RF-Ultrasound Distance Measurement for Indoor Localization – Final Report

Advisor: Dr. Bhaskar Krishnamachari

Author: Hou In (Stanley) Lio

Project Start Date: January 2015

University of Southern California

May 2015

[First Draft]

Table of Contents

Introduction......1

System Description......1

  Operation......1

  Implementation Details......2

Further Work......2

Appendix......3

  What Didn't Work......3

  A Note on Sonar Selection......3

  Effect of Timing in Software......3

Introduction

The goal of the project is to build a prototype system to evaluate a combined RF-ultrasound method for distance measurement as an alternative to camera-based technique for use in indoor localization. The key idea is to infer the distance between a Beacon and a Node by measuring the time of flight of ultrasound, with RF signal used for synchronization. The time of flight of the RF signal is taken as instantaneous (zero) regardless of distance. Once distances from a Node to multiple fixed Beacons are known, position of the Node can be found by trilateration.

A working prototype has been produced and the typical accuracy of the demo system was sub 10 cm (4”) at a distance of < 8 m (26’) (line of sight).

System Description

The hardware consists of two parts: the Beacon and the Node (Figure 1). There is one RF transceiver and one ultrasound transceiver on each end. The RF and ultrasound transceivers are used as transmitter on the Beacon, while on the Node they are used exclusively as receivers. The variable of interest is the physical distance between the Beacon and the Node.

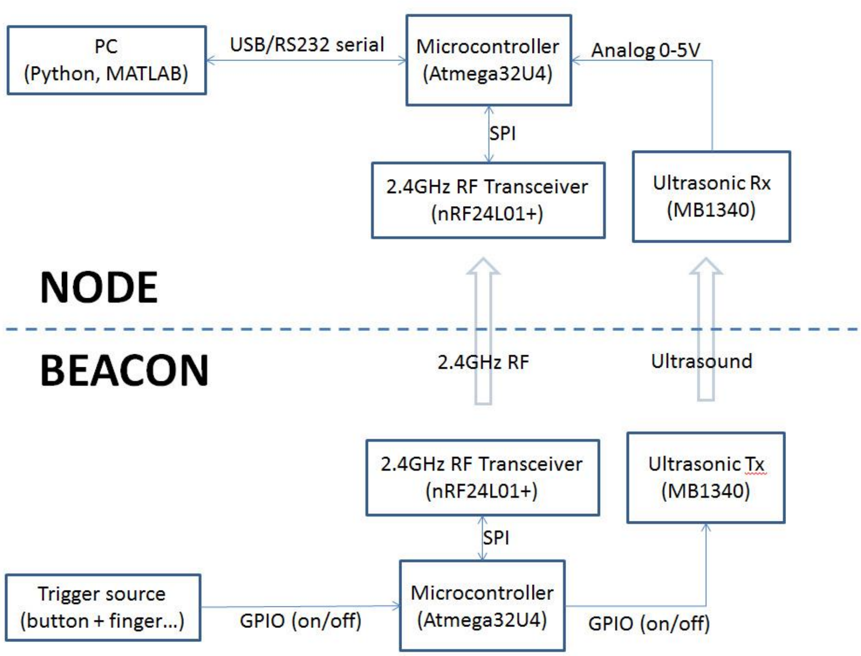

Figure 1, system block diagram

Operation

Overall, when a new reading is requested (triggered by a push-button in the demo), the output of the system is a single distance reading reported in meter, displayed on PC via serial port.

The sequence of events in a typical reading goes as follow:

1. Beacon sends an RF ping
2. Immediately afterward, Beacon sends an ultrasound pulse
3. Reception of the RF ping at the Node starts a timer
4. The ultrasound ping reached the Node; the timer is stopped
5. Time from RF reception to ultrasound ping detection is converted and reported via serial port

Implementation Details

The microcontrollers (Arduino Pro Micro clones) are responsible for all time-critical operations. The wireless transceivers are generic 2.4GHz nRF24L01+ breakout boards. The MB1340 ultrasonic ranging sonars are inhibited for normal ranging operation. It is enabled only when transmitting a single ultrasonic pulse.

The ultrasound ping carries no identifying information. The RF ping was an arbitrary ASCII character `'a'`. When more than one Beacon is used together, the ping can be used to identify the Beacon (for example, naming the Beacons from `'a'` to `'z'`).

In the prototype, the detection of ultrasound pulse is done with a fixed threshold. The timer is stopped the first time the received ultrasound intensity rises above a voltage level. Looking forward, a (rising) edge detection can be used instead to avoid hard-coding the fixed threshold.

The RF-ultrasound time of flight delay is measured by counting the number of ADC sampling periods between the two events. Say if the mapping from sampling period to physical distance is 10 ADC sampling periods per meter, then 25 periods correspond to a distance of 2.5 meter. The precise length of a tick does not need to be known precisely in terms of second (see the section "Effect of Timing in Software").

[Circuit diagram goes here? That depends on the purpose of this doc: if this is used to produce more units then detailed spec is indeed needed. However if this is an operation manual or a project report, such level of detail may not be appropriate (put in a separate technical document perhaps).]

Further Work

A prototype produced, demonstrated the feasibility and evaluated the performance. To aid production of more units, a printed circuit board should be produced (extra cost and time, proposal not yet approved).

Appendix

What Didn't Work

An attempt was made to port the demo system to the existing Tmote Sky + Contiki platform, making use of its built-in RF radio and microcontroller. Upon evaluation of the Tmote Sky + Contiki combo, it was found that the lack of real-time functionality in Contiki OS makes the platform unsuitable for replacing the microcontrollers. In particular, rather than having hardware interrupt notifies the main thread of the arrival of the RF/ultrasonic ping as soon as they arrive, Contiki would notify the main thread only when it "gets around to it" – the OS polls the communication stack at 8Hz by default. The polling rate can be increased to 128Hz maximum by tweaking the Contiki OS source code, but one period at 128Hz still corresponds to over 2.7 meter (8.9') of uncertainty in distance.

A Note on Sonar Selection

Notice that although many of the ultrasonic sensor modules on the market report the distance to an obstructing object, these sensors **cannot** be used because it is the distance between the Beacon and the Node that is being sought after, not the Node's distance to an arbitrary object.

The MaxBotix EZ1340 sonar range finder was chosen for its analog output[1] pin, which outputs the received ultrasound envelop (filtered and amplified) in real-time in addition to the estimated distance. For this purpose, the entire line of EZ13xx series can be used (they all provide the analog envelop, but differ in beam width and sensitivity).

Effect of Timing in Software

There are several unknowns in timing that could contribute to measurement accuracy. For example, in the Beacon, the delay between the transmission of the RF ping and the ultrasound ping as well as the time from enabling the ultrasound circuit to an ultrasound pulse actually being generated are not measured. In the Node, the ultrasonic sensor response time to the ultrasound pulse and the time it takes the software to respond to hardware interrupt are not measured either.

Their effect to the measurement accuracy can be eliminated by simple calibration. By measuring the time of flight of the ultrasound ping at various known distances:

[1] This is the unprocessed real-time ultrasound intensity signal, which is **not** a distance measurement. This should not be confused with the "analog output" of many such sensors which outputs a voltage level proportional to the distance measured. Effectively, only the analog filters, amplifiers and ultrasound modulation functions are used (while bypassing its internal distance measurement function).

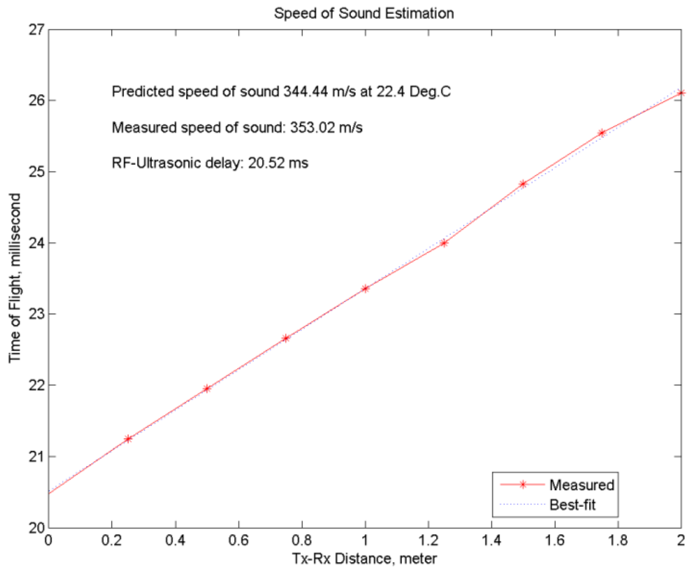

Figure 2, linear best-fit for calibration

In Figure 2, the delay at distance 0 is the delay between the RF ping transmission and the ultrasound pulse detection, while the speed of sound is given in the slope of the best-fit line. For this reason, even the ADC sampling rate at the Node side doesn't need to be known since the distance estimation can be calculated by mapping the sampling period count directly to physical distance.