Transmission Protocol — iCRAB

Noah Strong

January 27, 2018 – v1.1

Contents

1 Introduction

The Crab Tracker project aims to provide a simple, efficient, reliable, and cost-effective method for tracking crabs underwater. There are no accepted standards that we're aware of for achieving the results we hope to achieve, and to base our work too heavily off the work of existing products would violate the clauses in the licenses of those products that protect against reverse engineering. For these reasons and more, we must define our own technologies and protocols. Central to the project is the protocol that will be used to relay information from transmitters (attached to crabs) to the central receiver (affixed to a water-going vessel, such as a kayak). Documented herein is that protocol, as well as the motivations and requirements for many of the decisions behind it. As of this writing, **the protocol is still subject to change.** We may find shortcomings or other problems with the protocol during the prototyping stage of the product, at which point adjustments will be made. This document will be updated as needed to reflect these changes, and should always be treated as the official documentation for the protocol.

One of the major requirements of this project is the ability for each individual transmitter to be uniquely identifiable. Therefore, we must encode the device's unique identifier (herein referred to as the ID or UID) in each signal that the device broadcasts. Additionally, we may want to encode some accelerometer data to indicate that a given transmitter has stopped moving. This data will be encoded as a boolean value, which we refer to as `inert`.

Finally, because all transmitters transmit at the same audio frequency and their broadcasts are uncoordinated (no transmitter is aware of the broadcasts of adjacent transmitters), it is possible for two or more transmissions to overlap. In the event that two or more broadcasts are received simultaneously, the receiving software should always be able to detect this collision. Simple implementations of an encoding protocol can lead to situations in which collisions are not detectable, but the protocol proposed in this document aims to prevent the possibility of undetectable collisions. For a further discussion on how collisions may arise, proposed solutions, and other background information, please see RFC 1[1].

2 Background

For more background on some of the challenges, motivations, and decisions that have come up in the creation of this protocol, see RFC 1[1] (a discussion of collision possibilities and remedies) and RFC 2[2] (an examination of ways to encode a boolean value).

To satisfy the requirements of this project, we are designing a new protocol. This protocol will encode the UID of each transmitter in such a way that collisions (multiple simultaneous transmissions by different transmitters) can be detected by a receiving station and discarded. The protocol is a simple series of HIGH and LOW audio signals operating at a predefined frequency. (The specific frequency to be used will be documented elsewhere on the hardware engineering side of things.) The series of "pings" and the separation between them will be organized in a specific pattern based on the UID of the given transmitter. We'll generally refer to this pattern as the Unique Transmission Pattern (or UTP for short).

Additionally, we may want to have the ability to detect when a transmitter has stopped moving, possibly because the crab molted its shell or died. In this case, we want each broadcast to not only encode the UID of the transmitter, but also some boolean value.

Given the nature of the protocol and the project itself, we have named this specification the *id-correlated rhythmic audio broadcast* protocol, or iCRAB for short.

3 iCRAB Protocol Definition

3.1 Overview

At the core of this protocol is a single burst of information which is transmitted repeatedly on some randomly-varying interval. All transmitters have a unique identifier (an integer number) associated with them, and each burst of data will encode that identifier. The burst, hereafter referred to as a unique transmission pattern (UTP), will consist of two pings (short, continuous transmissions of the carrier frequency), separated by some delay time d. The duration of the pings and the delay time d will be functions of the transmitter's UID. The interval between UTPs will be random, and each transmitter will recalculate the interval time after each UTP according to a shared formula. See Figure 1 for an example of a UTP.

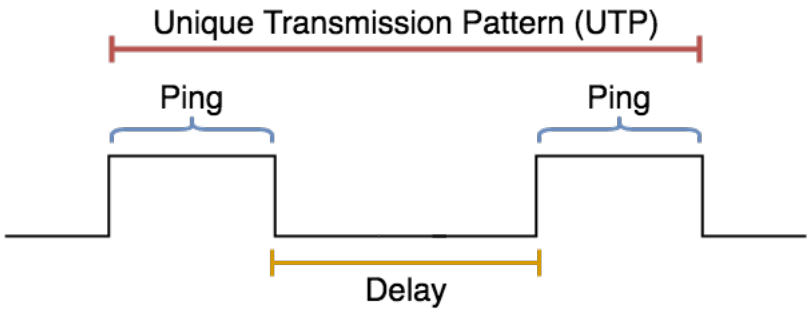

Figure 1: A Unique Transmission Pattern

3.2 Requirements, Rationale and Related Information

Based on the rough estimates given to us by our colleague in Electrical Engineering early in the project's history, we will initially comply with the following constraints.

1. The minimum ping duration should be 1 millisecond.
2. The "step size" (i.e. smallest difference in duration between any two pings) should be 0.1 milliseconds.
3. The minimum delay duration should be 10 milliseconds.
4. The "step size" for the delay should also be 0.1 milliseconds.

The remainder of this section provides details about requirements of the protocol, some reasoning for our choices, and other relevant information. However, this is meant to be a general look at the protocol; all specifics, including timing information, ID encoding ranges, and other constants, are detailed in Section 3.4. The motivation for leaving the information in this section (Section 3.2) somewhat unspecific is that requirements may change as we test and prototype our technology, and we'd prefer to have one single source of truth for all numerical values so as to reduce confusion.

Each transmitter will periodically broadcast a signal that the receiving system can detect. The receiving station will process the data encoded in the broadcast and also determine the direction, relative to the system, of that transmitter. The UTP broadcasts should be frequent enough for location updates to be practical to the user, while also being infrequent enough that the expected number of collisions is very low. For more information on collision statistics, see RFC Stats[3].

Observe, however, that if every transmitter transmitted on a fixed interval, then we could theoretically encounter a situation in which two transmitters transmit at almost exactly the same time for the entire duration of their deployment. In other words, every single one of their transmissions would collide until one of the transmitters depleted its battery charge, and the receiver would never be able to reliably determine the location of either one. To remedy this, we will instead randomly vary the interval between broadcasts. That way, if two transmitters happen to transmit at the same time in one instance, they won't necessarily collide the second time around. The random adjustment will need to be recalculated for each interval in order to reduce our chances of collisions.

The iCRAB protocol is expected to support only a finite number of transmitter UIDs. The upper bound selected is fairly arbitrary but based on the client's expected needs for the project's research.

Finally, this protocol is also expected to be able to encode (along with the ID of a given transmitter) a boolean value. For the purposes of the project, this value will be `true` if and only if a given transmitter is inert. This generally indicates that the crab died or molted its shell. We will encode this boolean value by effectively doubling the number of IDs that can be encoded. Half of the IDs are to be used when the boolean value is `false`, and the other half are used when it is `true`. Specifically, values in the range $[0, MAX\_UID)$ are to be used when `inert=false`. There is no additional mapping when IDs are in this range. For all values in the range $[MAX\_UID, 2 * MAX\_UID)$, the value encoded is actually MAX_UID bigger than the ID that is encoded, and `inert=true`.

For example, if a signal encodes the value 42, the receiver should report this as transmitter 42 with `inert=false`. However, if the value 542 is received, assuming that $MAX\_UID = 500$, the receiver should interpret this as transmitter 42 but with `inert=true`.

For more background on the topic of boolean value encoding, see RFC 2[2].

3.3 Preamble to Definitions and Specifications

Each Unique Transmission Pattern (UTP) will be formed by a ping, a delay, and another ping, in that order. Each transmitter will broadcast a UTP and will then wait for some interval before transmitting a UTP again. This process loops indefinitely throughout the transmitter's lifetime.

Defined in Table 1 are the constants that we will use for the various aspects of this protocol. This section of the document will be updated as needed if these values change.

Table 2 lists the various mathematical formulae we will use for encoding. Some functions are passed a single integer value, which is a UID.

Between each UTP broadcast, every transmitter should wait for some amount of time. As mentioned previously, this interval should include some random variation that is recalculated each time. The length of each interval should be a number of seconds in the range $[MIN\_INTERVAL,\ MAX\_INTERVAL]$.

Finally, we formally define the behavior of a transmitter in pseudo-code (see Listing 1.) The functions *HIGH* and LOW cause the physical transmitter to begin or cease transmitting, respectively. The *sleep()* function simply causes execution of the code to stop for a given number of milliseconds.

Section 3.4, on the following page, aims to include relevant constants, formulae, and other definitions that developers will need while programming the components of this project. This section is our single source of truth for the values used in the iCRAB protocol, and it will be updated as needed throughout the prototyping and testing phases of the project.

The remainder of this page is intentionally left blank.

3.4 iCRAB Definitions and Specifications

Constant	Value
MIN_INTERVAL	15 s
MAX_INTERVAL	20 s
MAX_ID	499
MIN_PING_DUR	1.0 ms
MIN_DELAY_DUR	10.0 ms
STEP_SIZE	0.1 ms

Table 1: Constants to be used for the iCRAB Protocol

Name	Definition
ping(id)	$(id \times STEP\_SIZE) + MIN\_PING\_DUR$
delay(id)	$(id \times STEP\_SIZE) + MIN\_DELAY\_DUR$

Table 2: Formulae to be used for the iCRAB Protocol

```
1  void doPing(int id){
2    HIGH()
3    sleep(ping(id))
4    LOW()
5  }
6
7  void loop(id){
8    doPing(id)
9    sleep(delay(id))
10   doPing(id)
11   sleep(interval())
12 }
```

Listing 1: Transmitter Behavior

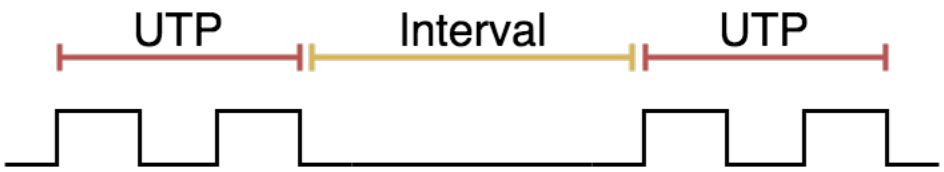

Figure 2: For reference, pictured are two UTPs separated by an interval

4 References

1. Strong, N. *RFC 1: Collisions in Delay-Based ID Encoding Protocol.* 2017. PDF. `https://github.com/cabeese/crab-tracker/blob/master/doc/rfc1/RFC1.pdf`

2. Strong, N. *RFC 2: Boolean Value Encoding in Transmission Protocol.* 2018. PDF. `https://github.com/cabeese/crab-tracker/blob/master/doc/rfc2/RFC2.pdf`

3. Yugawa, C. *RFC Stats.* 2017. PDF. `https://github.com/cabeese/crab-tracker/blob/collision-statistics/doc/rfc1/RFC1_stats.pdf`

A Glossary of Terms

Delay: in the context of ID encoding, the space between the falling edge of one **ping** and the rising of the next within a single **UTP**.

Delay Time (d): the duration (generally in milliseconds) of a given **delay**.

iCRAB (id-correlated rhythmic audio broadcast) protocol: the protocol designed by the members of the Crab Tracker project and described in detail in this document.

Inert: A transmitter will be marked as **inert** if it is determined that the transmitter has not moved "enough" in a given period of time. This definition is subject to change based on hardware constraints, and its complete definition is outside the scope of this document.

Interval: the time between two consecutive broadcasts of **UTP**. Measured by the distance between the final falling edge of one ping and the first rising edge of the next.

Ping: a single, continuous transmission of signal.

Ping Duration: the length of time between the rising and falling edges of a continuous transmission (a **ping**).

Unique Transmission Pattern, UTP: a sequence of two **ping**s separated by some **delay** used to encode the unique identifier of a transmitter.