Proposal

CS497 | Spring 22

Chris Uustal, Jason Paul, Eugene Choe

Project Overview

We want to do research into ranging and ultra low power applications with UWB. Chris is working for a company this summer which may be looking into it, so he is interested in getting more hands-on experience with the technology myself. Jason is more interested in the ultra-low power aspect of the project. Beyond that, we've all done prototyping within well established and classical domains with embedded systems, but UWB is quite new, so it may present us with some unique and interesting challenges. Additionally, none of us have ever used a wireless communication protocol which can accurately locate devices in relative 3D space like UWB is able to (BLE RSSI doesn't count), so this opens a whole new world of prototyping with devices aware of their location in 3D space.

Chris is a big fan of Arduino, so our plan was to use the DWM1000 module paired with an ESP32 or Teensy 4.1 for control programmed in Arduino using the DW1000 library. This library will be the basis of our work, but (at least according to the readme) it seems a little… rough around the edges. The basic features that we care about like basic communication and ranging appear to already be supported, but it also has lots of mentions of unimplemented features, bugs, and incomplete error handling, so there might be a lot of work to do.

Goals

At its core, we just want to get basic UWB ranging working consistently and reliably to a point where we can actually characterize its precision and accuracy. This might just *work* out of the box, but it also might not, so we're going to be very conservative and leave that as our only absolutely required goal that we all work on together initially. If that works without issue, then we'll split the project into whichever 3 of the stretch goals appear most feasible based on the existing code base.

We currently expect that ranging functionality to work relatively easily, so we assume the bulk of the project time will be spent on the stretch goals. Those goals include:

- Direction of travel approximation
- Relative device angle of attack
- Object interference detection (eg. if something suddenly passes between the two radios)
- Device power consumption optimization (eg. basic sleeping & waking of the device at regular intervals)
- Characterization of ultra-short range (<1m) range measurement accuracy
- Interference redundancy characterization

If all of these things turn out to be really easy or basically impossible, we'll revisit this list and maybe revise it, but this feels like a solid target to aim for in the span of only a month or two.

Requires Resources

Chris already has a bunch of ESP32's and a few Teensy's laying around, so we'll probably just use those for the MCU. We also already ordered two of the DWM1000 and designed + ordered a breakout board for it, so we don't assume we'll need any more than that for basic relativistic ranging. If we end up needing a third device for whatever reason (maybe to do interference testing), we'll just buy one at that point since they're well stocked and relatively cheap ($16).

Schedule for Completion

None of us like GANTT charts, so here's a schedule of what approximate deliverables we'd like to have done each week assuming the Arduino library is at least somewhat functional:

Due Date	Deliverable(s)
May 2	Get the library working and talk to the DWM1000
May 9	Get two DWM1000's to talk to each other; maybe get ranging (if it works)
May 16	Divide up stretch goals and determine their feasibility
May 23	Finish individual stretch goal core functionalities
May 30	Project done, set up demo(s) w/ some kind of visualization
June 6	Project report

If the library ends up being trash / not working, then all of May 16 + 23 will end up likely being devoted to getting that core functionality from May 9 working by the 30th.

Areas of Concern

Like we mentioned in the project overview, the main thing we're a little uncertain about right now is the actual firmware to interact with it. While this module has been out for quite a while (5+ years), it seems like there's basically just one dude who grinded out a library for it and that's about it. Not only that, but that library also seems to be fairly characteristic of something made by a single human—it has lots of small edge cases it doesn't handle properly, and it only implements the most important core features. With that said, the documentation from decaWave (the company that made the DWM1000) is very comprehensive (solid 242 page user manual) and we're not particularly scared by having to manually construct elements of the SPI driver for it if necessary. There is a small risk here this becomes an exercise of "figure out how to talk to the chip" instead of "figure out all the cool stuff you can do with the chip," but we assume we'll have a good idea of how that's going within the first week or so. If it does end up being the former situation, then we hope to just get it working and get the two ranging and barely meet my requirements. If it ends up being the latter—awesome! We get to actually play around implementing useful thing with UWB like are outlined in the extended features.