C++ units library research form

The goal of the research for a C++ units library is to have a standard units library for Inspiro which will be included in the CSDD-examples, and thus into future projects.
A **very** in depth explanation by an author of such a library can be found here: https://www.youtube.com/watch?v=pPSdmrmMdjY.

Some examples of code found in different libraries (this means they use different syntaxes) can be found below as the first question.
This was due to hitting a word limit in the description for this form...

This form was made in order to create an understanding of which units of measurement are currently used within Inspiro BV.
Knowing which units are used means the library can be better adapted to Inspiro's needs.

Your responses will be compiled into a table that will be included and referenced in the research paper.
Please note that this table (including your responses) will be visible to everyone within Inspiro.

I'd encourage you to answer the questions to the best of your knowledge, since your answers will affect the usability of the library.
If, for example, you use the quantities "length" and "electrical current" often in your work, but you forget to add the "length" unit in your answer, there is a chance that length won't be included in the library.

Lastly, all responses must be provided in English.

If there are any questions regarding the library research or regarding this form, please contact me at lars.van.duijnhoven@inspiro.nl.
Furthermore, thank you for taking the time to fill out this form!

* Required

* This form will record your name, please fill your name.

1. **Examples of library code:**

   Example of general code
   - Quantity<Meters, uint8_t> distance = meters(5); // Creates a distance of 5 meters
     std::cout << "Distance in meters: " << (distance + (centi(meters))(100)).as<uint8_t>(meters) << std::endl; // Prints a value of 6 m

   Example of automatic conversions
   - static_assert(180* km / (2 * h) == 25 * m / s); // Succeeds because of automatic conversions

   Example of temperature, since this has to include affine spaces. See this link for more information: https://en.wikipedia.org/wiki/Dimensional_analysis#Affine_quantities.
   - quantity_point temp = point<deg_c>(20.); // The dot makes it a float
     std::println("Temperature: {} ({})", temp.quantity_from_zero(), temp.in(deg_F).quantity_from_zero()); // Outputs "Temperature: 20 °C (68 °F)"

   Example of typesafety
   - Quantity<Seconds, uint64_t> timeSinceStart = seconds(0);// Initialize time variable
     timeSinceStart += seconds(2)+ minutes(3)+ hours(1);// Outputs 3782 seconds
     timeSinceStart = seconds(2)+ meters(3);// Gives a compiler error

   Example of declaring your own types
   - using MetersPerSecondSquared = decltype(Meters{} / squared(Seconds{})); // Declare your own type for usage below
     Quantity<MetersPerSecondSquared, uint8_t> acceleration = (meters / squared(second))(2); // Use your own compiletime created type!

   No answer is required for this question

   Enter your answer

2. Are there any problems you have had with unit conversions or unit mismatches? If so, please provide examples.
   Example of an astronomically big problem: https://en.wikipedia.org/wiki/Mars_Climate_Orbiter#Cause_of_failure.
   Another example: It was believed that the integer "timeSinceStart" was in seconds, but after having the program crash multiple times it appeared to be in ms. *

3. Do you believe that a library would have helped solving (or maybe solved entirely) the previously named problems? Please provide an explanation with your answer. *

4. Are there any problems you foresee that might be had when using a library for the units?
   Example: It's uncertain if code **with** the library types could conflict with code **without** the library types due to some form of mismatch between the types. *

5. What quantities and units do you currently use in your work?
   Example: "Temperature" is a quantity and "Kelvin" is a unit that belongs to the quantity "Temperature".
   Combinations such as meter per second also count, it does not have to be a single unit.

   The following links can be used as guidance for your answers: https://en.wikipedia.org/wiki/International_System_of_Units and https://en.wikipedia.org/wiki/List_of_physical_quantities.
   *