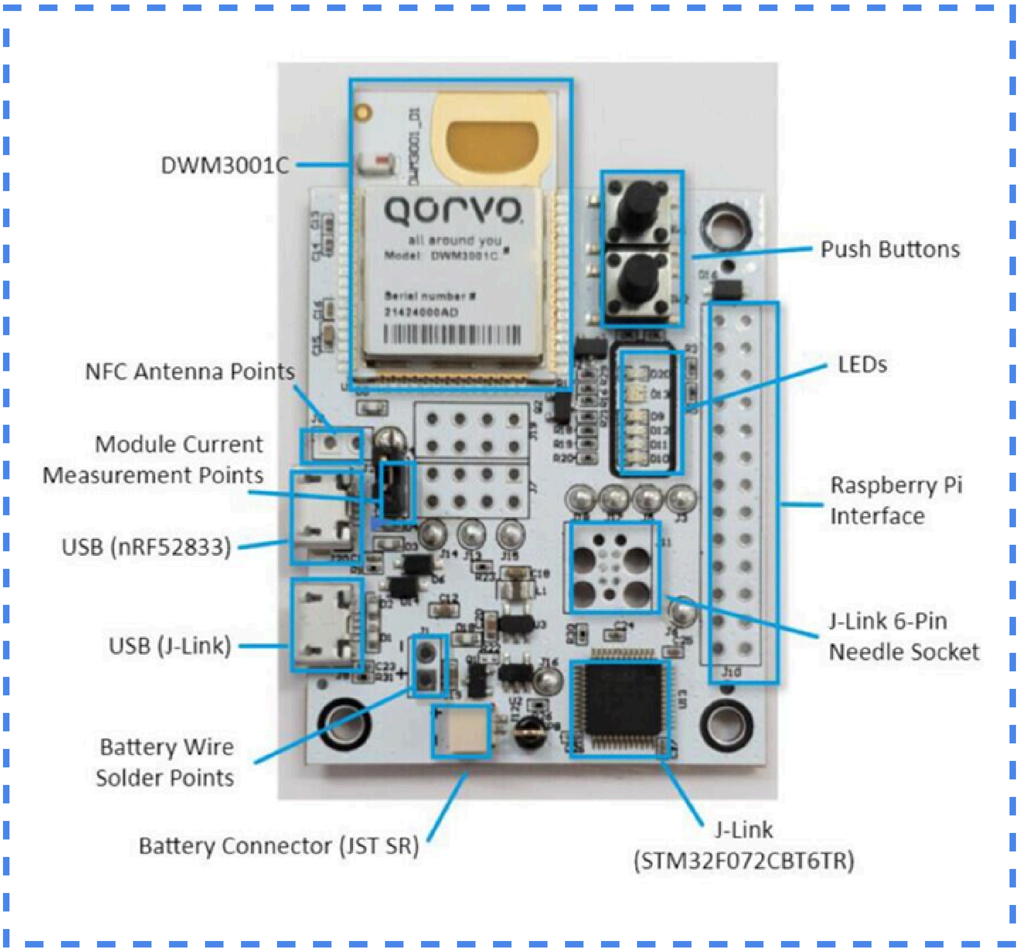
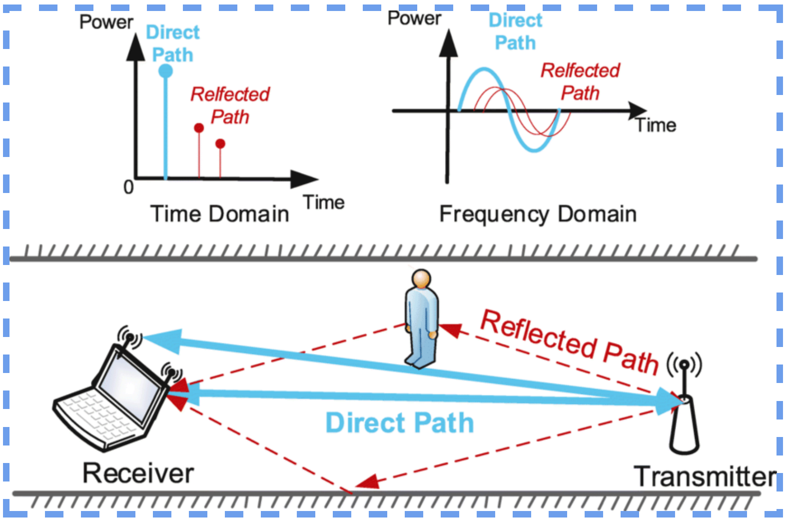


UWB card : Qorvo Decawave DWM3001CDK

Glossary :

- **multipath:** Multipath refers to the phenomenon where a radio signal does not arrive via a single path, but through multiple different paths due to reflections.



UltraWideBand (UWB)

Link : https://fr.mathworks.com/discovery/ultra-wideband.html

Ultra-Wideband relies on time-of-flight measurements of radio signals with a very large bandwidth. This approach enables highly accurate distance estimation (down to the centimeter level), with low sensitivity to multipath effects and signal power variations.

Why is UWB weakly sensitive to multipath?

UWB transmits extremely short pulses (≈ nanoseconds).

This allows the receiver to separate signal paths in time:

the first received peak corresponds to the direct path

the following peaks correspond to reflections (multipath)

The UWB receiver can therefore identify the first path and ignore reflected paths. Unlike Wi-Fi or Bluetooth, which transmit **continuous waves**, UWB sends very short pulses (on the order of a nanosecond).

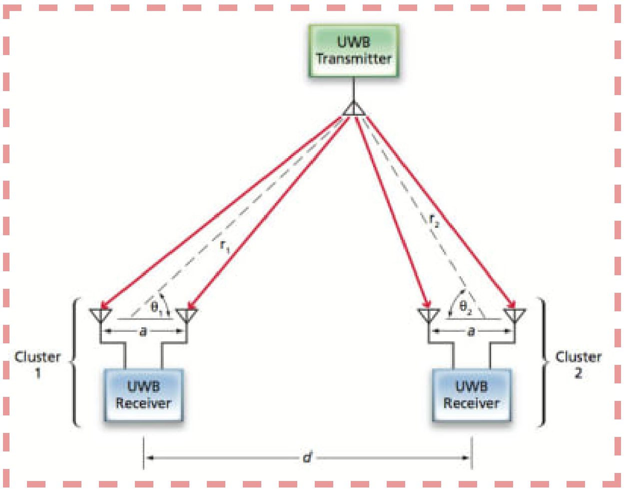
Distance measurement

Example

- Device **A** sends a UWB pulse
- Device **B** receives it and sends a response
- Device **A** measures the total elapsed time
- Distance = (time × speed of light / 2

Because the pulses are **ultra-short, extremely precise timing measurements** are possible → **very accurate distance estimation**.

Orientation / Localization Measurement

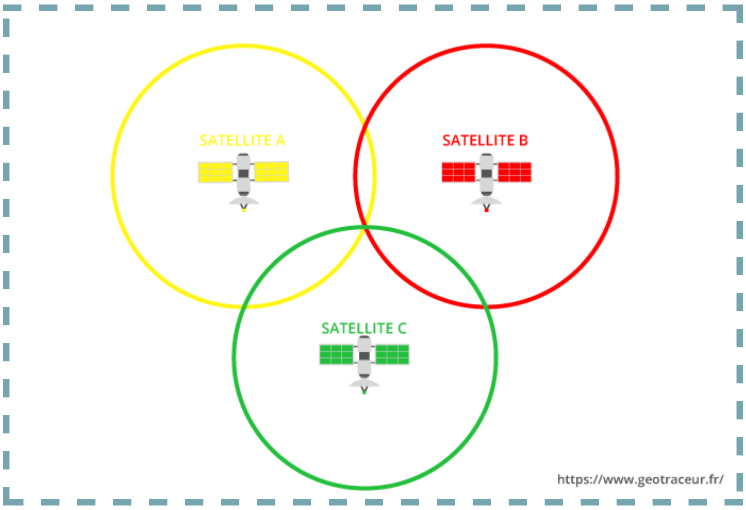


UWB does not rely on a single antenna but on multiple spaced antennas, and measures tiny time or phase differences between them.

Since the technology is based on very precise time measurements (on the order of nanoseconds), this enables **Angle of Arrival (AoA)** estimation.

The DWM3000EVB, based on the DWM3000 module / DW3110 chip, has only one antenna in its standard configuration. Therefore, it cannot measure **AoA or PDoA**, and cannot determine receiver orientation using UWB alone.

However, by using multiple anchors, it is possible to perform **trilateration**. By knowing the distance between the receiver and three transmitters, its position can be estimated.

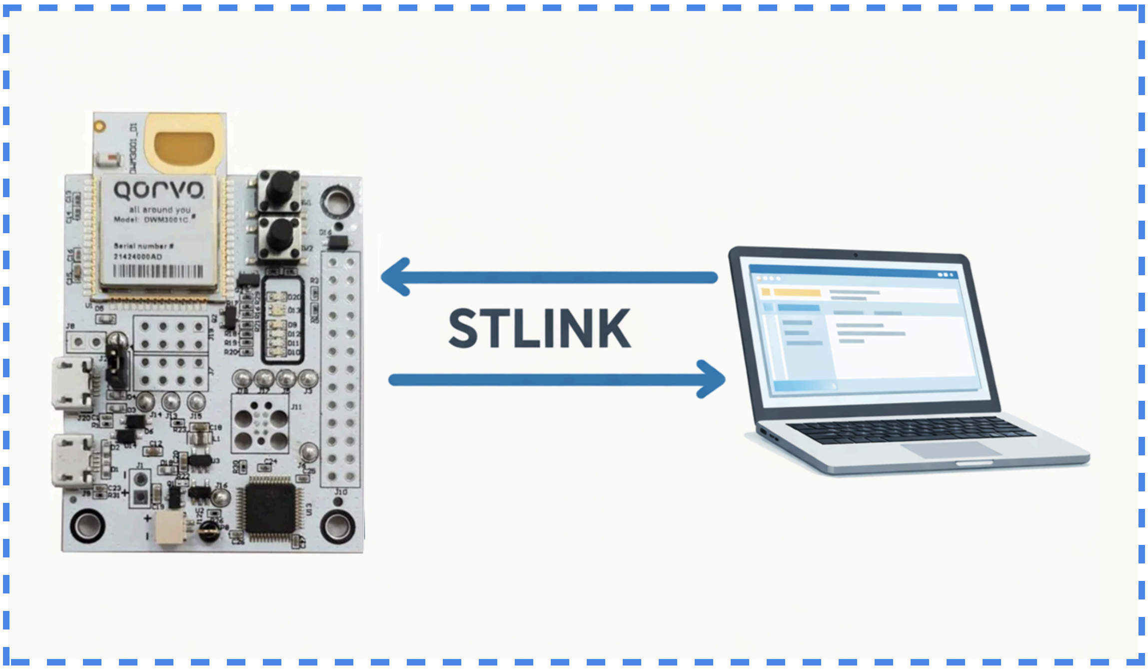


We will ultimately use the **DWM3001CDK (Ultra-Wideband Module Development Kit)**. This module **also does not support AoA**.

On the other hand, UWB involves higher hardware complexity, higher cost compared to Bluetooth solutions, and partial integration within the mobile ecosystem. These constraints raise questions about its suitability for mass-market solutions or cost-constrained applications.

Feature	Details / Values
Frequency band	Wide spectrum, typically between 3.1 GHz and 10.6 GHz
Bandwidth	Greater than 500 MHz (sometimes several GHz)
Range	Short range, approximately 10 to 30 meters under optimal conditions
Localization accuracy	Very high: between 5 and 10 cm
Data rate	Variable, from a few Mbps up to more than 480 Mbps depending on distance
Power consumption	Very low (impulse-based technology)
Security	Excellent (time-of-flight measurement is difficult to spoof with relay attacks).
Latin	Very low (< 1 ms).

User documentation : Module DWM3001CDK (UWB):



This document describes the procedure for setting up the development environment for the DWM3001CDK module (based on the nRF52833 and the DW3110/3000), from installing the tools to compiling the first UWB examples.

1. Communication Principle: PC ↔ Chip (Debug & Flash)

Although it is often referred to as "STLINK" out of habit (a term associated with STM32 chips), the DWM3001CDK module actually uses a debugging interface.**SEGGER J-Link OB**(On-Board) integrated directly onto the development board.

The operation is as follows:

1. **Physical Connection:**You connect the card to the computer via a micro-USB cable.
2. **Logical Interface:**The interface chip (J-Link) converts USB commands from the PC into signals**SWD (Serial Wire Debug)**understandable by the target nRF52833 microcontroller.
3. **Roles of the connection:**
    - **Flashage :**Writing the binary file (`.hex`or`.elf`) in the nRF52 memory.
    - **Debugging :**Step-by-step execution, with stopping points.
    - **UART/RTT :**Uploading logs (text) to the PC terminal (via a virtual COM port or the RTT Viewer).

---

2. Installing the Tools (Windows)

The recommended method uses Nordic's graphical path manager to avoid path errors (`PATH`) and dependencies.

Installation steps:

1. **nRF Connect for Desktop :**Download and install the software*nRF Connect for Desktop*from the Nordic Semiconductor website.
2. **J-Link drivers:**Download and install the latest version of*SEGGER J-Link*(essential for Windows to recognize the card's probe).
3. **Toolchain Manager :**Launch*nRF Connect for Desktop*search for the module**Toolchain Manager**and install it.
4. **VS Code Extension :**From the Toolchain Manager, open VS Code (or install it if necessary). Nordic's recommended method is to install the**nRF Connect for VS Code Extension Pack**directly in VS Code.

   ⚠️ **Warning :**This extension may conflict with other extensions such as*CMake Tools*Consider temporarily disabling these features if you encounter configuration problems.

---

3. Creating and Configuring an Application

Once VS Code is open with the nRF Connect extension active:

1. **Create the application:**
   - In the left sidebar, click on the icon**nRF**(the Nordic logo).
   - In the "Welcome" or "Applications" section, click on**Create a new application**.
2. **Configure the project:**
   - Choose **Create a blank application**(empty application) or start from a template (sample).
   - **Important :**Select a path**court** (ex: `C:\dev\uwb_app`) to avoid compilation errors related to the Windows character limit.
   - Confirm to generate the project tree.

---

4. Compilation (Build)

Once your project is open in the VS Code workspace:

1. **Add a build configuration:**
   - In the panel**APPLICATIONS**(left menu), hover over your project and click on the icon**+** (Add build configuration).
2. **Select the hardware target (Board Target):**
   - For the DWM3001CDK, choose the compatible target, usually:`nrf52833dk/nrf52833`.
3. **Start the compilation:**
   - Click the button**Generate and build configuration**.

---

5. Using the DW3000 (UWB) Examples

To test the location and distance measurement with the Decawave/Qorvo DW3000 module, we rely on the community driver for Zephyr OS.

A. Driver Integration

1. **Clone the example repository:**Open a terminal and run:
2. Bash

git clone https://github.com/br101/zephyr-dw3000-examples.git

3. **Open an example:**
   - In VS Code, do`File > Open Folder`and select the folder of a specific example (for example:`zephyr-dw3000-examples/examples/ex_02a_simple_rx`).
   - The nRF Connect extension will automatically detect the Zephyr project. If it doesn't, click "Create a new application" and point to this existing folder.

B. Hardware Configuration (Overlay)

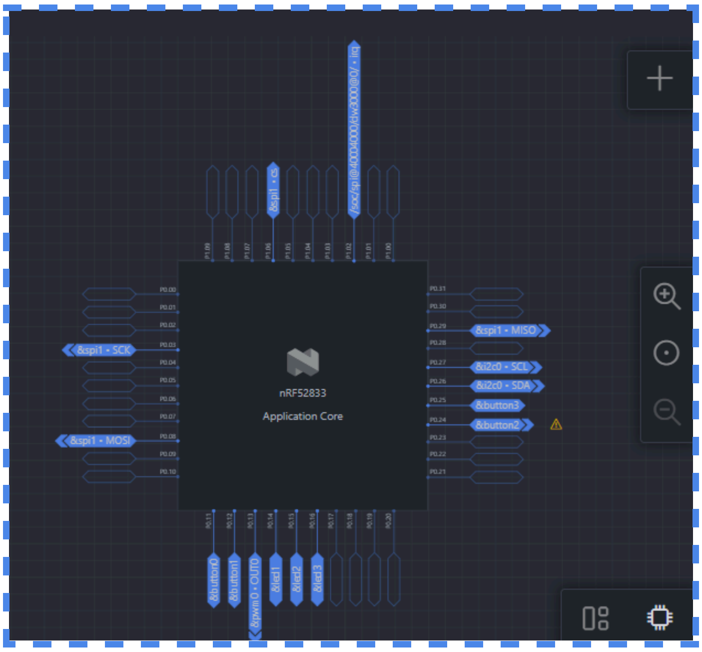
The nRF52 microcontroller communicates with the UWB DW3000 chipset via the SPI bus. It is imperative to verify that the file`.overlay`(or the DeviceTree of the board) correctly defines the physical pins:

- **Bus SPI :**Check the pins`SCK`, `MONEY`, And `MISO`.
- **CS (Chip Select) :**The SPI device selection pin.
- **Control signals:**
  - **IRQ :**The interrupt pin (to signal the receipt of a packet).
  - **RST :**The Reset pin to restart the UWB module.

If you want to view the board's connections, open the nrf connect application and go to the DeviceTree Overlay File tab. You can manually click on each pin to assign the desired connections.

- **Pin Mapping :**

- **SCK:** P0.03
- **MOSI:** P0.08
- **MISO:** P0.29 (with internal Pull-up enabled)
- **Chip Select (CS):** Port 1, Pin 06 (`&gpio1 6`)
- **IRQ (Interrupt):** Port 1, Pin 02 (`&gpio1 2`)
- **Reset:** Port 0, Pin 24 (`&gpio0 24`)



Saving this configuration here is equivalent to modifying the nrf52833dk_nrf52833.overlay file.

```
&pinctrl {
    /* Pin configuration for SPI1 */
    spi1_default: spi1_default {
        group1 {
            /* Mapping conforms to the DWM3001C schematic, page 3 */
            psels = <NRF_PSEL(SPIM_SCK, 0, 3)>,   /* CLK  : P0.03 */
                    <NRF_PSEL(SPIM_MOSI, 0, 8)>,  /* MOTION : P0.08 */
                    <NRF_PSEL(SPIM_MISO, 0, 29)>; /* MISO : P0.29 */
            bias-pull-up; /* Security on MISO */
        };
    };

    spi1_sleep: spi1_sleep {
        group1 {
            psels = <NRF_PSEL(SPIM_SCK, 0, 3)>,
                    <NRF_PSEL(SPIM_MOSI, 0, 8)>,
                    <NRF_PSEL(SPIM_MISO, 0, 29)>;
            low-power-enable;
        };
    };
};

/* We disable SPI3 and UART0 to avoid any conflicts on P0.08 or P0.29 */
```

```
&spi3 { status = "disabled"; };
&uart0 { status = "disabled"; };

/* We activate SPI1 with the correct CS on Port 1 */
&sleep1 {
    compatible = "nordic,nrf-spim";
    status = "okay";
    pinctrl-0 = <&spi1_default>;
    pinctrl-1 = <&spi1_sleep>;
    pinctrl-names = "default", "sleep";

    /* CS is on Port 1, Pin 06 */
    cs-gpios = <&gpio1 6 GPIO_ACTIVE_LOW>;

    dw3000@0 {
        compatible = "decawave,dw3000";
        status = "okay";
        reg = <0>;
        spi-max-frequency = <2000000>; /* 2 MHz for the test */

        /* IRQ is on Port 1, Pin 02 */
        irq-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;

        /* Reset remains at P0.24 as seen in the diagram */
        reset-gpios = <&gpio0 24 GPIO_ACTIVE_LOW>;
    };
};
```

C. Ranging Methods Explained

To measure the distance between two devices, the DW3000 uses **Time of Flight (ToF)**. This is the time it takes for a radio signal to travel from the transmitter to the receiver. Since radio waves travel at the speed of light, measuring this time allows us to calculate the distance.

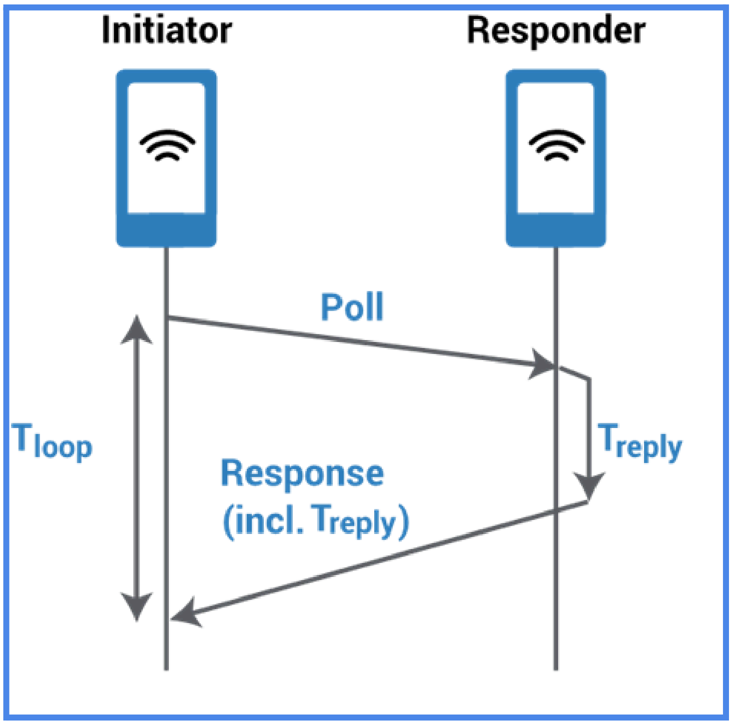
However, clocks on different devices are never perfectly synchronized. To solve this, we use **Two-Way Ranging (TWR)** protocols.

5.1 Single-Sided Two-Way Ranging (SS-TWR)

Concept SS-TWR is the simplest ranging method. It involves a simple round-trip exchange of two messages: a "Poll" and a "Response".

The Process

1. **Poll:** The **Initiator** sends a Poll message and records the transmission time (T1).
2. **Reception:** The **Responder** receives the Poll and records the reception time (T2).
3. **Processing Delay:** The Responder waits for a fixed delay time (to process the data and switch the radio to TX mode).
4. **Response:** The **Responder** sends a Response message containing its timestamps (T2 and T3) back to the Initiator. It records the transmission time (T3).
5. **Completion:** The **Initiator** receives the Response and records the reception time (T4).



Calculation The Initiator calculates the Time of Flight (Tprop) by subtracting the processing time taken by the Responder (Treply) from the total round-trip time (Tround).

Tprop=2(T4−T1)−(T3−T2)

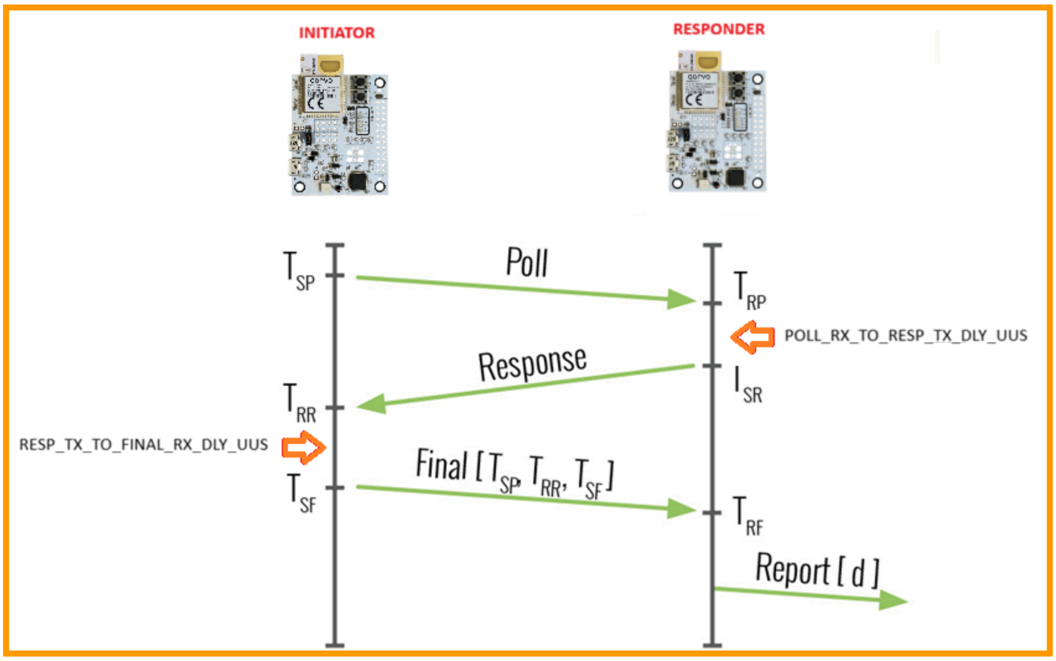
- **Pros:** Fast (only 2 messages), lower power consumption.
- **Cons: Clock Drift Sensitivity.** If the Responder's clock counts time slightly faster or slower than the Initiator's clock, the calculation of the processing time (T3−T2) will be inaccurate relative to the Initiator's timebase. This introduces significant error over long delays.

5.2 Double-Sided Two-Way Ranging (DS-TWR)

Concept DS-TWR extends the Single-Sided protocol by adding a third message ("Final"). This creates two round-trip measurements that are combined to mathematically cancel out the clock drift errors. This is the preferred method for high-precision applications.

The Process

1. **Poll:** The **Initiator** sends a Poll message (T1).
2. **Response:** The **Responder** receives the Poll (T2) and replies with a Response message (T3).
3. **Reception:** The **Initiator** receives the Response (T4). *(Up to here, it is identical to SS-TWR)*.
4. **Final:** After a processing delay, the **Initiator** sends a Final message (T5). This message usually contains the timestamps recorded by the Initiator so the Responder can calculate the distance (or vice versa).
5. **Completion:** The **Responder** receives the Final message (T6).



Calculation The computation uses an asymmetric formula to average the two round-trip times while compensating for clock frequency offsets.

Tprop=Tround1+Tround2+Treply1+Treply2(Tround1×Tround2)−(Treply1×Treply2)

6. Software and Timing Adjustments (DS-TWR)

Context: Relaxing Constraints

We increased the delays to provide a larger time margin for the MCU to process interrupts and schedule radio events.

Responder Modifications (ds_twr_responder.c)
Delays were increased to allow the Responder enough time to switch from RX to TX.

Macro	Old Value	New Value	Function
POLL_RX_TO_RESP_TX_DLY_UUS	900 µs	**1500 µs**	Delay before sending the *Response* to the *Poll*.
RESP_TX_TO_FINAL_RX_DLY_UUS	500 µs	**1200 µs**	Delay before enabling RX for the *Final* message.
FINAL_RX_TIMEOUT_UUS	220 µs	**3000 µs**	RX window widened to ensure *Final* capture.
PRE_TIMEOUT	5	**0**	Preamble detection timeout disabled (infinite wait).

Initiator Modifications (ds_twr_initiator.c)
The Initiator timings were adjusted to match the Responder's new schedule.

Macro	Old Value	New Value	Function

POLL_TX_TO_RESP_RX_DLY_UUS	300 µs + CPU_PROCESSING_TIME	**1200 µs**	Delay before enabling RX for the *Response*.
RESP_RX_TO_FINAL_TX_DLY_UUS	300 µs + CPU_PROCESSING_TIME	**1500 µs**	Delay before sending the *Final* message.
RESP_RX_TIMEOUT_UUS	300 µs	**3000 µs**	RX window widened to ensure *Response* capture.
PRE_TIMEOUT	5	**0**	Preamble detection timeout disabled.

We replaced the dynamic calculation (CPU_PROCESSING_TIME) with a fixed safety margin to ensure the system always has enough time to process data before transmitting.

7. Build Configuration and Deployment

To achieve a functional Double-Sided Two-Way Ranging (DS-TWR) system, specific build steps must be followed.

A. CMakeLists.txt & Source Selection

The `CMakeLists.txt` file must be modified to include the appropriate source file depending on the desired role. Since the project contains both codes, one must be selected for compilation:

- For Board A: Enable `ds_twr_initiator.c`
- For Board B: Enable `ds_twr_responder.c`

B. Board Selection in nRF Connect

In **nRF Connect for VS Code**, the build configuration must target the correct board definition that includes our custom overlay:

- **Target Board:** `nrf52833dk/nrf52833`
- **Overlay:** Ensure `nrf52833dk_nrf52833.overlay` is detected and applied during the build process.

C. Flashing Procedure

A complete ranging system requires two distinct physical devices:

1. **Device A (Initiator):** Flash the firmware compiled with the Initiator logic.
2. **Device B (Responder):** Flash the firmware compiled with the Responder logic.

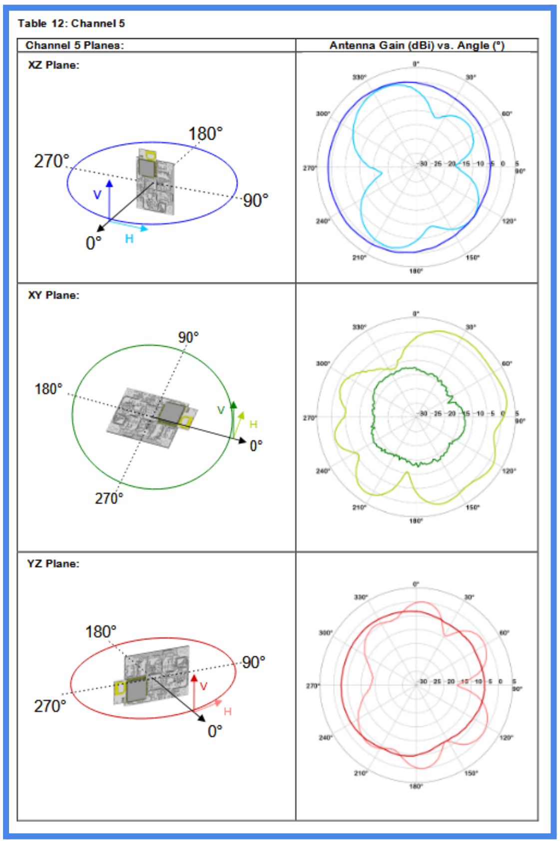
Once both boards are powered, the ranging exchange initiates automatically, and the calculated distance is output via the RTT/Serial console on the Responder side.

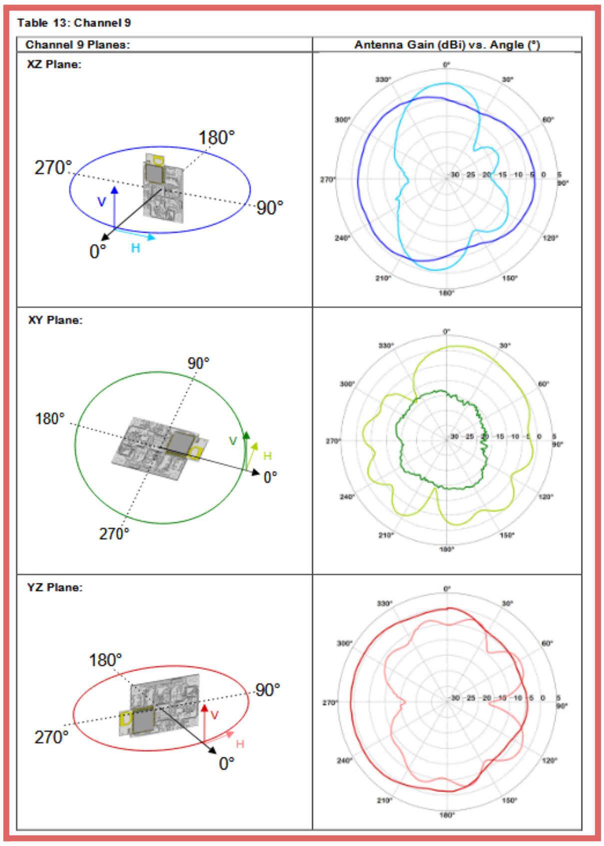
8. RF Propagation and Environmental Constraints

To achieve accurate Time-of-Flight (ToF) measurements, it is essential to understand how UWB radio waves (6.5 GHz – 8 GHz) interact with the physical environment.

Table 6: DWM3001C Receiver AC Characteristics

Parameter	Min.	Typ.	Max.	Units	Condition / Note
Centre Frequency CH5		6489.6			
Centre Frequency CH9		7987.2		MHz	
Channel bandwidth		500			Channel 5 and 9





You can use channel 5 for better penetration through obstacles. Basically, higher transmission power enables better long-range communication.However, due to frequency and propagation characteristics, there are trade-offs. The advantage of channel 9 is that it is less affected by Wi-Fi or other frequencies close to 6.5 GHz. However, it offers a shorter communication range.

c=299792458m/s
The wave length of the channel 9 is λ=c/f= 0.037m
The wave length of the channel 5 is λ=0.046m

8.1 Line of Sight (LOS) vs. Non-Line of Sight (NLOS)

- LOS (Ideal): The most accurate measurements occur when there is a clear, unobstructed straight line between antennas.
- NLOS (Obstructed): When an object blocks the direct path, the signal is attenuated or forced to travel a longer path (reflection).
    - Impact: NLOS conditions typically result in a positive bias, meaning the reported distance will be *longer* than the actual physical distance.

8.2 Material Interaction

Different materials affect UWB signals differently:

- Metal (Conductive): Acts as a radio mirror. It blocks transmission completely and causes strong reflections. Never place the antenna directly against a metal surface; maintain a clearance of at least 15mm.
- Dielectrics (Plastic, Glass, Wood): The signal penetrates these materials but is slowed down by their refractive index. Passing through a wall or thick plastic can add virtual centimeters to the measured distance.
- Water & Biological Tissue: The human body absorbs UWB energy significantly. A person standing directly in the LOS can completely block communication or severely reduce range.

8.3 Multipath and Reflections

In indoor environments, signals bounce off walls and floors, reaching the receiver via multiple paths.

- The UWB Advantage: Unlike narrowband radios (WiFi/Bluetooth), UWB uses ultra-short pulses (< 1ns).
- First Path Detection: The DW3000 chip can distinguish the First Path (the direct, shortest route) from Reflected Paths (stronger but later signals), ensuring accuracy even in cluttered environments.

8.4 Operation in Metal Environments (e.g., Vehicles)

While metal blocks RF signals, communication with metallic objects (like cars) is possible because they are not perfect Faraday cages.

- RF Windows: Signals enter through non-conductive apertures such as glass windows or plastic components (bumpers, handles, mirrors).
- Diffraction & Reflection: In complex environments, the system may rely on signals reflecting off the ground or nearby walls to reach an antenna located behind a metal obstacle.
- Antenna Placement: For successful integration, antennas must be placed behind these "RF windows" and never enclosed entirely within a metal shell.

Bibliography :

PDF QuickStart DWM3001CDK : https://www.farnell.com/datasheets/4509030.pdf
https://download.mikroe.com/documents/datasheets/DWM3001C_datasheet.pdf
https://www.mouser.com/datasheet/2/412/Qovro_7_7_2022_DWM3001CDK_Quick_Start_Guide-2998998.pdf
PDF Antenne du DWM3001C :
https://download.mikroe.com/documents/datasheets/DWM3001C_datasheet.pdf
Github used: https://github.com/br101/zephyr-dw3000-examples.git