Lecture 2 / Homework 2 for The Software Defined Radio (SDR) of Purdue VIP

Prepared by Prof. Chih-Chun Wang

The main difficulty of learning how to use the USRP is due to the following two challenges.

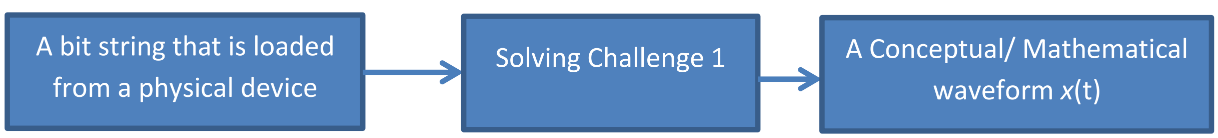
Challenge 1: We need to learn the basic principle of digital communication.
Challenge 2: We need to learn how to implement a communication system in a digital way.

In Lecture 1, we have briefly gone over Challenge 1. In this document, we will go over Challenge 2.

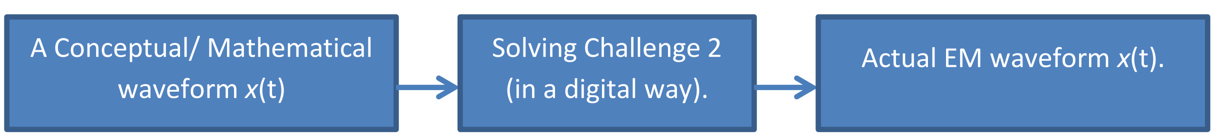
Along our discussion, you will be assigned several "tasks". Please complete the tasks in a different notebook and submit them as homework assignment. Your final grade will be partially based on your performance on individual tasks.

Challenge 2: Learning how to implement a communication system in a digital way.

In Lecture 1, we have discussed how to design an analog waveform $x(t)$ to represent a digital bit string 010110….



The remaining issue is how to transmit the analog waveform x(t) through hardware implementation.



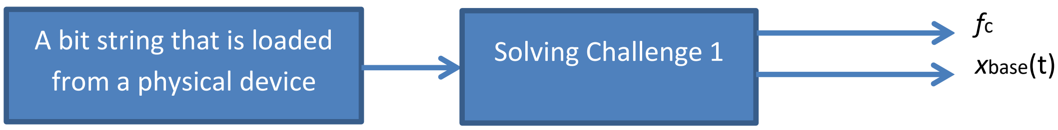
As can be seen, nowadays the second challenge is also solved digitally. That is, there will be plenty of digital manipulation when solving Challenge 2. It should not be confused with how we solve Challenge 1, which also involves a digital bit string. It turns out that these two challenges can be completely decoupled. Namely, no matter which way we use to solve Challenge 1, the final output of solving Challenge 1 is always an analog waveform x(t). Challenge 2 focuses exclusively on the analog waveform input x(t) and does not care how the x(t) is generated.

Since it involves building actual hardware, in general, there is only a limited number of conceptual/mathematical $x(t)$ can be successfully converted to the actual EM wave $x(t)$. That is, the input of the building block must be of the form

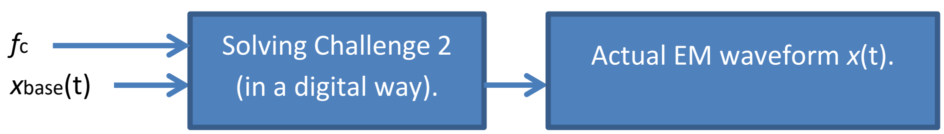
Equation 1

$x(t) = x_{base,Re}(t)\cos(2\pi f_c t) - x_{base,Im}(t)\sin(2\pi f_c t)$

As engineers, we like simpler expression. Therefore, instead of writing $x(t)$ in the above quite long form, we can now write the input as two parts: the so-called *base-band signal* $x_{base}(t)$ and the so-called *carrier frequency* f_c. Where the base band signal $x_{base}(t) = x_{base,Re}(t) + j\, x_{base,Im}(t)$ is a complex-valued signal containing two real-valued signals $x_{base,Re}(t)$ and $x_{base,Im}(t)$, respectively. With the above new representation, we can change the above block diagrams to



and



Remark: when solving Challenge 1 (when choosing the $f_0(t)$ and $f_1(t)$ waveforms as discussed in Lecture 1), we also need to make sure that the final output waveform $x(t)$ can be written as in Equation 1.

Task 1: When choosing f_c=2 Hz and $x_{base}(t) = 1$, plot the actual target waveform $x(t)$ for this particular choice.

Task 2: When choosing f_c=1 Hz and $x_{base}(t) = \cos(2\pi t) + j\sin(2\pi t)$, plot the actual target waveform $x(t)$ for this particular choice. What have you learned from Tasks 1 and 2?

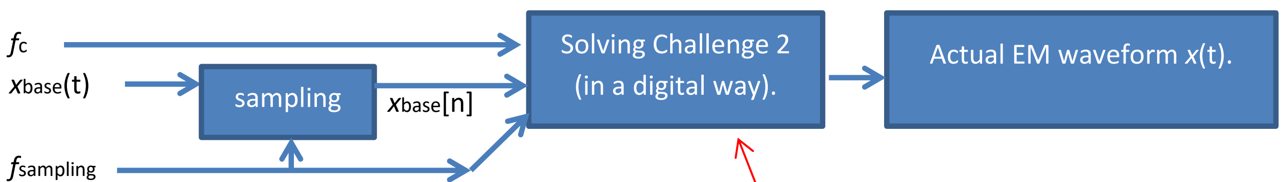
As one can envision, since we would like to use a digital way to solve Challenge 2, the very first step towards solving Challenge 2 is to use ADC to sample and convert $x_{base}(t)$ into a digital array $x_{base}[n]$. One popular choice of the *sampling rate* is 128k samples per second.

Task 3: Figure out what is the sampling rate used by the transmitter of our hardware this semester (usually, the hardware choices are USRP, USRP2, and Sidekiq). You can consult with TA for this question.

Task 4: Suppose we are only interested in sending the waveform between t=0 to 4 seconds. Suppose we use the sampling rate being *128 samples per second* (not 128k), and also suppose that we choose the f_c and $x_{base}(t)$ in the same way as in Task 1. What is the resulting digital array $x_{base}[n]$ (in terms of mathematical expression)? How many entries are there in the digital array $x_{base}[n]$? What are the values of the first 10 entries of the array?

Task 5: Suppose we are only interested in sending the waveform between t=0 to 4 seconds. Suppose we use the sampling rate being *128 samples per second* (not 128k), and also suppose that we choose the f_c and $x_{base}(t)$ in the same way as in Task 2. What is the resulting digital array $x_{base}[n]$ (in terms of mathematical expression)? How many entries are there in the digital array $x_{base}[n]$? What are the values of the first 10 entries of the array?

Based on the above discussion, we can further rewrite the block diagram of Challenge 2 by



As a result, the hardware only needs to focus on the three different inputs: the carrier frequency f_c, the sampled digital array $x_{base}[n]$, and the sampling rate $f_{sampling}$.

(Optional) Task 6: Consult with TA to figure out how USRP or Sidekiq completes the remaining block. Note: different hardware may have different ways of completing the remaining block. USRP/Sidekiq is just one way to implement this block. However, almost all hardware implementation will take the three inputs f_c, $x_{base}[n]$, and $f_{sampling}$ as specified in this document. Please write down a detailed block diagram for this remaining block. There may be some components for which the details are beyond your understanding. However, you should at least figure out the high-level gist of each of the components. What I am expecting from the students is that a student should know "which component he/she does not understand", and for the component he/she does understand, he/she should be able to clearly explain it to his/her classmates, TA, or the instructor. Again, please consult with TA for this task.

To complete this task, each team needs to turn in a 3-page report. The report will be peer-reviewed. You will be asked to evaluate other students' reports based on clarity and correctness. The TA and the instructor will also take a look at the report to decide the grade for this task. We only have 3 pages, so please focus on the technical content (describing the block diagram in details), rather than background introduction.

This concludes our discussion on solving Challenge 2. Next time, we will discuss what happens when we would like to combine the block diagrams of Challenge 1 and Challenge 2.