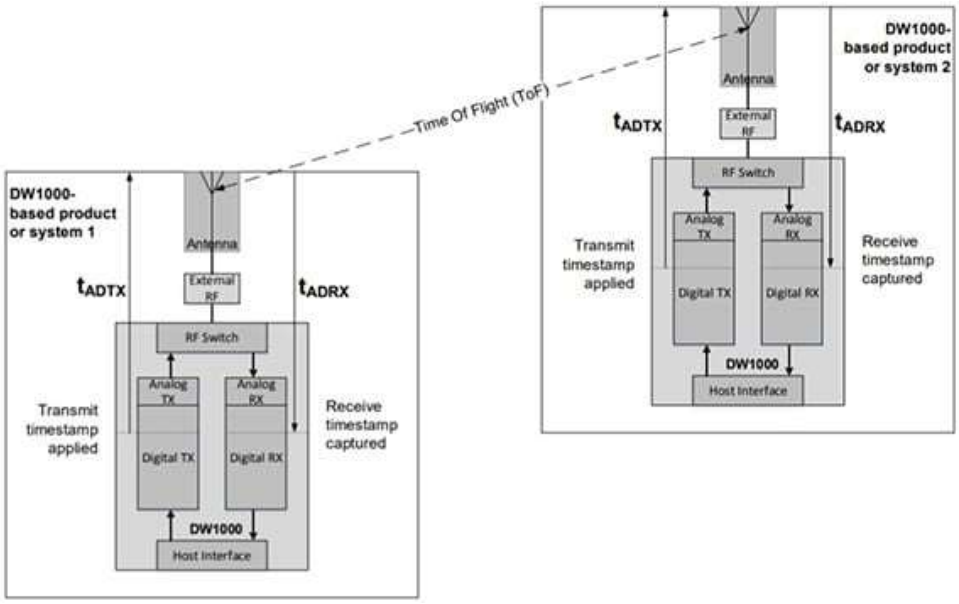
1. ESP32 UWB & DW1000 Antenna Delay

Makerfabs ESP32 UWB has been popular in the last few months, to give Makers a way for Indoor positioning applications. We got some feedback on the accuracy/error in usage, which seems mainly related to the Antenna Delay.

The DW1000 Antenna delay is internal to the chip and not included in the TOF, but are included in the propagation delay from transmission timestamp to receive message timestamp:

tMeasured = tADTX + TOF + tADR

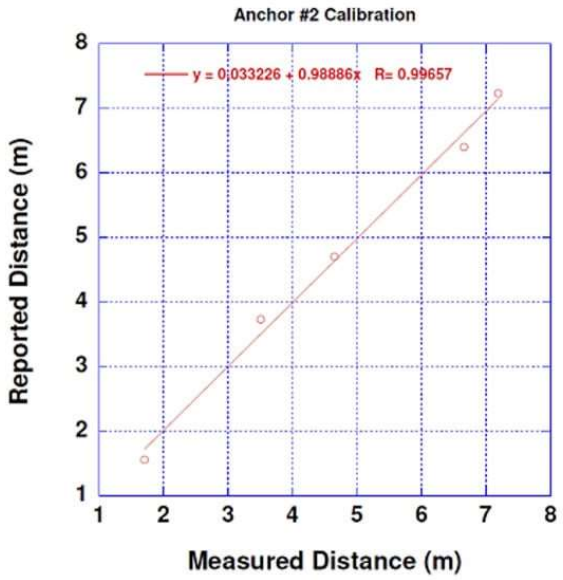
- ToF = Time of Flight
- tMeasured = The measured time from the transmit timestamp to the receive timestamp
- tADTX = Transmit antenna delay
- tADRX = Receive antenna delay



The internal propagation delays in DW1000 devices vary slightly from chip to chip. There can also be variations due to components between DW1000 and the antenna. When the producer makes the IC/ module, it is impossible for them to make/ set the delays differently in the hardware, so it needs the users to set/ calibrate for this in the firmware.

2.How to Calibrate

Thanks to Makerfabs customer Jim Remington, for Makerfabs ESP32 UWB module that runs on Arduino, Jim provides us a simple/easy way to calibrate this, based on Thomas Trojer's DW1000 library, Jim modified it a little to make the antenna calibration there. The updated DW1000 library here. With the antenna calibration, the distance measuring seems more accurate.



The Steps of the Calibration:

1) **Download** the Jim Remington library here to substitute the original DW1000 library;-

2) Download the **ESP32_UWB_setup_tag.ino** to **Tag**;

3) Place the **Tag** and **Anchor** with a fixed distance (such as 8m);

4) Set the distance **at ESP32_anchor_autocalibrate.ino** with 8m, and download the ketch to the **Anchor**;

```
29
30  char this_anchor_addr[] = "83:11:22:EA:82:60:3B:9C";
31  float this_anchor_target_distance = 8.0; //measured distance to anchor in m
32
```

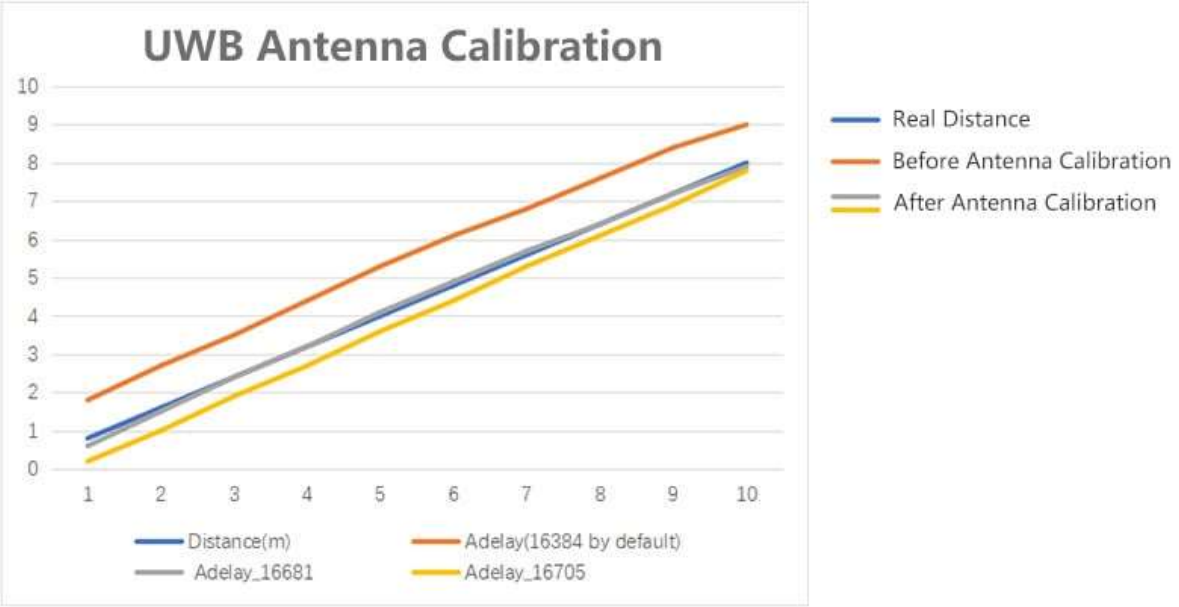
5) Have a record of the **Adelay** from the Anchor's serial output;

6) Reset the **Adelay** parameter in the anchor firmware. (which is 16384 by default) for further measuring.

```
 8  // leftmost two bytes below will become the "short address"
 9  char anchor_addr[] = "84:00:5B:D5:A9:9A:E2:9C"; //#4
10
11  //calibrated Antenna Delay setting for this anchor
12  uint16_t Adelay = 16681;
13
```

3. Comparison: The Data Before & After Antenna Calibration

ID	Distance(m)	Adelay(16384 by default)	Adelay_16681	Adelay_16705
1	0.8	1.8	0.6	0.2
2	1.6	2.7	1.5	1
3	2.4	3.5	2.4	1.9
4	3.2	4.4	3.2	2.7
5	4	5.3	4.1	3.6
6	4.8	6.1	4.9	4.4
7	5.6	6.8	5.7	5.3
8	6.4	7.6	6.4	6.1
9	7.2	8.4	7.2	6.9
10	2	9	7.9	7.8



We can see that, with the calibrated **Adelay** parameter, the measured distance (color **Yellow** and **Gray**) get much closed to the real distance (color **Blue**). With a proper Adelay set, the measurement accuracy could be improved greatly.

4. Conclusion

(1) The antenna calibration improves the UWB distance measuring much.

(2) The measured distance larger as Adelay spec bigger.

Source: link

5.Extras

According to our setup, we have placed the anchor and tag at a distance of 1 meter from each other but the Serial Monitor shows a distance above 2 meters.

From here we can say, the measured distance is incorrect. This is because of the Antenna Delay problem. The DW1000 Antenna delay is internal to the chip and not included in the Time of Flight (TOF) calculation. Thus we need to calibrate this Antenna Delay issue.

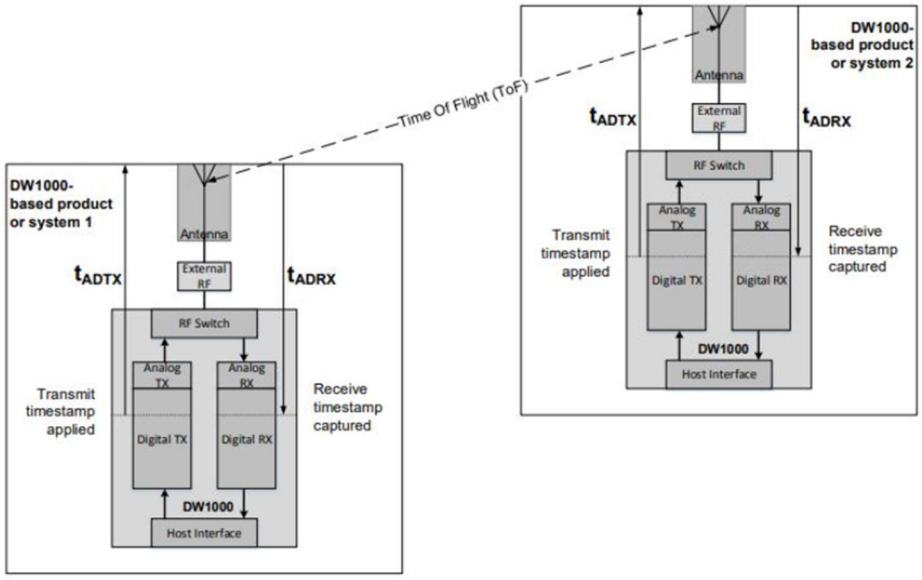
What is Antenna Delay of DW1000?

DecaWave's DW1000, a multi-channel transceiver based on Ultra Wideband (UWB) radio communications, allows very accurate time-stamping of messages as they leave from and arrive at the transceiver.

The delays which are measured in these timestamps include the propagation delay through the DW1000 devices from the points at which the transmitter timestamps are applied to the points at which the receiver timestamps are captured. These delays are referred to as the

transmit/receive antenna delays.

These antenna delays are internal to the chip and not included in the Time of Flight (ToF) but are included in the propagation delay from transmission timestamp to receive message timestamp.



$$tMeasured = tADTX + ToF + tADRX$$

where:

ToF = Time of Flight

tMeasured = The measured time from the transmit timestamp to the receive timestamp

tADTX = Transmit antenna delay

tADRX = Receive antenna delay

The internal propagation delays in DW1000 devices vary slightly from chip to chip. There can also be variations due to components between DW1000 and the antenna. Since we are measuring RF signals moving at the speed of light these variations can make differences to ranging measurements in the tens of centimeters. Antenna delay calibration is used to remove these variations.

Antenna Delay Calibration of DW1000

To fix the Antenna Delay issue the above Thomas Trojer's DW1000 library is modified by Jim Remington.

The updated DW1000 library from Jim can be downloaded from jremington Github repository. With the antenna calibration, the distance measuring seems more accurate.

Delete the old DW1000 library from the Arduino Library folder and add the latest downloaded jremington DW1000 library file to the Arduino library folder.

ESP32 Anchor Autocalibrate Code

Copy the following code and upload it to the first ESP32 UWB Board.

In the following code, make changes to the following line:

```
float this_anchor_target_distance = 1; //measured distance to anchor in m
```

Replace the distance at which you are calibrating. Calibration code is here.

Calculating the Adelay Parameter

After uploading the code, open the both Serial Monitor. Place the module at a fix distance of 1 meter. Then press the reset button.

The **DW1000 Antenna Delay Calibration** factor called **Adelay** can be determined from the code. In the Serial Monitor, the **Adelay factor** is something like **16586**. Copy this number as it is required in the final Code.

Measuring the distance Accurately with Adelay inclusion

Now the **DW1000 Antenna Delay Calibration factor** called **Adelay** is determined. Therefore, we need to upload the **final code** to the Anchor board. The code for tag remains the same.

ESP32 UWB Setup Anchor Code

The code needs modification in these two lines.

```
uint16_t Adelay = 16611;
float dist_m = 1; //meters
```

Replace the **Adelay** factor in the code and also mention the **distance** at which you found the factor.

Source: link