## 1 System design

This is the current scenario:

In this scenario we have the IR channel and the I2C channel. The IR is our main channel, while the i2c is used to send the expected message that the receiver should receive. In Figure 1 is an overview of how the whole system for the BER computation works.

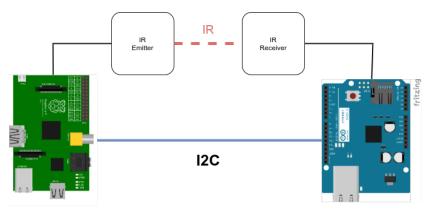


Figure 1: Simple scenario diagram

In the following figure there is a flowchart to understand how the data is processed:

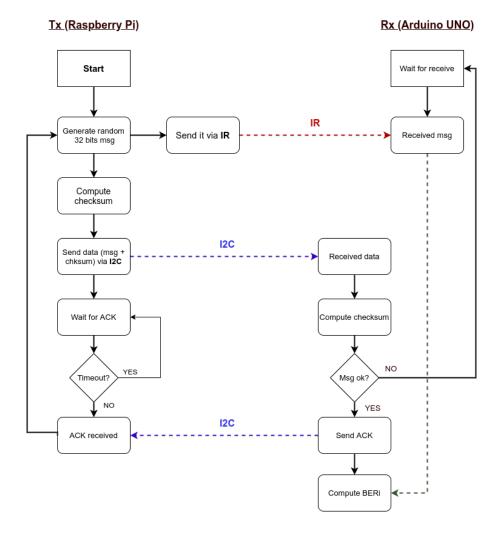


Figure 2: Flowchart for the BER computation

## 1.1 The I2C bus

The main purpose of this channel is to let the receiver know what the expected message received through IR channel is. Since we are sending different messages all the time, the receiver needs to know what the expected message is, in order to properly compute the BER.

## 1.1.1 Error control protocol

A simple Stop & Wait protocol has been implemented in order to decrease the chances to receive a wrong message through the I2C bus. The workflow of the protocol can be seen in Figure 2.

The data sent consists in a 4 bytes message and 1 more byte that corresponds to the checksum. Regarding the checksum computation, it consists in getting the first byte of the sum of each byte of the message. Below, there is the detailed explanation of the packet generation sent via I2C.

Let the message be  $M = [m_0, m_1, m_2, m_3]$ . Where M is a 32 bit word randomly generated and  $m_i$  is each bit of the message.

Let the checksum CK be the first byte of the sum of each byte of the message:

$$CK = \sum_{i=0}^{3} m_i \wedge 255$$

Then, the packet sent consists in P = [M, CK] with a length of 5 bytes.

## 2 BER computation

- Let the expected received message be  $X = [x_0, x_1, ..., x_N]$ . Where X is a 32 bit word randomly generated. N is the number of sent messages and  $x_i$  is a 32 bit word randomly generated.
- Let the number of wrong bits in the each received message be  $WB = [Wb_0, Wb_1, ..., Wb_N]$ . Where  $Wb_i$  corresponds to the amount of errors in a 32 bit word, i.e each received message.

The BER of each received message  $x_i$  is  $BER_i = \frac{Wb_i}{32N}$  since all the received signals are 32 bit words. So we have a N dimension array of BER:  $BER = [BER_0, BER_1, ..., BER_N]$ .

Then the average BER is:

$$BER_{AVG} = E\{BER\} = \frac{1}{32}E\{WB\} = \frac{1}{32}\sum_{i=0}^{N}\frac{1}{N}Wb_i = \frac{1}{32N}\sum_{i=0}^{N}Wb_i$$