: Circuito Esquemático

A su vez, cabe mencionar que las diferentes piezas que conforman el mecanismo del anemómetro han sido diseñadas en SOLIDWORKS y luego

fabricadas en impresora 3D. Además, como estructura de soporte de todas las piezas se ha utilizado un tubo de PVC de 40 mm de diámetro interior.

Descripción del funcionamiento del anemómetro:

Como mencionamos anteriormente, el anemómetro está compuesto por 4 sensores, 3 de estos están destinados a determinar la dirección del viento, y el otro restante lo usamos para calcular la velocidad. A continuación explicaremos cómo a través de estos sensores podemos medir la velocidad y la dirección del viento.

- Velocidad:

En primer lugar, en uno de los extremos del anemómetro colocamos una serie de copas, las cuales serán empujadas continuamente por el viento y transmitirán este movimiento de giro hacia un eje central sobre el cual se encontrará acoplado un disco insertado en la ranura central del sensor TCST 2103. Dicho disco es en su mayoría hueco, sin embargo tiene 4 rayos los cuales servirán para que el sensor produzca pulsos y los envíe al Arduino.

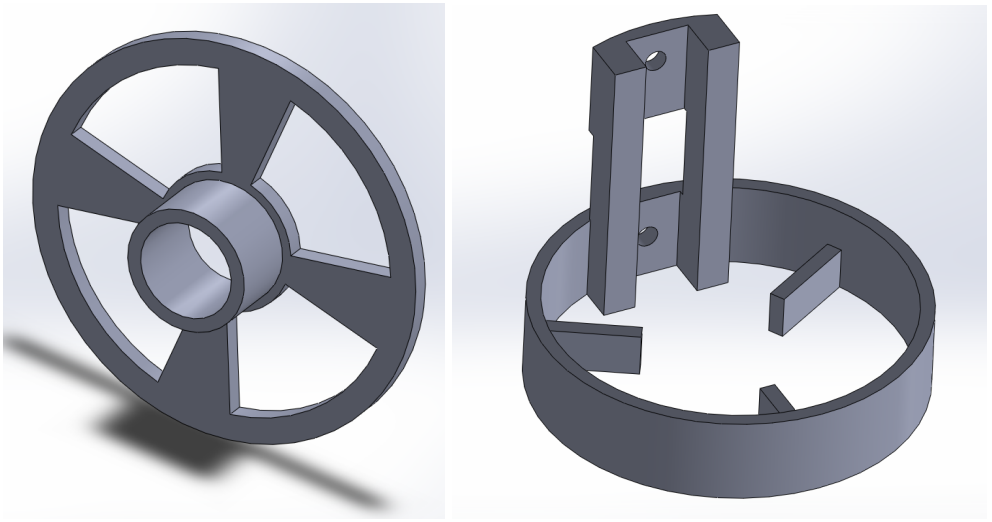


Figura 6: Disco ranurado y soporte optoacoplador TCST 2103

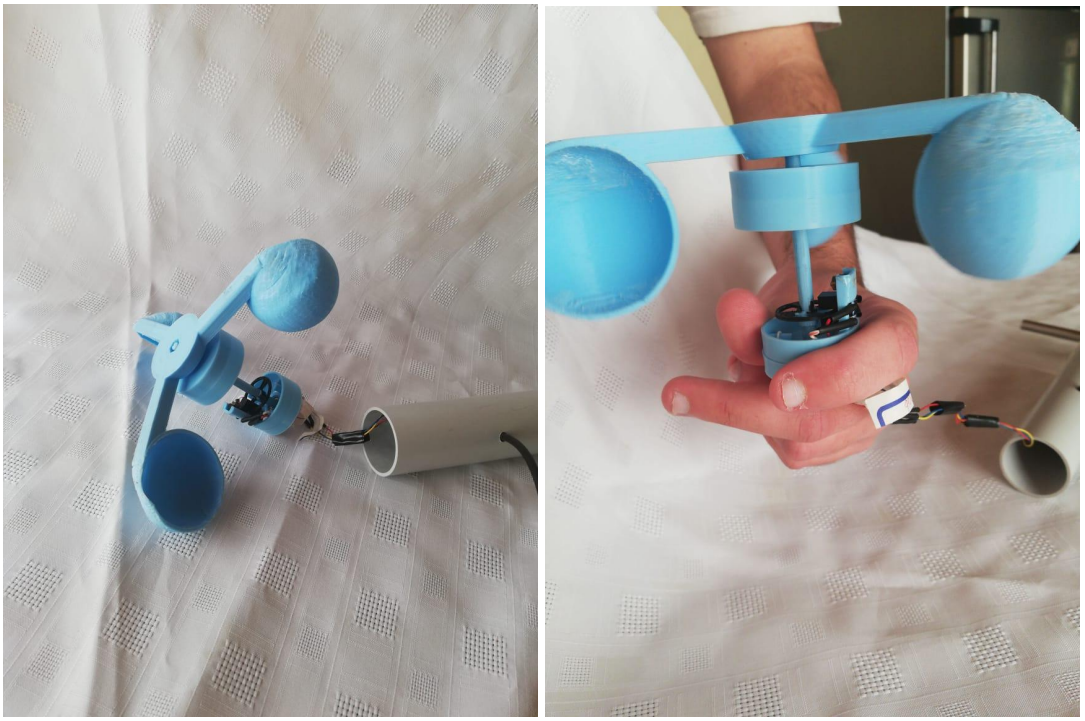


Figura 7: Copas del Anemómetro y Montaje

A medida que los pulsos van llegando al Arduino, este comienza a contar la cantidad de pulsos que llegaron en un tiempo de muestreo **T**, que por defecto está configurado en 5 segundos. Luego, una vez que transcurre este tiempo de muestreo, calculamos las revoluciones por minuto (RPM) del anemómetro, teniendo en cuenta que 1 revolución = 4 pulsos, ya que el disco que mencionamos anteriormente tiene 4 rayos.

Luego, transformamos las RPM a km/h a través de una recta de regresión

$$V = b * RPM$$

donde **b** es una constante cuyo valor fue determinado experimentalmente a través de una calibración de velocidades.

- Dirección del viento:

En primer lugar, para medir la dirección del viento utilizamos una veleta plana la cual, al ser golpeada por el viento, tiende a orientarse paralelamente al mismo, y a su vez, haciendo rotar un eje que se encuentra en el interior de la estructura del anemómetro.

Sin embargo, para medir la dirección del viento necesitábamos poder determinar la **posición absoluta** en la que se encuentra apuntando el viento al que está sometida la veleta, por lo que decidimos utilizar el **código rotatorio de Gray**, que permite transformar la posición angular de un eje en un código digital de bits. En la siguiente figura se muestra un disco con un código de Gray de 3 bits:

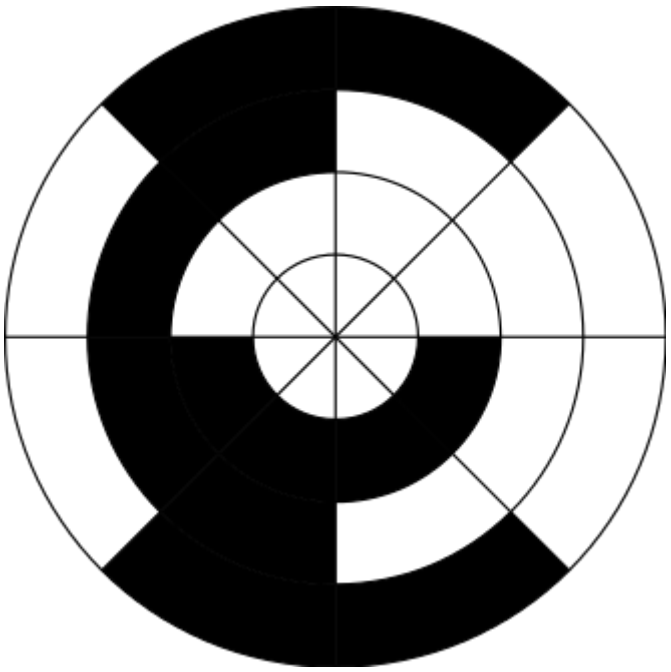


Figura 8: Disco con código de Gray de 3 bits

Sector	Sensor 1	Sensor 2	Sensor 3	Ángulo
1	1	1	1	0° a 45°
2	1	1	0	45° a 90°
3	1	0	0	90° a 135°
4	1	0	1	135° a 180°
5	0	0	1	180° a 225°
6	0	0	0	225° a 270°
7	0	1	0	270° a 315°
8	0	1	1	315° a 360°

De esta forma, dependiendo del valor que tomen los 3 sensores CNY podemos determinar la dirección del viento con una apreciación de 45°.

Cabe destacar que para la construcción de dicho código de Gray hemos dividido cada una de las pistas del disco de la figura anterior, donde van colocados los 3 sensores, en **3 discos independientes de menor diámetro**, luego dichos discos se ubicaron adecuadamente para mantener el código rotatorio. De esta manera, se logró reducir considerablemente el tamaño radial del anemómetro.

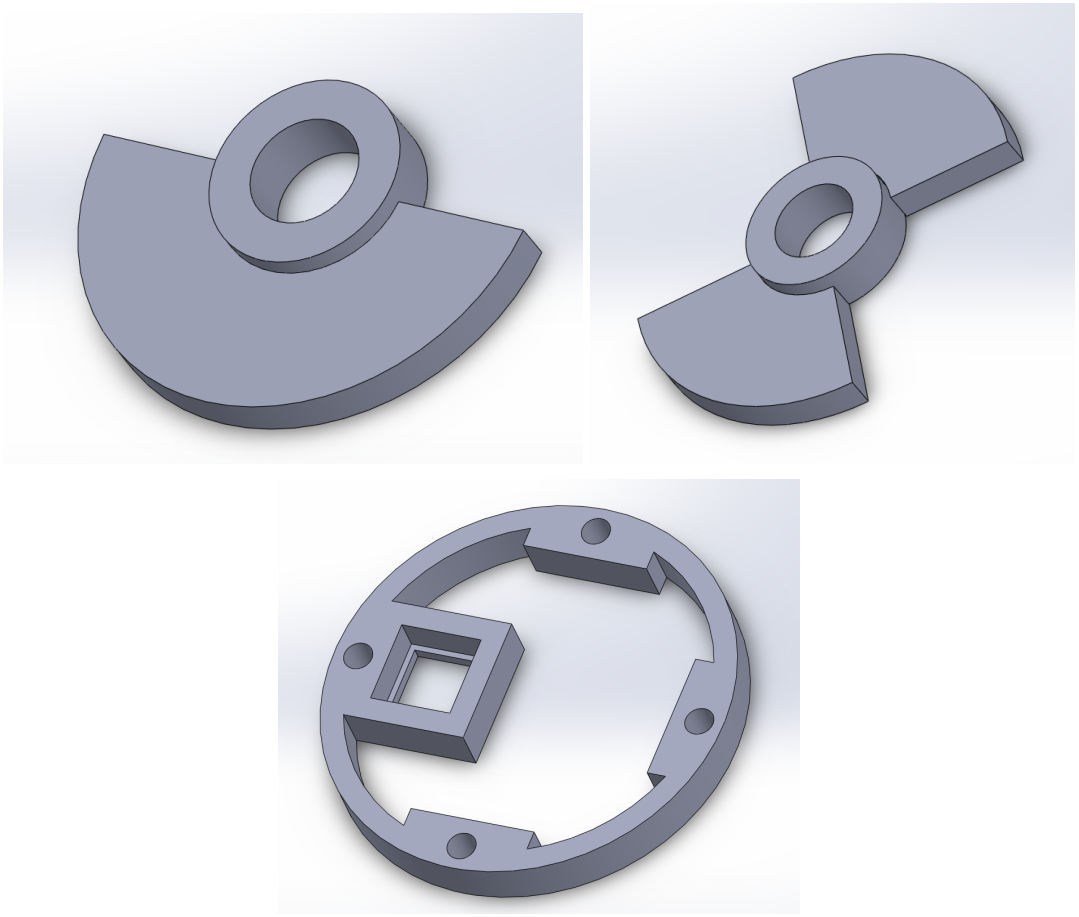


Figura 9: Discos individuales y soporte para los sensores CNY70.

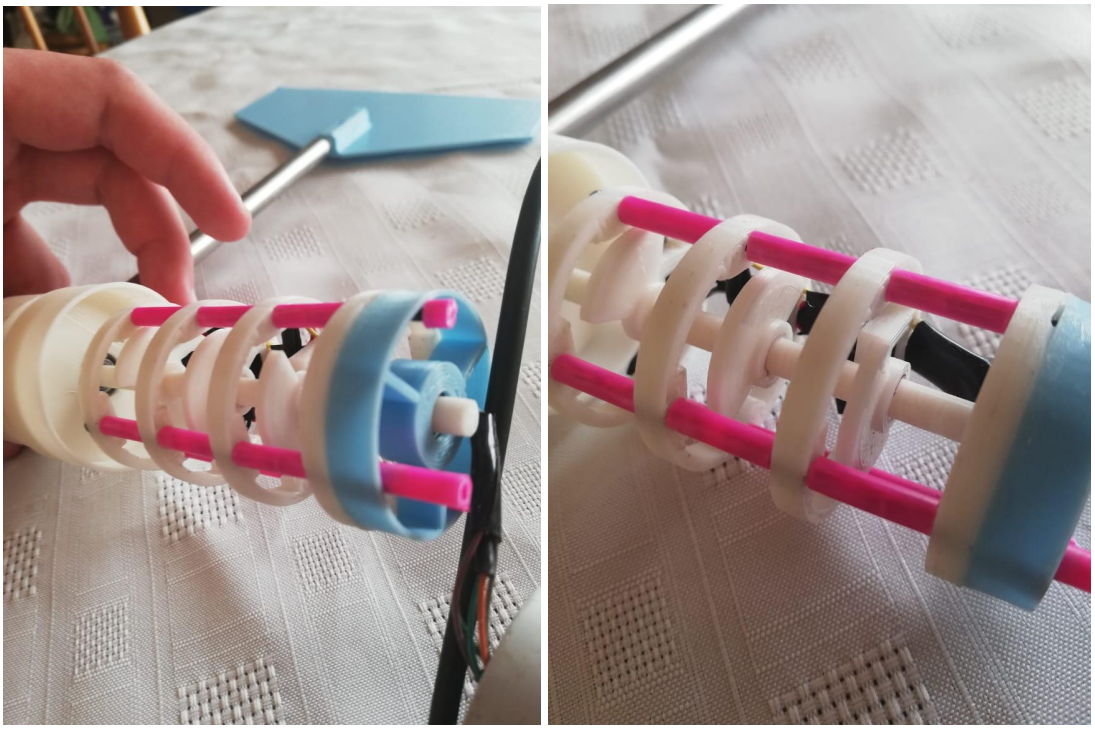


Figura 10: Montaje disco con código de Gray dividido en 3 niveles.

Descripción de la comunicación PC - Arduino:

Para establecer la comunicación entre el supervisor (PC) y el microcontrolador (Arduino) conectamos el puerto serie de la PC (puerto COM) y el puerto serie del Arduino (interfaz UART). Además, para que ambas partes pudieran procesar correctamente los mensajes, decidimos implementar un protocolo de comunicación simple. Por ejemplo, para que el supervisor pueda interactuar con el microcontrolador debe enviar consultas o comandos que constan de la siguiente estructura:

INICIO	COMANDO	DATOS (OPCIONAL)	FIN

Como **carácter de inicio** se ha utilizado el carácter ":" (código ASCII 58) y como *carácter de final* se utiliza el carácter no imprimible "<CR>", el cual se escribe cuando presionamos la letra ENTER del teclado. Luego, los **carácteres de comando** son "V" y "C".

El **comando V** permite modificar la velocidad máxima permitida, por lo que dicho carácter de comando deberá estar seguido de la nueva velocidad máxima, por ejemplo ":V80<CR>".

Por otro lado, el **comando C** permite consultar la velocidad máxima permitida y dicho carácter de comando no debe estar seguido por ningún carácter de dato. Por ejemplo, si la velocidad máxima es de 30km/h, al ingresar el comando ":C<CR>", el Arduino devolverá el mensaje ":C30<CR>", indicando que, efectivamente, la velocidad máxima es 30km/h.

A su vez, cabe destacar que el Arduino envía a la PC los datos de velocidad en km/h, RPM y dirección del viento cada 5 segundos. Dicho mensaje se envía siguiendo el siguiente formato:

INICIO	VELOCIDAD	SEPARADOR	RPM	SEPARADOR	DIRECCIÓN	FIN

donde, al igual que antes, los carácteres de inicio y fin son ":" y "<CR>" respectivamente y el **carácter de separación** es "-". Así por ejemplo, si el Arduino detecta una velocidad de 10 km/h, 35 RPM y dirección norte, el mensaje que enviará será ":10-35-Norte<CR>"

Proceso de calibración para la conversión de RPM a km/h:

Puesto que el optoacoplador que hemos utilizado envía simplemente pulsos al Arduino, los cuales pueden ser convertidos fácilmente a revoluciones por minuto, tuvimos que calibrar el anemómetro y asociar las distintas RPM a ciertas velocidades en km/h.

Para ello, un día que no hubo demasiado viento, subimos el anemómetro a un auto, lo aceleramos a diferentes velocidades y fuimos midiendo cuantas RPM teníamos para cada una de las velocidades. Los datos obtenidos fueron:

Velocidad (km/h)	RPM
0	0
15	126
20	189
25	240
30	348
35	447

Con estos datos, armamos la recta de regresión **V = a + b * RPM**, donde resultó que **a=0** y **b=0.0762108262**. Además, obtuvimos que los datos obtenidos se corresponden en una linealidad del 98%, con lo cual **el cálculo de las velocidades a través de dicha recta de regresión es aceptable**.

Diagrama de Bloques:

A continuación se muestra el diagrama de bloques del proyecto, las entradas al microcontrolador son los pulsos que envía el optoacoplador TCST 2103 y el estado de los 3 sensores CNY70. Además, el supervisor puede ingresar algún comando al Arduino.

Por otro lado, el Arduino envía periódicamente los datos de velocidad, RPM y dirección al supervisor y puede activar o desactivar una alarma.

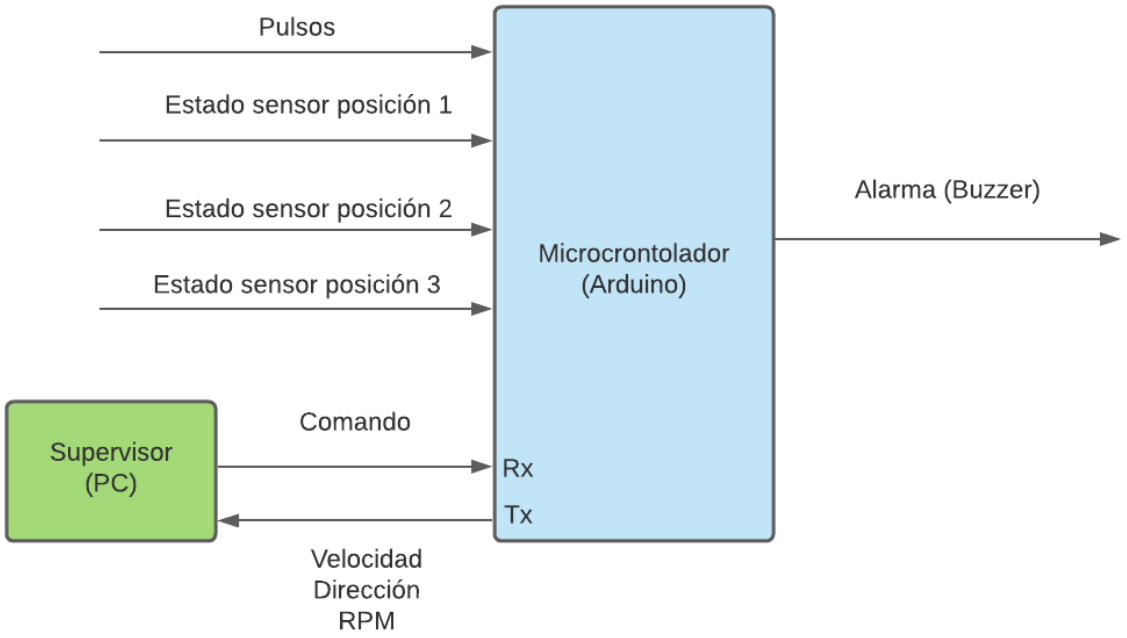


Diagrama de Flujo:

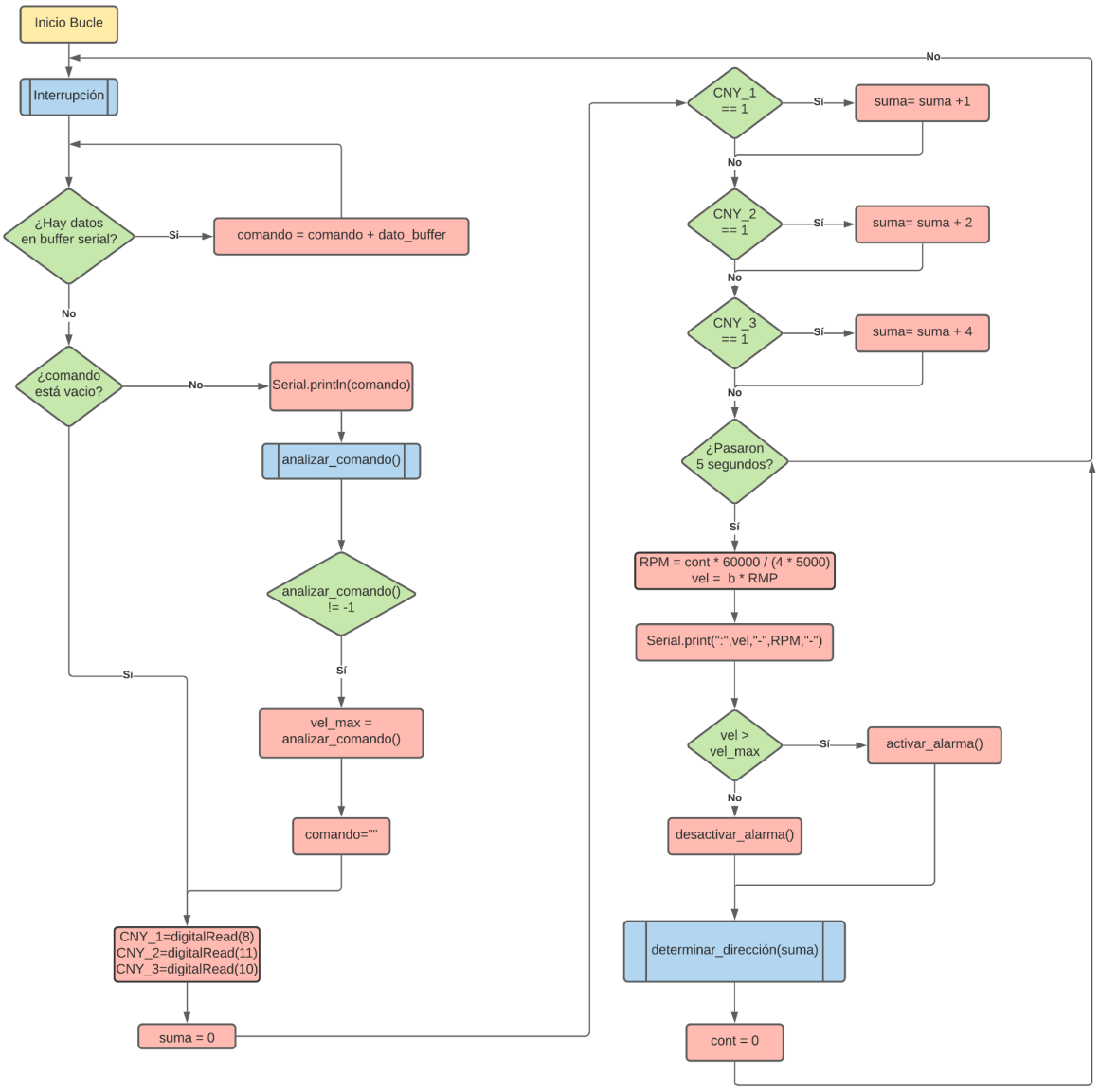
En las próximas figuras se muestran los diagramas de flujo de las diferentes partes que conforman la lógica del proyecto.

Para el funcionamiento del anemómetro hemos planteado una interrupción en el Pin Digital 2, el cual se encuentra conectado al optoacoplador TCST 2103. Dicha interrupción se ejecutará cada vez que detecte un flanco de subida en dicho pin y simplemente sumará uno a la cantidad de pulsos que ha recibido el microcontrolador durante los últimos 5 segundos, además de encender o apagar el LED asociado al Pin Digital 13 del Arduino.

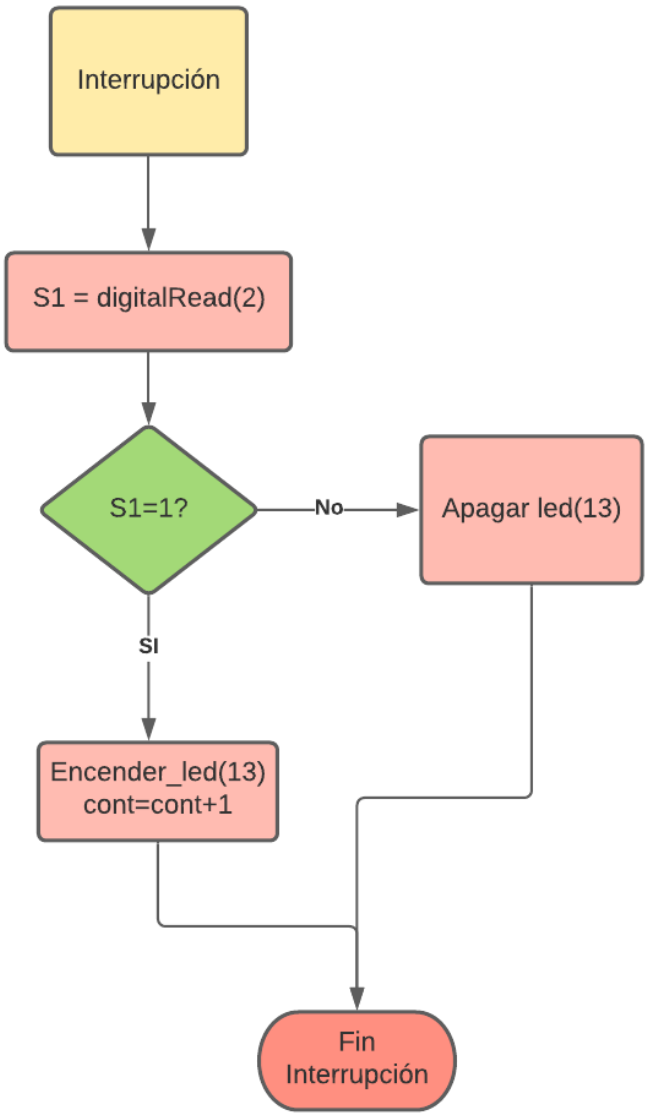
Por otro lado, la función ***analizar_comando()*** sirve para analizar el mensaje que el Arduino recibe por el puerto serie y determina la instrucción que debe ejecutarse en caso de que el mensaje sea un comando válido.

Por último, la función ***determinar_dirección()*** simplemente analiza el valor de la variable suma y envía a través del puerto serie la dirección del viento. Para su implementación en Arduino se ha utilizado una sentencia **switch… case...**

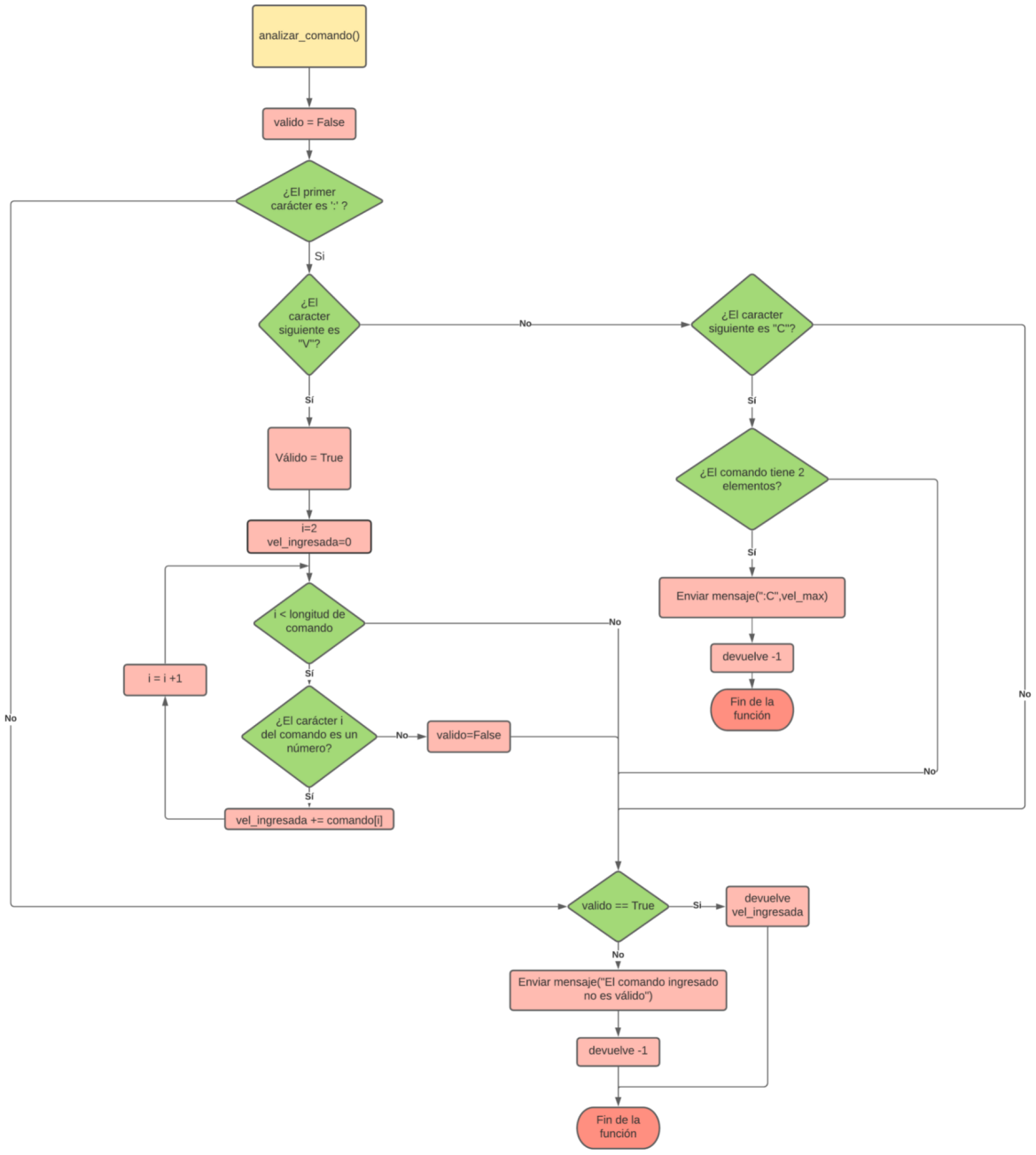
- Diagrama de flujo del Bucle Principal (void loop):



- Diagrama de flujo de la Interrupción en el Pin Digital 2:



- Diagrama de flujo de la función analizar_comando():



- Diagrama de flujo de la función determinar_dirección():