Facultad de Ciencias

Sistemas de posicionamiento empleando familias de secuencias binarias de baja correlación

Positioning systems using families of binary sequences with low correlation

Trabajo de fin de máster
para acceder al

Máster EN MATEMÁTICAS Y COMPUTACION

Autor: Silvia Luengo
Codirectora: Ana Isabel Gómez Pérez
Codirector: Domingo Gómez Pérez

19 de julio de 2021

Agradecimientos

I would like to thank Domingo, Anabel and Andrew for their constant willingness to help me. I am also really grateful to TST for supporting me in this project and giving me this opportunity. I can't forget to thank my family and friends for unconditionally encouraging me.

Sistemas de posicionamiento empleando familias de secuencias binarias de baja correlación

Resumen

palabras clave: sistemas embebidos, secuencias binarias, sistemas posicionamiento interior, ultrasonido

El objetivo de este trabajo es el estudio de la aplicación de familias de secuencias binarias de baja correlación para su uso en sistemas de posicionamiento en tiempo real en interiores como por ejemplo en entornos industriales o de almacenamiento. Actualmente es un problemas abierto para el que se han propuesto distintas tecnologías como sistemas basados en visión artificial o en redes de sensores entre otros.

En este proyecto se ha implementado un sistema de posicionamiento en interiores de bajos recursos por medio de secuencias binarias de baja correlación. La investigación se ha centrado en la revisión de las tecnologías existentes en el mercado, la búsqueda de las secuencias binarias más apropiadas y el estudio de sus propiedades. Siguiendo el modelo GPS como sistema de localización en exteriores, se ha construido un propotipo basado en placas Arduino. Nuestra propuesta codifica la información mediante secuencias Pseudo Noise, códigos Gold y Kasami. Posteriormente estas secuencias son transmitidas utilizando señales de ultrasonido. En el receptor, las señales recibidas se pueden procesar para obtener medidas como la distancia entre dispositivos y el ángulo de llegada entre otras.

Positioning systems using families of binary sequences with low correlation

Abstract

keywords: embedded systems, binary sequences, indoor positioning system, ultrasonic

The aim of this project is the study of families of binary sequences of low correlation and its application to real-time indoor positioning systems in industrial or warehousing environments. Many different approaches based on different technologies such as artificial vision or sensor networks have been proposed for indoor localization but it still remains an open problem.

In this work, we have implemented a low resources indoor positioning system over over a embedded system, that uses binary sequences of low correlation. The research has focused on existent technologies in the market, on the search of the most appropriate family of sequences and the study of their properties. Taking GPS as a reference model for outdoor localization, we have built a prototype based on Arduino boards. Our approach encodes messages with Pseudo Noise sequences, Gold and Kasami Codes. Afterwards, the sequences are transmitted as ultrasonic signals. Then, the receiver processes the incoming signal to obtain measures such as the distances between devices and the angle of arrival of the signal.

Contents

1 Introduction **1**
- 1.1 Understanding GPS . . . 2
- 1.2 Indoor Positioning Systems . . . 3

2 Pseudo-Random Noise Sequences **7**
- 2.1 Preliminaries . . . 7
  - 2.1.1 Periodic sequences . . . 7
  - 2.1.2 Irreducible polynomials and finite fields . . . 9
- 2.2 Binary shift-register sequences (LFSR) . . . 10
- 2.3 Important families of binary sequences . . . 17
  - 2.3.1 Gold Sequences . . . 17
  - 2.3.2 Kasami sequences . . . 19
- 2.4 Sonar sequences . . . 19
  - 2.4.1 Construction of sonar sequences from m-sequences . . . 21
  - 2.4.2 Families of sequences with low auto and cross-hits . . . 21

3 Positioning techniques **23**
- 3.1 Location sensing . . . 23
- 3.2 Measurement methods . . . 24
  - 3.2.1 Time of Arrival (TOA) . . . 24
  - 3.2.2 Time Difference of Arrival (TDOA) . . . 25
  - 3.2.3 Angle of Arrival (AOA) . . . 26
  - 3.2.4 Received Signal Strength (RSS) . . . 26
- 3.3 Positioning algorithms . . . 27
  - 3.3.1 Trilateration . . . 27
  - 3.3.2 Multilateration . . . 29
  - 3.3.3 Triangulation . . . 29
  - 3.3.4 Fingerprinting . . . 30

4 Signal Modulation and Communication Protocols **31**
- 4.1 Signal Modulation . . . 31
  - 4.1.1 Pulse Code Modulation . . . 31
  - 4.1.2 Pulse Density Modulation . . . 32
  - 4.1.3 Pulse Position Modulation . . . 33
  - 4.1.4 DSSS . . . 34
- 4.2 Short Distance Communication Protocols . . . 34
  - 4.2.1 Two's Complement Representation . . . 35
  - 4.2.2 I^2S protocol . . . 35

5 Set-up **37**
5.1 Components 37
5.1.1 Arduino 37
5.1.2 Ultrasonic sensor 38
5.1.3 Analog-to-digital Converter 39
5.1.4 Oscilloscope 40
5.2 Final Prototype 41
5.3 Conclusions and future work 43

A Code **45**

Bibliography **57**

1 Introduction

Since the evolution of smartphones, Global Positioning Systems (GPS) has become an essential tool in our lives. Apps which implement it such as the world-known Google Maps are currently among the most used ones. This substitute for traditional paper maps identifies our location in a matter of seconds and makes it easy for humans to move from one place to another establishing a navigational path between both points.

This can be very useful also in closed spaces of big dimensions such as shopping centers, airports, underground parking lots, hospitals or supermarkets to easily find people, places (shops, gates, rooms, corridors) or objects. Unfortunately, GPS is not appropriate for inner spaces mainly due to two facts:

- Indoors structures produce high attenuation of the GPS signals, which are very likely to be reflected by walls and other obstacles. There is Non-Line of Sight conditions, and there are fast temporal changes due to, for example, people movement and opening doors. As a result, satellite signals cannot be received properly.
- Higher precision is required. GPS can reach at most $5m$-$10m$ accuracy, which is way less than what is expected for an indoor system.

These reasons, together with the recent development of Internet of Things (IOT), have raised the interest of designing an equivalent technology for indoor environments. This is how Indoor positioning systems (IPS) are born. Multiple technologies have been proposed to address this problem, just as an standalone product or as a combination of several solutions, that can be divided into three broad categories depending on the physical principle that supports them:

- Inertial navigation is a self contained technique based mostly on accelerometers and gyroscopes.
- Electromagnetic waves can use the visible, infrared, microwave or radio spectrum.
- Mechanical waves are mostly based in audible or ultrasound.

In the last few years, there has been some progress in the search of an analogous technology with a usability similar to GPS outdoors, but unlike GPS, there is no standard established yet. Some of the existing proposals include technologies based on Wireless Fidelity (WiFi), Bluetooth, Infrared, Radio Frequency (RF), Ultra Wide Band (UWB) and Ultrasound.

In addition to person and asset tracking, their countless applications include security, visitor guidance, commercial applications, workplace safety or surveillance. For instance, this tool can be decisive in accident situations such as fire where rescue teams must act very quickly. For example, if the location of the victims inside the building is known, vital time could be saved. Also in hospitals it can be extremely helpful for tracking patients or even equipment that is needed for emergencies. Not only this, but IPS can have customer-oriented commercial uses such as finding a desired product in a supermarket or business-oriented uses such as analyzing which products are the most attractive. In the industrial field, the motion of autonomous systems such as robots can be improved since they can be guided to follow a certain route or avoid obstacles. Intelligent worker protection and collision avoidance can be achieved using information provided by the positioning system, as well as workers

can be navigated to new assignments.

What all the approaches have in common is that they have taken GPS as the reference model and they have adapted it. Therefore, in the next section we will provide a brief recap of this technology.

1.1 Understanding GPS

Shortly presented above, GPS was designed in the 1970s by the United States Department of Defense (DoD) to determine the coordinates of any object on earth. For this, a large infrastructure was developed consisting of 32 satellites whose position with respect to earth changes over time. At least 24 of them are active at any given time, which means that they are continuously transmitting data on two frequencies, L1 (1575.42 MHz) and L2 (1227.60 MHz).

GPS provides two different positioning services: the Precise Positioning Service (PPS) and the Standard Positioning Service (SPS). The first of these is exclusively for military purposes, so its access is limited to US Armed Forces, US Federal agencies, selected allied armed forces and governments. It is designed for higher positioning accuracy. The SPS, instead, is intended for civilian uses, thus it is freely available to any user at any time and place. In PPS the satellites transmit a signal through both the L1 and L2 frequencies whereas in SPS only the L1 frequency is used.

The fundamental idea behind GPS is trilateration, a positioning technique which can determine the location of an item knowing its distance from four different reference points and their positions. The purpose of the satellites, that are constantly transmitting a signal, is precisely that the target (the item to be tracked) possesses the above data so it can infer its position by performing the required calculations.

Therefore, each satellite generates their own navigation message which contains the following information in the instant that the signal left:

- The time on the satellite.
- The position of the moving satellite.
- The state of the satellite.
- Additional information about the other satellites and about the atmosphere.

All these details result in a message of 37500 encoded bits.

As soon as the target receives and processes the signal, it calculates the delay taken by the signal to arrive (Time of Arrival-TOA). For this reason, this system relies on the fact that the clocks of the satellites and the target are all synchronized. The previous measure (TOA) enables the device to compute the distance to each satellite and locate itself in an intersection of spheres centered at the satellite positions and with radius equal to the computed distance.

Another aspect to take into account is that all the satellites of the framework are sending a signal simultaneously through the same channel. This is made possible by using codes to separate each signal and requires a wide frequency range, which receives the name of Code Division Multiple Access (CDMA). GPS employs DSSS or DS-CDMA in which the message is scrambled and spread using a sequence with adequate properties. For smooth communication without interference, each satellite encodes its navegation message using a different pseudorandom noise code (PRN) as key. GPS uses two kinds of PRNs, C/A(Coarse Acquisition) codes and the P(Precision) code. The C/A codes, which are used for the SPS, are Gold Codes. On the other hand, the P code is used for the PPS and it is a

much longer code.

At operation, the satellites broadcast pilot signals. These pilot signals transmit the PRN codes associated with each satellite, used to synchronize and track the locally generated PRN codes for despreading. At the receivers, the PRN code is derived by a code generator, that usually is based on linear feedback shift registers. After the code is obtained, the receiver can find the time shift that gives the maximum correlation with the satellite PRN code (See Figure 1).

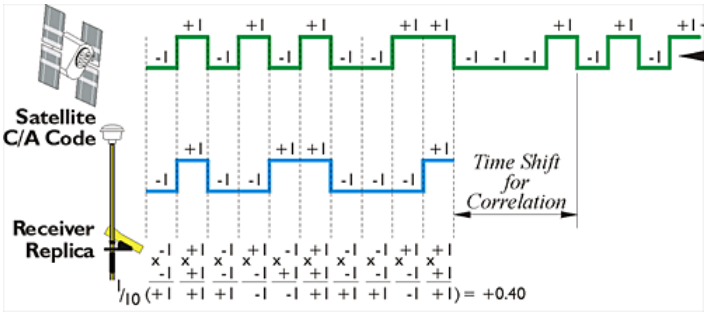


Figure 1: Code correlation performed by GPS signals to determine signal travel time and the distance to an unknown ground position. Source:Van Sickle [2008]

The theoretical background required to perform the steps required for GPS to work will be further explained in the following chapters.

1.2 Indoor Positioning Systems

A large number of technologies have been proposed for indoor positioning systems (IPS). Some of the most popular ones are:

- WiFi-Based Systems.
- Ultra Wide-band (UWB) Systems.
- Acoustic Systems (based on ultrasound).
- Optical Systems (based on infrared).

Technology	Typical Accuracy	Typical Coverage (m)	Typical Measuring Principle
Cameras	[mm-dm]	1-10	angle measurements from images
Infrared	[cm-m]	1-5	thermal imaging, active beacons
Sound	[cm]	2-10	distances from time of arrival
WLAN / WiFi	[m]	20-50	fingerprinting
RFID	dm-m	1-50	proximity detection, fingerprinting
Ultra-Wideband	cm-m	1-50	body reflection, time of arrival
Other Radio Frequencies	m	10-1000	fingerprinting, proximity
Inertial Navigation	1%	10-100	dead reckoning

Table 1: Performance parameters for popular indoor positioning technologies based in Table 1.1 in Mautz [2012]

In Table 1 we provide a summary of some of the principal characteristics of several technologies and the measuring principle that is based on.

In general, the most accurate IPS models are based either in ultrasound or UWB signals and have gathered the most interest (See Ruiz and Granja [2017]). The performance of both systems is very similar, being the main difference between them the emitting frequency. The ultrasonic signal is transmitted using a carrier frequency of 40 kHz, whereas UWB signal uses 500 MHz.

UWB is the least susceptible to interference but it needs more infrastructure which tends to be quite expensive. It can reach accuracies of order of tens of centimeters. On the other hand, ultrasonic systems are low-priced and achieve accurate sub-centimeter results. However, ultrasound can not penetrate solid walls and the signal can be affected by reflections, obstructions (Non- Line of Sight) or interferences. Ultrasonic devices have a maximum range of $10m$, which is a bit limited in comparison with the maximum range of $100m$ with Line of Sight (LOS) of UWB.
With all this in mind, we have decided to approach the indoor positioning problem using ultrasound technology. Previous work has been done in this area by The Massachusetts Institute of Technology (MIT) that also implemented a IPS based in ultrasound in one of their projects called Cricket Priyantha [2005].

All the devices in the structure of Cricket (both target and auxiliary devices) are Arduino boards composed of an RF transmitter, an ultrasound transmitter and a microcontroller. Access Points or beacons are deployed on ceilings and walls. Nodes can produce location information when they receive an RF signal from the beacons emitting an ultrasonic signal moments later. In the same way as GPS, the target is the receiver of the signals, hence behaving as a passive system. Its position is again deduced as a result of trilateration after measuring the different times of arrival. All nodes are necessarily synchronized, something accomplished by first sending a radio frequency signal (RF) which tells the Cricket devices the starting point for timing and other parameters such as ID, while the ultrasonic signal carries only a pulse.However the rate of interference raises greatly as more nodes are added to the system (See Wang and Han [2009]), this problem can be increased if the nodes adopt an active mode instead of a passive mode. The advantages of the passive mode include being able to contain as many nodes as possible and give privacy to the listeners as the location is only known to them. On the contrary active nodes improve the response time and do not require central monitoring, thus working better in dynamic target tracing for example. In Balakrishnan et al. [2003], a hybrid approach is proposed in which nodes can transmit signals to communicate their position under certain conditions of operations, while beacons remain active at all times.

This system can achieve an accuracy of $1 - 2cm$ in an indoor area of 10 m^3 with 24 devices. To overcome some of these limitations, Dolphin was proposed in Fukuju et al. [2003] that uses the broadband technique instead of the narrowband technique used by other ultrasound positioning systems and performs better in the presence of noise and multipath effect.

Spread spectrum techniques are considered robust against jamming attacks, have low probability of interception and detection and allow multiple users to share a common channel by code-division multiple access (CDMA) using a set of pre-shared keys for generation of employed pseudo-random noise (PNR) sequences. We present an approach that uses pulse position modulation (PPM) to transmit our chosen sequences instead of using a simple train of pulses over a ultrasonic signal. This allows for each node to be identified by a sequence. The number of simultaneous devices is a trade-off with the refresh rate, as the increase of devices will require to increase the length of the PNR sequences.

In the second chapter of this project, the pseudo-random noise sequences and the mathematical notions behind them are presented. In the third chapter, the principal positioning methods applied in positioning systems are explained. This is followed by a fourth chapter where a technical description

of aspects related to signal transmission is given. Finally, in the last chapter, our prototype and the results obtained are discussed as well as open questions and future work.

2 Pseudo-Random Noise Sequences

In this chapter we introduce the mathematical theory behind pseudo-random noise sequences and its generation. Of great importance in communication theory, pseudo-random noise sequences are deterministic sequences which solve many practical problems as synchronization, ranging and sharing channel among several users. As pseudo-random noise sequences might indicate, these deterministic sequences behave somehow as noise. This similarity, reflected in some properties of the PRN sequences, makes it possible to establish simultaneous communications on the same channel in the presence of noise and interference.

2.1 Preliminaries

This section is an introduction to sequences. A sequence is just a list of digits where all the digits are taken from a finite set, called the alphabet. Although the sequences are infinite, we focus in the case where the sequences repeat after a certain number of digits, which it is the most interesting case for application. Also, most of the chapter studies sequences where the alphabet consists on two element, the so-called binary sequences.

2.1.1 Periodic sequences

Definition 1. *An infinite sequence $u = u_0u_1u_2...$ is periodic if $\exists N$ such that $u_i = u_{i+N}$, $\forall i \geqslant 0$. The smallest N verifying the previous condition is defined as the period of the sequence. In such case, u can be expressed as $\overline{u_0u_1u_2\dots u_{N-1}}$.*

Definition 2. *Let $u = \overline{u_0u_1u_2\dots u_{N-1}}$, we define T as the left shift operator.*

- $T^0u = u$
- $Tu = \overline{u_1u_2\dots u_{N-1}u_0}$
- $T^ku = \overline{u_k\dots u_{k-1}}$
- $T^Nu = u$

T^ku is called a phase of u.

The symbols in the alphabet of binary sequences are usually -1 and 1, this allows to introduce the concept of balance.

Definition 3. *The balance of $u = \overline{u_0u_1u_2\dots u_{N-1}}$ can be calculated as the sum of its N first digits,*

$$Balance = \sum_{i=0}^{N-1} u_i$$

Definition 4. *Let u, v be two binary sequences of period N.*

i) The periodic cross-correlation function is defined as $\theta_{u,v}(\ell) = \langle u, T^\ell v\rangle = \sum_{i=0}^{N-1} u_i \cdot v_{i+\ell}, \quad \ell \in \mathbb{Z}$

ii) The periodic autocorrelation function is $\theta_u(\ell) = \theta_{u,u}(\ell), \quad \ell \in \mathbb{Z}$

iii) u *and* v *are said to be uncorrelated if the absolute value of* $\theta_{u,v}(\ell)$ *is zero or* 1 *for all* ℓ.

The correlation is used in statistics as a measure between random variables to show linear dependence. The definition presented above is the discrete version for sequences. Notice that the minimum absolute value of the correlation function depends on the period of the sequence. For $\{1, -1\}-$sequences, if the period is even, then the minimum absolute value is 0, otherwise it is 1.

Remark. *By the periodicity of sequence* v*, for each* $\ell \in \mathbb{Z}$ *it holds that* $\theta_{u,v}(\ell) = \theta_{u,v}(\ell + N)$.

Definition 5. *Let* u *be a binary periodic sequence. A run of* u *is a subsequence of* u *consisting of consecutive digits. In particular, if* $u_i \in \{0, 1\}$ *a run of 0's is called a gap whereas a run of 1's a block.*

In his first book, Golomb [1967] was the first to suggest some properties or conditions that a deterministic sequence should meet to look random.

Definition 6. ***Golomb's Randomness Postulates for binary sequences:***

- ***Balanceness:*** *Both symbols must appear along the sequence with the same probability. This property will hold as long as the amounts of both symbols differ by at most 1.*
- ***Distribution of the runs:*** *At least half of the runs have length 1, at least one fourth have length 2, one eight have lenght 3 and so on, as long as these fractions give integral numbers of runs. In general, there are at least* $\frac{1}{2^k}$ *runs of length* k *out of the total number of runs. Furthermore, there are almost the same number of runs of both symbols.*
- ***Ideal autocorrelation***[1]***:*** *Computation of the correlation between a sequence and one of its phases must not give any information about the delay. Thus, the autocorrelation function is two-valued:*

$$\exists K \text{ such that } \theta_u(\ell) = \begin{cases} N, \text{ if } \ell \equiv N \\ K, \text{ if } 1 \leqslant \ell \leqslant N-1 \end{cases}$$

If a sequence verifies these conditions it is said to be a pseudo-random noise sequence or PRN-sequence.

For applications, it is necessary to have a set of binary sequences, also called a family of binary sequences. The design of this family should meet certain criteria, in particular the auto and cross-correlation of the sequences of the family should be as low as possible. Furthermore, we are interested in families of sequences as large as possible. Intuitively, the objectives of having small cross correlation and having a large family are conflicting. The next theorem relates size of the family, length of the members and the maximum cross-correlation θ^2_{max} of the family. We present the bound for a set of sequences.

Theorem 1 (Welch bound)**.** *Given a set of* k *sequences of period* N *with* θ_{max} *being the maximal cross-correlation, then*

$$(\theta_{max})^2 \geqslant \frac{1}{k(k-1)} \left[\frac{(\sum_{i=1}^{k} \theta_{i,i})^2}{N} - \sum_{i=1}^{k} \theta^2_{i,i} \right]$$

Proof. Let $S = \{u^i = u^i_0 \dots u^i_{N-1} \mid 1 \leqslant i \leqslant k\}$ be the set of sequences of period N.
To simplify the notation, let $\theta_{i,j} = \theta_{u^i,u^j}(0)$ and $\theta_{max} = \max_{i \neq j} |\theta_{i,j}|$. Then,

$$\sum_{1 \leqslant i,j \leqslant k} \theta^2_{i,j} = \sum_{1 \leqslant i \neq j \leqslant k} \theta^2_{i,j} + \sum_{1 \leqslant i \leqslant k} \theta^2_{i,i} \leqslant k(k-1)\theta^2_{max} + \sum_{1 \leqslant i \leqslant k} \theta^2_{i,i} \tag{1}$$

[1] In some references, this postulate asserts that the function # of agreements − # of disagreements is two-valued. Both statements are equivalent.

On the other hand,

$$\sum_{1\leqslant i,j\leqslant k}\theta_{i,j}^2=\sum_{1\leqslant i,j\leqslant k}(\sum_{0\leqslant t\leqslant N-1}u_t^i\cdot u_t^j)(\sum_{0\leqslant t\leqslant N-1}u_t^i\cdot u_t^j)=\sum_{1\leqslant i,j\leqslant k}\sum_{0\leqslant t,s\leqslant N-1}u_t^i\cdot u_t^j\cdot u_s^i\cdot u_s^j$$

Interchanging orders of summation,

$$\sum_{1\leqslant i,j\leqslant k}\theta_{i,j}^2=\sum_{0\leqslant t,s\leqslant N-1}\sum_{1\leqslant i,j\leqslant k}u_t^i\cdot u_s^i\cdot u_t^j\cdot u_s^j=\sum_{0\leqslant t,s\leqslant N-1}(\sum_{1\leqslant i\leqslant k}u_t^i\cdot u_s^i)^2$$

Applying Cauchy's inequality,

$$\sum_{1\leqslant i,j\leqslant k}\theta_{i,j}^2\geqslant\frac{\left(\sum\limits_{t=0}^{N-1}\sum\limits_{i=1}^{k}(u_t^i)^2\right)^2}{N}=\frac{\left(\sum\limits_{i=1}^{k}\sum\limits_{t=0}^{N-1}(u_t^i)^2\right)^2}{N}=\frac{\left(\sum\limits_{i=1}^{k}\theta_{i,i}\right)^2}{N}\tag{2}$$

Combining (1) and (2) we obtain

$$\frac{(\sum\limits_{i=1}^{k}\theta_{i,i})^2}{N}\leqslant k(k-1)\theta_{max}^2+\sum_{1\leqslant i\leqslant k}\theta_{i,i}^2$$

which leads to the desired result. □

Definition 7. *Given a periodic sequence u and a positive integer q, one can generate a new sequence $v=u[q]$ by taking elements separated by q positions of u:*

$$v_i=u_{qi}\quad\forall i\in\mathbb{Z}$$

The sequence v is a decimation by q of u, and will be denoted by $u[q]$.

Proposition 1. *If u is a sequence of period N and $v=u[q]$, then v is periodic with period $\frac{N}{gcd(N,q)}$. Consequently, v has period N if and only if N and q are coprime. In this case, v is called a proper decimation.*

Proof. With no loss of generality, one can assume that $q\leqslant N$ since $u[q+l\cdot N]=u[q]$.
The sequence v has period r iff $v=\overline{u_0u_qu_{2q}\dots u_{(r-1)q}}\implies u_{rq}=u_0$ and $rq=lcm(q,N)$

$$r=\frac{lcm(q,N)}{q}=\frac{N}{gcd(q,N)}$$

This finishes the proof. □

2.1.2 Irreducible polynomials and finite fields

This subsection is devoted to an introduction to maximal length sequences, which are constructed from finite fields. We suppose that the reader is familiar with the field of integers modulo a prime p and also polynomial rings over fields.

Definition 8. *Let $f(x)=c_0+c_1x+\dots+c_nx^n$ be a polynomial of degree n over a field $\mathbb{K}$, its reciprocal polynomial is defined as $\hat{f}(x)=x^nf(\frac{1}{x})=c_n+c_{n-1}x+\dots+c_0x^n$.*

Proposition 2. *$f(x)$ is irreducible iff so is $\hat{f}(x)$.*

Proof. Notice it is enough to prove the first direction of the theorem.
By assumption f is irreducible, so it is different from a monomial, i.e. $f(x)\neq c_nx^n\implies deg(\hat{f})\geqslant 1$.
Now, by proving of contradiction, assume that $\hat{f}(x)=g(x)h(x)$, with $deg(g),deg(h)\geqslant 1$.

$$f(x) = x^n \hat{f}(\tfrac{1}{x}) = x^n g(\tfrac{1}{x}) h(\tfrac{1}{x}) = x^{n-deg(\hat{f})} \underbrace{\left(x^{deg(g)} g\left(\frac{1}{x}\right)\right)}_{r(x)} \underbrace{\left(x^{deg(h)} h\left(\frac{1}{x}\right)\right)}_{s(x)}$$

Since $f(\frac{1}{x}), g(\frac{1}{x}) \neq 0$, $deg(r), deg(s) \geqslant 1$, which contradicts the irreducibility of f. □

Definition 9. *If $f(x)$ is an irreducible polynomial of degree n in $\mathbb{F}_p[x]$, we define*

$$\mathbb{F}_{p^n} = \frac{\mathbb{F}_p[x]}{(f(x))}$$

as the finite field of p^n elements. This field is unique up to isomorphism.

The following results refer to the field $\mathbb{F}_{2^n}$, but can be extended to $\mathbb{F}_{p^n}$ for any prime p. The proofs will be omitted but can be found in Golomb and Gong [2005].

Definition 10. *An element $\alpha \in \mathbb{F}_{2^n}$ is primitive if it is a generator of the multiplicative group of the field, i.e. any nonzero $\beta \in \mathbb{F}_{2^n}$ can be represented as*

$$\beta = \alpha^i \text{ for some } i \in \{1, ..., 2^n - 1\}$$

Proposition 3. *For any elements $\beta, \gamma \in \mathbb{F}_{2^n}$ and $k \in \mathbb{N}$,*

i) $\beta^{2^n} = \beta$

ii) $(\beta + \gamma)^{2^k} = \beta^{2^k} + \gamma^{2^k}$

Definition 11. *Let $\beta \in \mathbb{F}_{2^n}$, we define its trace as:*

$$\mathrm{Tr}_{\mathbb{F}_{2^n}}(\beta) = \sum_{i=0}^{n-1} \beta^{2^i}$$

Proposition 4. *If β, $\gamma \in \mathbb{F}_{2^n}$ the following properties hold:*

i) $\mathrm{Tr}_{\mathbb{F}_{2^n}}(\beta) = \mathrm{Tr}_{\mathbb{F}_{2^n}}(\beta^2)$

ii) $\mathrm{Tr}_{\mathbb{F}_{2^n}}(\beta + \gamma) = \mathrm{Tr}_{\mathbb{F}_{2^n}}(\beta) + \mathrm{Tr}_{\mathbb{F}_{2^n}}(\gamma)$

iii) $\sum_{\beta \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\gamma \cdot \beta)} = \begin{cases} 2^n, \text{ if } \gamma = 0 \\ 0, \text{ otherwise} \end{cases}$

Proposition 5. *Let e be a divisor of n, then there exists a subfield $\mathbb{F}_{2^e} \subset \mathbb{F}_{2^n}$ of 2^e elements. Evenmore, $\beta \in \mathbb{F}_{2^e} \iff \beta^{2^e} = \beta$.*

2.2 Binary shift-register sequences (LFSR)

One of the simplest and most commonly used methods of producing pseudo-random sequences are Feedback Shift Registers.

Definition 12. *A n-stage feedback shift register is a circuit whose register has n cells, each one storing one bit. The circuit is controlled by a clock, in such a way that at every clock period, the state of the register changes as the values of the cells are shifted one position. The bit of the last cell comes as output, whereas the first one is the result of applying a function on the previous state.*

The initial state of the register is called the seed and the cells that affect the next state are called taps.

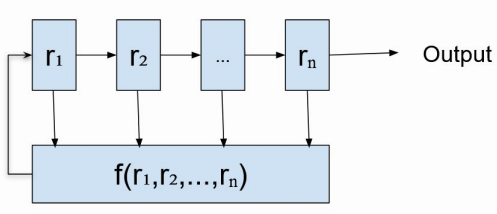


Figure 2: FSR

Linear feedback shift registers (LFSR) are those circuits whose feedback function is linear.

Remarks.

i) Given an initial state, the output of a LFSR generates a binary periodic sequence. If the seed is $\underbrace{0\dots0}_{n}$*, the sequence generated is the all-zeros sequence.*

ii) In any n-stage LFSR, there are 2^n *possible states for the register. Thus, the maximum period of a sequence generated by it is* $N = 2^n - 1$*. Sequences of maximal period are called m-sequences.*

iii) We will choose the exclusive-or as the linear function. This way, if $r_i(t)$ *represents the bit of cell* i *at the instant* t*, the recurrence relation can be expressed in the following manner:*

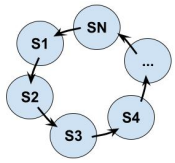
$$\begin{cases} r_1(t+1) = \sum_{i=1}^{n} c_i \cdot r_i(t) \ mod\ 2 \\ r_i(t+1) = r_{i-1}(t), \quad \forall i \in \{2,..,n\} \end{cases}$$

where $c_i = 1$ *if cell* i *is a tap and 0, otherwise. By definition,* $c_n = 1$*. Otherwise, the* n*th cell could be removed having a* $(n-1)$*-stage LFSR with the same output sequence.*

Proposition 6. *If* u *is a m-sequence generated by a LFSR with initial seed* s*, the sequence obtained starting with any other non-trivial state* s' *is a shifted-version of* u*.*

Proof. Since u is a m-sequence, the register takes all possible non-zero states exactly once starting from s. This includes state s', say after i iterations.

$$\underbrace{s_1}_{s} \to s_2 \to s_3 \to \dots \to \underbrace{s_{i+1}}_{s'} \to \dots \to s_N \implies s_1$$



Let v be the sequence obtained taking s' as the initial state. Clearly, v has the same period as u and in fact, $v = T^i u$. □

Definition 13. *Given a n-stage LFSR, we define the following polynomials:*

- *The characteristic polynomial is*

$$f(x) = c_0 + c_1 x + \dots + c_{n-1}x^{n-1} + c_n x^n = \sum_{i=0}^{n} c_i x^i \in \mathbb{F}_2[x]$$

where the coefficients c_i *take the same values as before and* $c_0 = 1$*.*

- *The minimal polynomial is the*

Proposition 7. *$f(x)$ is irreducible if and only if every sequence generated by the corresponding LFSR has the same period, independently of the seed.*

Notice this result is no longer true when $f(x)$ is reducible. Take for example $f(x) = 1+x+x^2+x^3$. It is reducible since $f(x) = (x+1)(x^2+1)$,

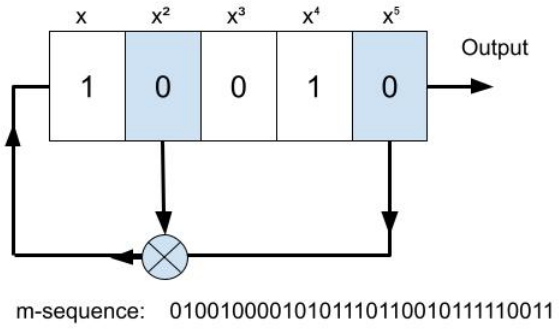
- If $s = 001$, the sequence obtained is $\overline{1001}$
- If $s = 101$, the sequence obtained is $\overline{10}$

Remark. *It can be proved that if the characteristic polynomial $f(x)$ is irreducible, then it is the non-zero polynomial of lowest degree satisfying*

$$\bar{f}(u) \coloneqq c_0 u + c_1 T u + \ldots + c_{n-1} T^{n-1} u + c_n T^n u = \sum_{i=0}^{n} c_i T^i u = \overline{0}$$

for every sequence u obtained from the LFSR.

Example:



The given LFSR has 2 taps. The characteristic polynomial is $x^5 + x^2 + 1$

The sequence generated has length 31

$x = \overline{0100100001010111011001011111001 1}$

Figure 3: 5-stage LFSR

Proposition 8. *If $f(x)$ is irreducible and $g(x) \in \mathbb{F}_2[x]$ is another polynomial verifying $\bar{g}(u) = 0$ for every sequence u obtained from the LFSR. Then $f(x) \mid g(x)$.*

Proof. Let $deg(g) = m > n$. By the Euclidean algorithm, $\exists q(x), r(x)$ with $deg(r) < n$ s.t

$$g(x) = f(x) \cdot q(x) + r(x) \implies \underbrace{\bar{g}(u)}_{\overline{0}} = \underbrace{\bar{f}(u) \cdot \bar{q}(u)}_{\overline{0}} + \bar{r}(u) \implies \bar{r}(u) = \overline{0}$$

Since $deg(r) < n$, necessarily, $r(x) = 0$. □

Corollary 1. *Let $f(x)$ be the characteristic polynomial of an n-stage LFSR. If $f(x)$ is irreducible, then there exists an integer N such that $f(x) \mid x^N + 1$.*

Proof. It suffices to show that the property above holds for some polynomial $g(x) = x^N + 1$.
Let N_S be the period of the sequence u_S obtained from the LFSR given S as initial seed. Considering $N = lcm\{N_S \mid S \in \mathbb{F}_2^n \backslash \{0\}\}$, $T^N u_S = u_S \implies T^N u_S + u_S = \overline{0}$, $\forall S \in \mathbb{F}_2^n \backslash \{0\}$. □

Definition 14. *Let $f(x)$ be an irreducible polynomial of degree $n > 1$ over $\mathbb{F}_2$. The order of $f(x)$ is the smallest integer N such that $f(x) \mid x^N + 1$. If $N = 2^n - 1$, then $f(x)$ is said to be a primitive polynomial.*

Theorem 2. *A LFSR generates a m-sequence if and only if its characteristic polynomial is a primitive polynomial.*

Proof.

- $\Rightarrow$ If a given n-stage LFSR generates a m-sequence, as shown in proposition 2, it does not depend on the chosen seed. Up to rotation, all the sequences resulting from this LFSR are the same, so they have the same period, $N = 2^n - 1$, which means the characteristic polynomial must be irreducible. Now, applying corollary 1, the order of the characteristic polynomial is N, so it is primitive.
- $\Leftarrow$ If the characteristic polynomial of the LFSR is primitive, it is irreducible and has order N. From corollary 1, we know the order is $lcm\{N_S \mid S \in \mathbb{F}_2^n \backslash \{0\}\}$, which necessarily means $p_S = N$ $\forall S \in \mathbb{F}_2^n \backslash \{0\}$, so all the output sequences are m-sequences.

□

Due to their importance, primitive polynomials have been widely studied. Giant lists or tables containing a great number of them have been published (see Živković [1994] as an example) and different algorithms have been proposed for their search.

Proposition 9. *A polynomial is primitive if and only if its reciprocal is primitive.*

Proof. Again, it suffices to prove the first implication, because the reciprocal of the reciprocal polynomial is the polynomial.
Assuming f is primitive, $\hat{f}$ is irreducible by proposition 2. Let's prove that $\hat{f}$ has order N

$$f \mid x^N+1 \implies x^N+1 = f(x)q(x) \implies \frac{1}{x^N}+1 = f\left(\frac{1}{x}\right)q\left(\frac{1}{x}\right) \implies x^n\left(\frac{1}{x^N}+1\right) = \underbrace{x^n f\left(\frac{1}{x}\right)}_{\hat{f}(x)} q\left(\frac{1}{x}\right)$$

Now, multiplying both sides of the equality by $x^{deg(q)} = x^{N-n}$,

$$x^{N-n}x^n\left(\frac{1}{x^N}+1\right) = \underbrace{x^n f\left(\frac{1}{x}\right)}_{\hat{f}(x)} \underbrace{x^{N-n}q\left(\frac{1}{x}\right)}_{\hat{q}(x)} \implies x^N+1 = \hat{f}(x)\hat{q}(x) \implies f \mid x^N+1$$

It remains to show that $\hat{f} \nmid x^k+1$ $\forall k < N$.
Assuming the contrary, i.e. $\exists k < N$ such that $\hat{f} \mid x^k+1$. Then, using the same argument as before, $f \mid x^k+1$, which contradicts that $f(x)$ has order N. □

Remark. *One can prove that, for every $n \geqslant 1$, there is a primitive polynomial of degree n. There are exactly,*

$$\frac{\phi(2^n-1)}{n}$$

degree-n primitive polynomials, where ϕ is the Euler phi-function. By the last proposition, this number must be an even number.

Corollary 2. *If a LFSR generates a maximum-length sequence, the number of taps is even.*

Proof. Let us assume the opposite.
If the number of taps is odd, the characteristic polynomial $f(x)$ has an even number of non-zero coefficients. Thus, $f(1) = 0$ and $x+1 \mid f(x)$, so it is not irreducible and therefore not primitive, which contradicts last theorem. □

Remark. *The characteristic polynomials of n-stage LFSRs with 2 taps are trinomials:*

$$1 + x^k + x^n$$

A lot of research has been done on primitive trinomials because having only two inputs for the XOR function, they are the most efficient computationally. For instance, Swan's theorem Swan [1962] states there are no irreducible trinomials of degrees that are multiple of 8.

Proposition 10. *Let u be an m-sequence of period N generated by an LFSR whose characteristic polynomial is $h(x)$ and $v = u[q]$. If v is a proper decimation, then it is also an m-sequence generated by a primitive polynomial. In particular, $v[N-1]$ is generated by $\hat{h}(x)$.*

For practical reasons, we will transform any $\{0,1\}$-sequence into a $\{1,-1\}$-sequence to compute correlation easily. This change can be done with the function $\chi(\alpha) = (-1)^\alpha$, which applies component-wise to the whole sequence.

From now on, we will be using $\theta_{u,v}$ to denote $\theta_{\chi(u),\chi(v)}$, which is an abuse of notation.

Remark. *With this notation, the Welch Bound can be simplified to the following expression:*

$$\theta_{max}^2 \geqslant \frac{Nk - N^2}{N-1}$$

Actually, given a set of k sequences of length N, we can consider a bigger set consisting of the k sequences and all their phases. In this case,

$$\theta_{max}^2 \geqslant \frac{kN^2 - N^2}{kN-1} \implies \theta_{max} \geqslant N\sqrt{\frac{k-1}{kN-1}}$$

Proposition 11. *If u is a m-sequence generated by an n-stage LFSR the following properties hold:*

i) The period of u is $N = 2^n - 1$

ii) Shift-and-add property: Given $i \neq j$ with $0 \leqslant i, j < N$, there exists a unique k, different from i, j such that

$$T^i u \oplus T^j u = T^k u$$

iii) The balance of the sequence is -1 and $\theta_u(\ell) = \begin{cases} N, & \text{if } \ell \equiv 0 \bmod N \\ -1, & \text{if } \ell \not\equiv 0 \bmod N \end{cases}$ *(G. Postulates I & III)*

Proof.

i) By definition.

ii) Shift-and-add property:

Trivially, the set $\{\overline{0}, u, Tu, \ldots, T^{2^n-2}u\}$ is a vector space of dimension n over $\mathbb{F}_2^N$. Since it is closed under addition, $\forall\ 0 \leqslant i \neq j < N\ \exists k$ s.t.

$$T^i u \oplus T^j u = T^k u$$

iii) Since u is a m-sequence the cell of the register has taken all possible values except for the all-zeros vector. That implies that u is formed by 2^{n-1} 1s and $2^{n-1} - 1$ 0s (Balanceness).

- If $\ell \equiv 0 \bmod N$, $\theta_u(\ell) = \sum_{i=0}^{N-1} (-1)^{u_i}(-1)^{u_{i+pN}} = \sum_{i=0}^{N-1} 1 = N.$
- If $\ell \not\equiv 0 \bmod N$, $\theta_u(\ell) = \sum_{i=0}^{N-1} (-1)^{u_i}(-1)^{u_{i+\ell}} = \sum_{i=0}^{N-1} (-1)^{u_{i+k}} = -1.$

□

Remark. *Autocorrelation peak is reached when there is no delay.*
As an example, the following graph represents the periodic autocorrelation function of the m-sequence generated by the primitive polynomial $x^6 + x^5 + x^3 + x^2 + 1$.

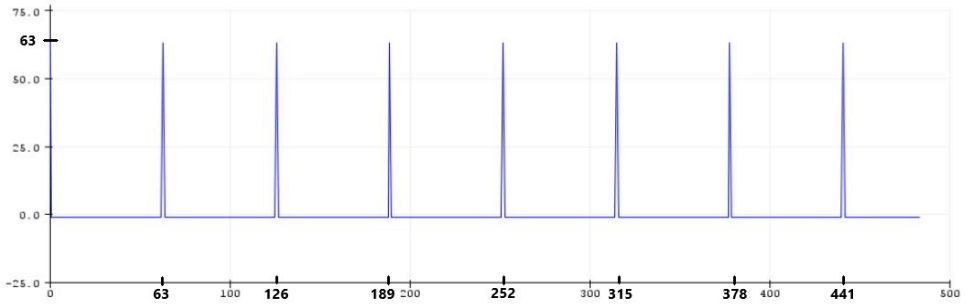


Figure 4: Autocorrelation of the binary m-sequence of length 63 for different phases. Notice that it is two-valued as given by the Golomb's postulate III.

Corollary 3. *All Golomb's Randomness Postulates hold for m-sequences.*

Proof. Postulates I and III have already been proved in the last proposition.
Let us prove postulate II for a $\{0,1\}$-m-sequence by first calculating the number of runs of length k, $\forall k$.

- If $k \leqslant n-2$, a run will happen if the last $k+2$ cells of the register are either

$$1\,\overbrace{0\ldots0}^{k}\,1 \text{ or } 0\,\overbrace{1\ldots1}^{k}\,0.$$

  Since the sequence has maximal period, all non-zero states will appear in the register and each of the previous subsequences will occur in $2^{n-(k+2)}$ states.

- The state $\overbrace{1\ldots1}^{n-1}\,0$ is always followed by $\overbrace{1\ldots1}^{n}$, which is followed by $0\,\overbrace{1\ldots1}^{n-1}$.
  Consequently, there are no blocks of length $n-1$, but there is one of length n.

- The state $\overbrace{0\ldots0}^{n-1}\,1$ is followed by $1\,\overbrace{0\ldots0}^{n-1}$, so there is one gap of length $n-1$.
  However, there are no gaps of length n as the zero-state never occurs.

Hence, the total number of runs is

$$2 + 2 \cdot \sum_{k=1}^{n-2} 2^{n-k-2} = 2 + 2(2^{n-2} - 1) = 2^{n-1}$$

For each k, there are $2^{n-k-1} = \frac{2^{n-1}}{2^k}$ runs. □

Proposition 12. *If* $u = \overline{u_0 \ldots u_{N-1}}$ *is a m-sequence of length* $N = 2^n - 1$*, there exists a primitive element* $\alpha \in \mathbb{F}_{2^n}$ *such that*

$$u_i = \mathrm{Tr}_{\mathbb{F}_{2^n}}(\alpha^i) \quad \forall i \in \{0, \ldots, N-1\}$$

Theorem 3. *Let* u *be a m-sequence of period* $2^n - 1$ *and* $v = u[q]$ *a decimation of* u*. If* $q = 2^k + 1$ *or* $q = 2^{2k} - 2^k + 1$ *for some* k *and* $e = gcd(n,k)$ *verifies* $\frac{n}{e}$ *is odd, then the spectrum of* $\theta_{u,v}$ *takes the following three values:*

$$\theta_{u,v}(\ell) = \begin{cases} -1 \\ -1 + 2^{\frac{n+e}{2}} \\ -1 - 2^{\frac{n+e}{2}} \end{cases}$$

Proof. The proof will be done for the case $q = 2^k + 1$ but it could be done in a similar way for $q = 2^{2k} - 2^k + 1$.
Property *iii*) from proposition 4 can be extended to the following result, let $\lambda \in \mathbb{F}_{2^n}$, the following sum takes only two values:

$$\sum_{x \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^e}}(\lambda x)} = 2^e \quad \text{or} \quad \sum_{x \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^e}}(\lambda x)} = 0.$$

The proof is trivial, squaring

$$\left(\sum_{x \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^e}}(\lambda x)} \right)^2 = \sum_{x \in \mathbb{F}_{2^e}} \sum_{y \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^e}}(\lambda(x+y))} = 2^e \sum_{x \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^e}}(\lambda(x))}$$

To calculate the correlation between u and v, we will calculate the square of the correlation function:

$$(\theta_{u,v}(\ell) + 1)^2 = \left(1 + \sum_{i=0}^{2^n - 1} (-1)^{u_i + v_{i+\ell}} \right)^2 = \left(\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda x + x^q)} \right)^2 \quad \text{where } \lambda = \alpha^{-\ell}.$$

By squaring and substituting $q = 2^k + 1$ we get

$$\begin{aligned} \left(\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda x + x^q)} \right)^2 &= \left(\sum_{x,w \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda x + x^q + \lambda(x+w) + (x+w)^q)} \right) \\ &= \left(\sum_{w \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda w + w^{2^k+1})} \left(\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(w x^{2^k} + x w^{2^k})} \right) \right) \\ &= \left(\sum_{w \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda w + w^{2^k+1})} \left(\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(x(w^{2^{n-k}} + w^{2^k}))} \right) \right). \end{aligned}$$

The inner sum is a sum of values of a trace, so the inner sum on w has only to be applied when $w^{2^{n-k}} = w^{2^k}$.

Notice that elements that satisfy $w^{2^{n-k}} = w^{2^k}$ are exactly $w^{2^{2k}} = w$. Consequently, $w^{2^{gcd(n,2k)}} = w$, in other words $w \in \mathbb{F}_{2^e}$, which means that

$$\left(\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda x + x^q)} \right)^2 = 2^n \left(\sum_{w \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda w + w^{2^k+1})} \right).$$

Notice that if $w \in \mathbb{F}_{2^e}$, then $w^{2^k+1} = w^2$ so $\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda w + w^{2^k+1}) = \mathrm{Tr}_{\mathbb{F}_{2^n}}(w(\lambda + 1))$, which can be substituted giving,

$$2^n \left(\sum_{w \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(\lambda w + w^{2^k+1})} \right) = 2^n \left(\sum_{w \in \mathbb{F}_{2^e}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}}(w(\lambda+1))} \right).$$

Putting all together,

$$(\theta_{u,v}(\ell) + 1)^2 = 2^{n+e} \quad \text{or} \quad 0,$$

which gives the result. □

In general, it is difficult to characterize the values of the correlation function. Knowing the values of the correlation of two m-sequences allows to define families of binary sequences of size N, as we will consider in the next section.

Definition 15. *If u and v satisfy the correlation properties of Theorem 3 they are called a preferred pair of m-sequences.*

Corollary 4. *If $n \not\equiv 0 \mod 4$, there exist preferred pairs of m-sequences in the strict sense where the crosscorrelation function takes the following three values:*

$$\theta_{u,v}(\ell) = \begin{cases} -1 \\ -t(n) \\ t(n) - 2 \end{cases}, \text{ where } t(n) = 1 + 2^{\lfloor \frac{n+2}{2} \rfloor}$$

Proof. The corollary is a direct application of Theorem 3 for different decimations. In all the cases, we try to find a value k such that $e = 1, 2$. If $n \not\equiv 0 \mod 4$, there are two different cases, either n is odd or $n \equiv 2 \mod 4$. Let us discuss them separately:

- If n is odd, the theorem holds for $q = 3, 5, 13$
  - $q = 3 = 2^1 + 1$
  - $q = 5 = 2^2 + 1 \implies e = 1, \frac{n}{e}$ is odd
  - $q = 13 = 2^{2 \cdot 2} - 2^2 + 1$
- If $n \equiv 2 \mod 4$, the theorem holds for $q = 5, 13$
  - $q = 5 = 2^2 + 1$
  - $q = 13 = 2^{2 \cdot 2} - 2^2 + 1$ $\implies e = 2, \frac{n}{e}$ is odd

This finishes the proof. □

In the literature, preferred pairs exists for any n as the case for four-valued correlation function is also included ($n \equiv 0 \mod 4$). However, the number of preferred pairs is still an open problem.

2.3 Important families of binary sequences

We are interested in sets of sequences with small cross-correlation. Usually more than two sequences are required since there are more than two transmitters in the same channel.

2.3.1 Gold Sequences

Definition 16. *Let u, v be a preferred pair of m-sequences of length $N = 2^n - 1$ we define the set of sequences $G(u, v)$ as*

$$G(u, v) = \{u, v, u \oplus v, u \oplus Tv, ..., u \oplus T^{N-1}v\}$$

$G(u, v)$ is a set of Gold sequences.

Remark. *Some of the sequences included in $G(u, v)$ are not m-sequences.*
As an example, let $u = \overline{1110100}$, the m-sequence generated by $x^3 + x + 1$ and $s = 111$, and $v = u[3] = \overline{1001011}$.

$$G(u, v) = \{u, v, \overline{0111111}, \overline{0010001}, \overline{0000110}, \overline{1001101}, \overline{0101000}, \overline{1011010}, \overline{1100011}\}$$

It is clear that balanceness does not hold.

Lemma 1. $\theta_{T^k u, T^j v}(\ell) = \theta_{u,v}(\ell + j - k)$

Proof. $\theta_{T^k u, T^j v}(\ell) = \sum_{i=0}^{N-1} (-1)^{u_{i+k}} (-1)^{v_{i+j+\ell}} = \sum_{i=0}^{N-1} (-1)^{u_i} (-1)^{v_{i+j+\ell-k}} = \theta_{u,v}(\ell + j - k)$ □

Proposition 13.

i) $\theta_{w,z}(\ell) \in \{-1, -t(n), t(n) - 2\}$ *for any* $w, z \in G(u, v)$, $w \neq z$, $\ell \in \mathbb{Z}$

ii) $\theta_w(\ell) \in \{-1, -t(n), t(n) - 2\}$ *for any* $w \in G(u, v)$, $\ell \not\equiv 0 \mod N$

Proof. Let $\ell, m, k \in \mathbb{Z}$,

$$\theta_{u \oplus T^m v, u \oplus T^k v}(\ell) = \sum_{i=0}^{N-1} (-1)^{u_i \oplus v_{i+m}} (-1)^{u_{i+\ell} \oplus v_{i+k+\ell}} = \sum_{i=0}^{N-1} (-1)^{(u_i \oplus u_{i+\ell}) \oplus (v_{i+m} \oplus v_{i+k+\ell})}.$$

By the shift and add property $\exists r, s$ such that

$$\theta_{u \oplus T^m v, u \oplus T^k v}(\ell) = \sum_{i=0}^{N-1} (-1)^{u_{i+r}} (-1)^{v_{i+s}} = \theta_{T^r u, T^s v}(0) = \theta_{u,v}(s - r) \in \{-1, -t(n), t(n) - 2\}.$$

The same argument applies for the case k, ℓ

$$\theta_{u, u \oplus T^k v}(\ell) \text{ and } \theta_{v, u \oplus T^k v}(\ell) \in \{-1, -t(n), t(n) - 2\}$$

This covers all the possible cases and finishes the proof. □

In the cases studied in Corollary 5, we have the families that are close to optimal with respect to the Welch bound.

Corollary 5. *If* $n \not\equiv 0 \mod 4$, *the maximum value of the correlation* $t(n)$ *is less than four times the Welch bound, i.e.* $(2^n - 1)\sqrt{\frac{2^n}{2^{2n}-2}}$.

Proof. If $n \not\equiv 0 \mod 4$ there exists a preferred pair of sequences, (u, v), which generates a set of Gold Codes $G(u, v)$, where

- $N = 2^n - 1$ is the length of the sequences
- $k = 2^n + 1$ is the size of the family

Now, applying the Welch bound,

$$max|\theta_{w,z}(\ell)| \geqslant N\sqrt{\frac{k-1}{Nk-1}} = (2^n - 1)\sqrt{\frac{2^n}{2^{2n}-2}},$$

and dividing $t(n)$ by this, we get

$$\frac{(1 + 2^{\lfloor \frac{n+2}{2} \rfloor}) \cdot \sqrt{2^{2n} - 2}}{(2^n - 1) \cdot 2^{n/2}} \leqslant \frac{(1 + 2^{\frac{n+2}{2}}) \cdot \sqrt{2^{2n} - 2}}{(2^n - 1) \cdot 2^{n/2}}.$$

When $n \implies \infty$, the function tends to 3, the derivative is negative if $n > 1$ and the maximum is less than four. This finishes the proof. □

Remark. *The C/A codes GPS (see Van Sickle [2008]) are preferred pairs defined by m-sequences* u, v*:*

- u *is the m-sequence generated by* $x^{10} + x^3 + 1$, *with seed* $s = \underbrace{1 \ldots 1}_{10}$
- v *is the m-sequence generated by* $x^{10} + x^9 + x^8 + x^6 + x^3 + x^2 + 1$, *with seed* $s = \underbrace{1 \ldots 1}_{10}$

With the help of some lines of code, it can be checked that v *is a shifted version of* $u[q]$ *for*

$$q = 17, 34, 65, 68, 130, 136, 260, 272, 520, 544.$$

2.3.2 Kasami sequences

Definition 17. *Let u be a m-sequence of length $N = 2^{2m} - 1$ and $v = u[2^m + 1]$ we can define the set of sequences $K_s(u)$ as*

$$K_s(u) = \{u, u \oplus v, u \oplus Tv, \ldots, u \oplus T^{2^m-2}v\}$$

$K_s(u)$ is a set of Kasami sequences.

Proposition 14. *Let $s(m) = 1 + 2^m$*

i) $\theta_{w,z}(\ell) \in \{-1, -s(m), s(m) - 2\}$ for any $w, z \in K_s(u)$, $w \neq z$, $\ell \in \mathbb{Z}$

ii) $\theta_w(\ell) \in \{-1, -s(m), s(m) - 2\}$ for any $w \in K_s(u)$, $\ell \not\equiv 0$ mod N

This proof will be omitted since it is very similar to the one given for Theorem 3. The maximum value of the correlation of the Kasami family is also very close to the optimal value given by the Welch bound.

Corollary 6. *The maximum value of the correlation of the Kasami family is optimal with respect of the Welch bound.*

Proof. Consider the set of Kasami sequences generated from a m-sequence of period $N = 2^{2m} - 1$ and its $\{2^m + 1\}$-th decimation. Then, $k = 2^m$ is the size of the family. By the Welch Bound,

$$max|\theta_{w,z}(\ell)| \geqslant N\sqrt{\frac{k-1}{Nk-1}} = (2^{2m}-1)\sqrt{\frac{2^m-1}{2^m(2^{2m}-1)-1}} >$$

$$(2^m - 1) \cdot (2^m + 1) \cdot \sqrt{\frac{2^m - 1}{2^m(2^{2m} - 1)}} = (2^m - 1) \cdot \sqrt{\frac{2^m + 1}{2^m}} > 2^m - 1.$$

Notice that the correlation has to be an odd integer because the length of the sequences is odd, so this gives the result. □

2.4 Sonar sequences

This section considers the case of sequences with low correlation but defined over bigger alphabets. Their use can be traced back to the first military prototypes of Sonar and Radar. These devices send a signal to detect the presence of objects, their distance and speed. When a signal is reflected from the target and received back by the observer, it is shifted in frequency and time. These shifts characterize the distance and speed of the object by the Doppler effect. Therefore, it was necessary to design sequences for which it was simple to recover both shifts easily.

Definition 18. *An $M \times N$ sonar sequence[2] is a sequence $a_1 a_2 \ldots a_N$ with $a_i \in \{0, \ldots, M-1\}$ which satisfies the synchronization property:*

$\forall k \in \{1, \ldots, N-1\}$, the list of differences $(a_{i+k} - a_i)$ with $i \in \{1, \ldots, N-k\}$ contains distinct entries.

Originally, the set $\{0, \ldots, M-1\}$ was designed as a set of frequencies and the research focused on $N = M$. Despite the efforts to design sequences with the synchronization property from the Engineering and Mathematical community, there are few known algebraic constructions. These constructions have some restrictions on the factorization of N, see Golomb and Taylor [1984] for a historical review as well as most known methods to generate Costas sequences.

[2] Some authors refer to sonar sequences as Costas sequences since they were first introduced by John Costas in 1975.

Remark. *A sonar sequence can be represented as an M by N sonar array (with rows and columns numbered 0 to $M-1$ and 1 to N, respectively) in which column i has a single dot in row a_i.*

Definition 19. *Let A and B be two $M \times N$ sonar arrays. Their cross-hit is defined as the number of hits between them. This definition can be generalized to any translation (vertically and/or horizontally) of B giving rise to $C_{A,B}(\tau, d)$, the number of hits between A and a version of B which has been shifted to the right by τ and up by d. If $A = B$, we will rather refer to cross-hit as auto-hit.*

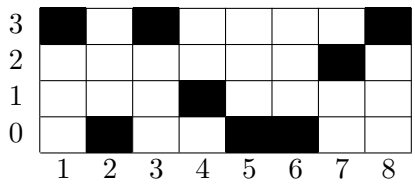
Proposition 15. *The synchronization property of a sonar sequence is equivalent to the fact that its corresponding sonar array agrees with any horizontal and/or vertical copy in at most one position. In other words, the auto-hit function has out-of-phase values of at most 1.*

Remark. *Notice that a horizontal shift of the array corresponds to a cyclic shift of the sequence and a vertical shift adds an integer modulo N to the whole sequence.*

Example:

The 4×8 sonar sequence **30310023** verifies the synchronization property and can be represented with the following array:

$$
\begin{array}{ll}
k=1 & -3 \quad 3 \quad -2 \quad -1 \quad 0 \quad 2 \quad 1 \\
k=2 & \quad 0 \quad 1 \quad -3 \quad -1 \quad 2 \quad 3 \\
k=3 & \quad -2 \quad 0 \quad -3 \quad 1 \quad 3 \\
k=4 & \quad -3 \quad 0 \quad -1 \quad 2 \\
k=5 & \quad -3 \quad 2 \quad 0 \\
k=6 & \quad -1 \quad 3 \\
k=7 & \quad 0
\end{array}
$$



The interest of sonar sequences for signal transmission lies in their low out-of-phase auto-hit as occured with autocorrelation in m-sequences. In addition, sonar sequences are not affected by the lack of synchronization between sender and receiver. Let us illustrate this property with the previous example.

Assume the previous sequences is modulated using PPM and the receiver starts listening later. The receiver builds the sequence in two steps:

1. Measuring the time differences between every two HIGH consecutive pulses.

   In this case: 1,7,2,3,4,6,5

2. Considering 0 as the first element, the following terms are derived recursively adding to the previous term its corresponding time difference and reducing modulo the number of symbols.

   In this case: **01021130**

This sequence corresponds to a vertical shift of the original one.

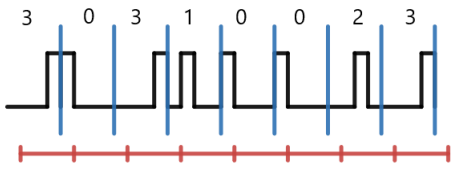


Figure 5: Out of phase Sonar sequence

2.4.1 Construction of sonar sequences from m-sequences

Several methods have been proposed for the construction of sonar sequences. In this case, we will follow Games approach Games [1987], which makes use of m-sequences.

Let $f(x)$ be a primitive polynomial of degree $2n$ over $\mathbb{F}_2$ and s a seed. It is known from the previous section that a m-sequence of period $2^{2n}-1$ can be generated. This m-sequence can be arranged in a $(2^n-1)\times(2^n+1)$ matrix filling it row by row. First column is set as the reference column and it is assigned a 0. There are now two possibilities with each of the remaining columns: either it's a zero-column or it is a phase of the reference column. In the first case, the column is assigned the ∞ symbol, whereas in the second one, the delay from the reference.

This generates a sequence of 2^n+1 terms. Considering its vertical shifts we get a longer sequence of 2^{2n} terms. Shifting the ∞ symbol to the beginning, the next 2^n-2 terms form a sonar sequence.

Example:

Let $f(x)=x^6+x^5+x^3+x^2+1$ be the characteristic polynomial of a 6-stage LFSR and $s=100011$, the initial seed. The resulting m-sequence corresponds to the following 7×9 matrix:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & \infty & 1 & 2 & 5 & 4 & 1 & 3 \end{pmatrix}$$

The following terms given by the shift sequences are: 1 1 ∞ 2 3 6 5 2 4 | 2 2 ∞ 3 4 0 6 3 5 | 3 3 ....
Shifting the ∞ term to the beginning we get the sonar sequence **12541311**.

$$\begin{array}{ll} k=1 & 1 \quad 3 \quad -1 \quad -3 \quad 2 \quad -2 \quad 0 \\ k=2 & \quad 4 \quad 2 \quad -4 \quad -1 \quad 0 \quad -2 \\ k=3 & \quad\quad 3 \quad -1 \quad -2 \quad -3 \quad 0 \\ k=4 & \quad\quad\quad 0 \quad 1 \quad -4 \quad -3 \\ k=5 & \quad\quad\quad\quad 2 \quad -1 \quad -4 \\ k=6 & \quad\quad\quad\quad\quad 0 \quad -1 \\ k=7 & \quad\quad\quad\quad\quad\quad 0 \end{array}$$

2.4.2 Families of sequences with low auto and cross-hits

One could be tempted to try to find families of sonar sequences. Unfortunately, it is known that any two sonar sequences must have cross-hit at least of 2, see the paper by Freedman and Levanon [1985]. Similarly as done for m-sequences, the sequences in the Kasami family can be rearranged into a matrix. If the sequences have length $2^{2n}-1$, then a $(2^n-1)\times(2^n+1)$ matrix is generated. The columns of each matrix are m-sequences of length 2^n-1 and there are at most two constant columns, Green and Amarasinghe [1991]. Even more, it can be proved that this family has at most two cross-hits for any pair of sequences, see the paper by Gómez et al. [2020]. The way to generate the sequences is very efficient because it is only necessary to calculate the shift of the reference m-sequence of length 2^n-1 and the different columns.

3 Positioning techniques

The objective of positioning systems is to determine the position of an object, which receives the name of target or tag. Additional devices used to locate the tag are usually called reference or access points (APs). Generally, positioning systems are made of a location sensing model, a measurement method and a positioning algorithm.

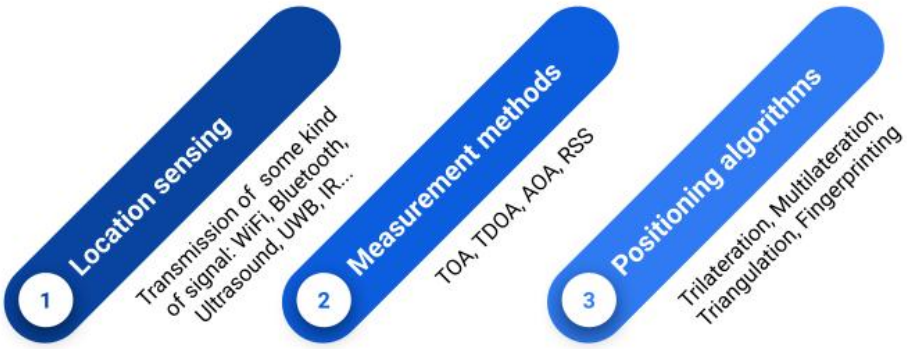


Figure 6: Structure of positioning systems

Positioning systems can find the location of an item in two or three dimensions practically analogously. The main difference is that in 3D the number of reference points required increases.

3.1 Location sensing

In order to obtain information about the position of the target some sort of signal has to be transmitted such as UWB, Ultrasound, Bluetooth, WIFI, IR, etc. The transmitter of this signal can be both the target and the reference points, thus dividing the positioning systems into active (the target is the transmitter) or passive (the target is the receiver).

Access points can either be fixed (in this case they can also be called anchors), or change their location over time. The latter happens in GPS, where the satellites are in constant motion. However, typically in indoor positioning systems, the APs remain stationary in the same position (positioning systems with Fixed Infrastructure). By contrast with the target, the position of the anchors is not unknown. This piece of information can be either known beforehand (usually with anchors) or be embedded in the signal.

Some positioning systems even include a third party along with the target and the reference points, which is in charge of processing all the information and calculating the target position. It receives the name of a base station.

3.2 Measurement methods

From the transmitted signal, some useful information for estimating the position of the target can be deduced. By information we mean physical magnitudes such as the distance that separates the tag from the anchors or the time taken to the signal to arrive to the receiver. The way in which these values are derived gives rise to different measurement methods.

Most of the techniques presented below make the assumption that there is a Line of Sight (LoS) between sender and receiver, in other words, the signal travels through the shortest path. If this path is blocked, the signal will go through a longer path because of reflections, which will lead to an incorrect target location. This often happens in tunnels and underground spaces where GPS systems can not receive a valid signal.

3.2.1 Time of Arrival (TOA)

Also known as Time of Flight (TOF), this method consists of measuring the time taken to the signal to arrive to the receiving node. For this purpose, very precise clocks are required, sometimes even atomic ones are used. From the TOA and the speed of the signal through the air, the distance between sender and receiver can be determined.

$$d = c \cdot TOA$$

Multiple versions of this scheme can be implemented which differ from each other by factors such as clock synchronization, the choice of the sender or the number of messages sent. Let us just describe the simplest ones.

One Way Ranging (OWR)

This model requires that all the devices in the system (APs and tag) are properly synchronized and that the sender includes in the message a Time Stamp, i.e, the exact hour when the signal is being sent. This way, the receiver can calculate the existing delay comparing it with the reception time. In this scenario the APs are the transmitters and the tag the receiver. This is how GPS operates.

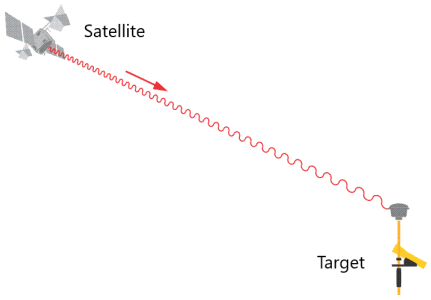


Figure 7: GPS One Way Ranging

Two Way Ranging (TWR)

Now, both the reference points and the target act as transmitter and receiver at some point of the communication. Thanks to this, the clocks needn't be synchronized. Again, there are multiple variants for TWR but only SS-TWR and DS-TWR will be discussed here. Let device A be the one which sends the first message and B the second one.

- **SS-TWR**: In Single Sided TWR, each device sends one message. Device A transmits the first one, sometimes called poll and stores the time of departure (t_0^A). Device B receives the message,

processes it storing the time of reception (t_0^B) and sends a response with the actual time (t_1^B) and the recorded one. Now is device A the receiver, which with this new time of arrival (t_1^A) and the previously stored ones can compute the TOA doing the next computation:

$$t_1^A - t_0^A = 2 \cdot TOA + t_1^B - t_0^B \implies TOA = \frac{(t_1^A - t_0^A) - (t_1^B - t_0^B)}{2} \tag{3}$$

- **DS-TWR**: In Double Sided TWR, device A sends two messages and device B one following the order shown in figure 6. In this version, it is device B that calculates the TOA based on the sending and receiving times that it has been storing and the respective times that A sends to it in the last message.

$$(t_2^B - t_1^B) + (t_1^A - t_0^A) = 4 \cdot TOA + (t_2^A - t_1^A) + (t_1^B - t_0^B)$$

$$TOA = \frac{(t_2^B - t_1^B) + (t_1^A - t_0^A) - (t_2^A - t_1^A) - (t_1^B - t_0^B)}{4} \tag{4}$$

There exists an alternative double-sided two-way ranging method in which the TOA is deduced applying the following formula:

$$TOA = \frac{t_{round_1} \cdot t_{round_2} - t_{reply_1} \cdot t_{reply_2}}{t_{round_1} + t_{round_2} + t_{reply_1} + t_{reply_2}} \tag{5}$$

where

$$\begin{aligned} t_{round_1} &= t_1^A - t_0^A \\ t_{round_2} &= t_2^B - t_1^B \\ t_{reply_1} &= t_1^B - t_0^B \\ t_{reply_2} &= t_2^A - t_1^A \end{aligned}$$

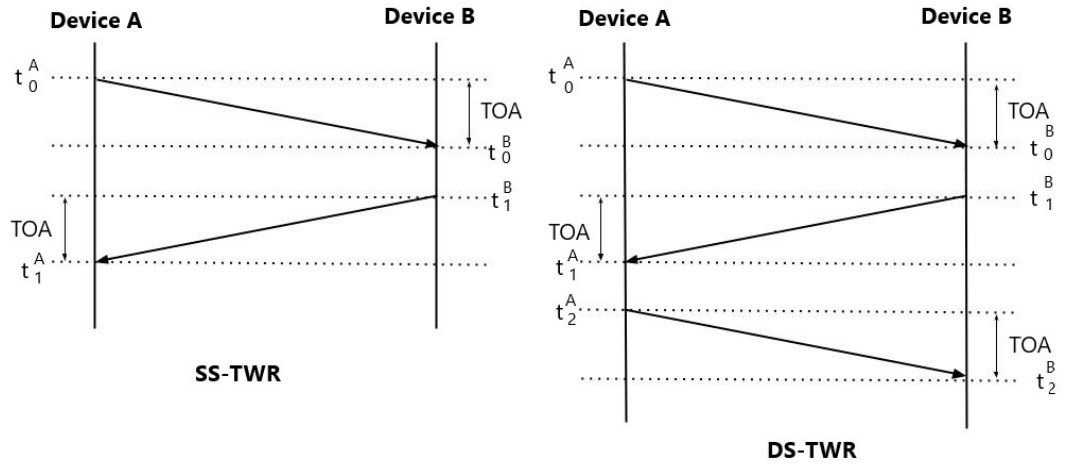


Figure 8: SS Two Way Ranging

3.2.2 Time Difference of Arrival (TDOA)

In this case the difference of time of arrival of a signal at distinct receiving stations are calculated according to a time reference which is common to all the receiving nodes. The process is the following:

The tag sends a blink which is received by the anchors at different time instants. The reference points, that are all synchronized, record the time when they receive the message and send it to a centralized base station or to the target in the absence of it. In this scenario, the tag does not need be synchronized with the reference points.

The difference in arrival time of two reference points can be used to calculate the difference in distances between them and the target.This distance can be calculated as

$$\Delta d = c \cdot \Delta t \text{ , where } c \text{ represents the speed of the signal through air}$$

This is the case for electromagnetic waves, in the case of using other types of waves speed has to be changed accordingly.

3.2.3 Angle of Arrival (AOA)

AOA, also known as Degree of Arrival (DOA), consists of determinig the angle with which the signal reaches different receiver stations. In this context, the target sends a blink which is received by the anchors.

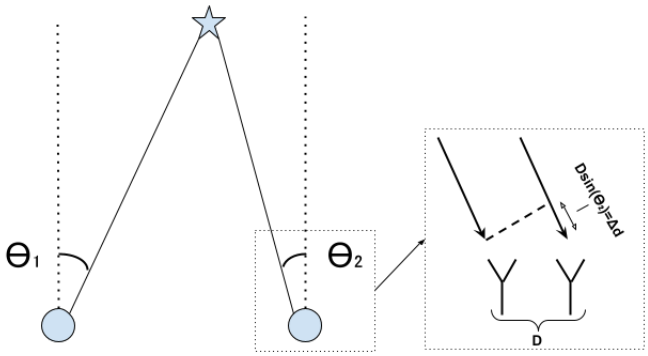


Figure 9: Angle of Arrival

In order to approximate the angles of arrival, this scheme is generally implemented using an array of antennas in each anchor. The difference in arrival times together with the distance between antennas allows estimating the degree of arrival.

3.2.4 Received Signal Strength (RSS)

The received signal strength method estimates the distance between the transmitter and the receiver based on the detected signal power (RSSI-Received Signal Strength Indicator), a non-constant magnitude which decreases as the signal propagates. This reduction in power intensity is measured by the path loss.

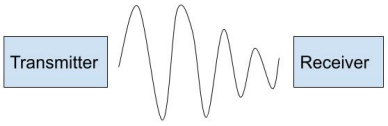


Figure 10: Received Signal Strength

There exist different propagation models that predict the path loss of a signal in terms of the environment conditions. For instance, Friis Free Space Propagation Model is restricted to free spaces where there is a clear path between the transmitter and the receiver. The Log Distance Path Loss Model, in contrast, is more appropriate for indoor scenarios where the signal can be altered by absorption, diffraction, reflections,etc. See Rappaport et al. [1996] for further reading.

According to this model, the path loss in decibels (dB) at a distance d from the transmitter can be estimated with the following formula:

$$PL(d) = P_T - P_R = PL(d_0) + 10n \log_{10}\left(\frac{d}{d_0}\right) + X$$

where

- P_T, P_R denote the signal power at the transmitter and receiver respectively
- d_0 is the distance from the transmitter to a close reference point.
- n is a constant value which depends on the environment being modeled.
- $PL(d_0)$ is the total path loss at a distance d_0 from the transmitter.
- X is a normal random variable used only when there is a shadowing effect due to signal blockage caused by obstacles.

Thus, without shadow

$$d = d_0 10^{\frac{PL(d)-PL(d_0)}{10n}}.$$

3.3 Positioning algorithms

When the positions of the reference points and some additional information is known, an algorithm makes use of this values to estimate the tag's location.

Let us introduce some notation:

(x, y)	position of the target
(x_i, y_i)	position of the anchor i
d_i	distance from anchor i to the target
t_i	time taken for anchor i to receive the signal from the target

3.3.1 Trilateration

Assuming the distances of three anchors to the target as well as their positions are known, the tag position in two dimensions can be calculated by trilateration. In three dimensions, one more anchor is necessary. Let us explain it for 2D.

For $i = 1, 2, 3$, the target must belong to C_i, the circle of center (x_i, y_i) and radius d_i. Since three circles at most intersect at one point, if any, (x, y) can be uniquely determined.

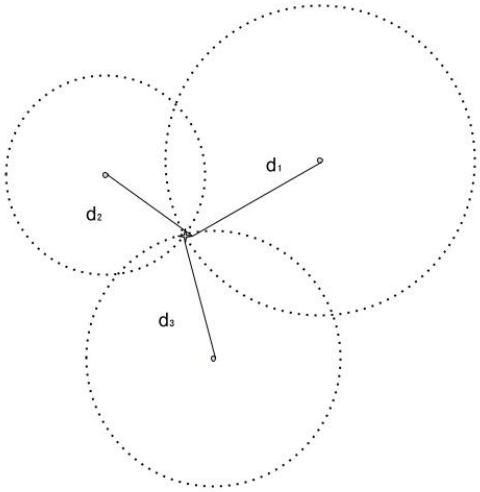


Figure 11: Trilateration

$$C_i : (x - x_i)^2 + (y - y_i)^2 = d_i^2 \implies C_i : x^2 + x_i^2 - 2xx_i + y^2 + y_i^2 - 2yy_i = d_i^2, \quad i = 1, 2, 3 \tag{6}$$

The system of nonlinear equations $\{C_1, C_2, C_3\}$ can be solved in a few steps as it is shown next.

If we substract the equations of C_1 and C_3 to C_2, we get the following expressions:

$$\begin{aligned} d_2^2 - d_1^2 &= 2x(x_1 - x_2) + x_2^2 - x_1^2 + 2y(y_1 - y_2) + y_2^2 - y_1^2 \\ d_2^2 - d_3^2 &= 2x(x_3 - x_2) + x_2^2 - x_3^2 + 2y(y_3 - y_2) + y_2^2 - y_3^2 \end{aligned}$$

Rearranging the terms of the previous equalities,

$$x(x_1 - x_2) + y(y_1 - y_2) = \frac{(x_1^2 - x_2^2) + (y_1^2 - y_2^2) + (d_2^2 - d_1^2)}{2} =: V_1 \tag{7}$$

$$x(x_3 - x_2) + y(y_3 - y_2) = \frac{(x_3^2 - x_2^2) + (y_3^2 - y_2^2) + (d_2^2 - d_3^2)}{2} =: V_2 \tag{8}$$

Finally, by reduction

$$y = \frac{V_1(x_2 - x_3) - V_2(x_2 - x_1)}{(y_1 - y_2)(x_2 - x_3) - (y_3 - y_2)(x_2 - x_1)} \tag{9}$$

$$x = \frac{V_1 - y(y_1 - y_2)}{x_1 - x_2} = \frac{y(y_1 - y_2) - V_1}{x_2 - x_1} \tag{10}$$

In practice, due to imperfect distance measurement or errors in the reference locations, it may happen that there is no intersection point between the three circles. This is usually solved, by increasing the number of anchors and using a numerical approach such as least mean squares for determining the best approximation.

3.3.2 Multilateration

When at least three differences in distances between pairs of anchors and the target are known, in addition to the positions of the former, the tag can be located in 2D by means of multilateration by measuring diference on the Time of Arrival (TOA). It is the most popular technique for indoor localization systems.

If Δd represents the difference in distances between a pair of anchors and the tag, the latter must belong to the hyperbola with foci at the anchors and whose points have a constant difference of distances to the focal points of Δd. In fact, the search can be limited to only one of its branches.

This is why this algorithm is also known as Hyperbolic Lateration.

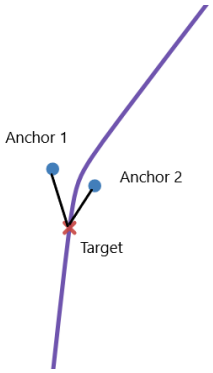


Figure 12: Hyperbola

Assuming anchor j is closer to the tag than anchor i, we have the following equation:

$$H^{i,j} : \Delta d_{i,j} = \underbrace{d_i - d_j}_{unknown} = \sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2}, \quad 1 \leqslant i < j \leqslant 3$$

Three hyperbolas are enough to identify the tag position by solving the system of nonlinear equations $\{H^{1,2}, H^{1,3}, H^{2,3}\}$. This system has to be solved numerically with algorithms such as Newton-Raphson method.

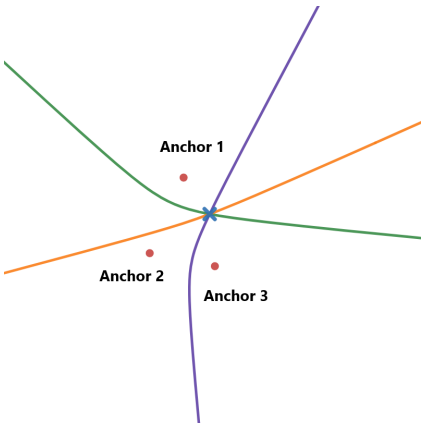


Figure 13: Multilateration

3.3.3 Triangulation

Frequently used in navigation or civil engineering, triangulation calculates the position of the tag applying triangular geometry.

Supposing the positions of the APs and their arriving angles (θ_1 and θ_2) are known, (x, y) can be easily computed.

$$L \coloneqq d\big((x_1, y_1), (x_2, y_2)\big)$$
$$L = \frac{d}{tan(\theta_1)} + \frac{d}{tan(\theta_2)} = \frac{d\sin(\theta_1 + \theta_2)}{\sin(\theta_1\theta_2)}$$
$$d = \frac{L\sin(\theta_1\theta_2)}{\sin(\theta_1 + \theta_2)}$$

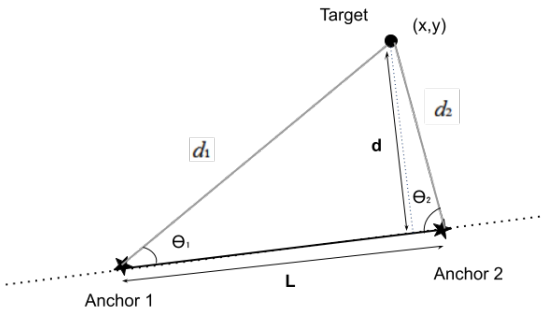


Figure 14: Triangulation

3.3.4 Fingerprinting

This method consists of previously dividing the area of study into smaller areas or cells. Some attributes of interest (such as the RSSI from the different anchors) are measured at representative points and these values are stored in a database together with their positions. This is called the calibration phase or offline stage.

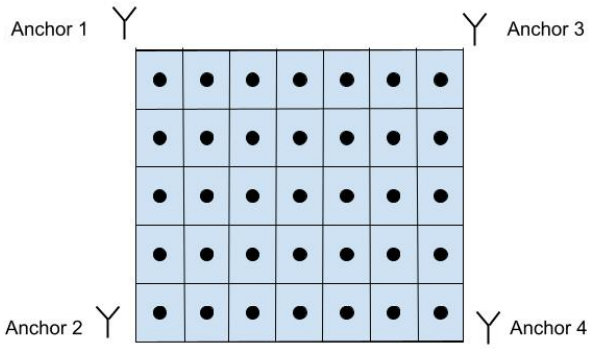


Figure 15: Possible setup for fingerprinting

Once all the areas have been described, the attributes obtained in the current tag position are compared with those in the database finding the best estimation for its location.

4 Signal Modulation and Communication Protocols

In this section, we introduce the basic concepts required to build a local positioning system using a hardware platform.

4.1 Signal Modulation

A signal is an electronic or electromagnetic current used for carrying data from one device or network to another. In electronics, a signal usually represents the variation of voltage through time. Signals can be either analog or digital.

Unlike digital signals, analog ones are continuous in time, which means they can take infinitely many different values of voltage within an interval. Digital ones, on the contrary, can only take a finite number of them. Mostly implemented in digital circuits, binary signals are an example of digital signals which only have two possible levels, 0 (LOW) and 1 (HIGH), represented as ground (0 volts) and the supply voltage, respectively.

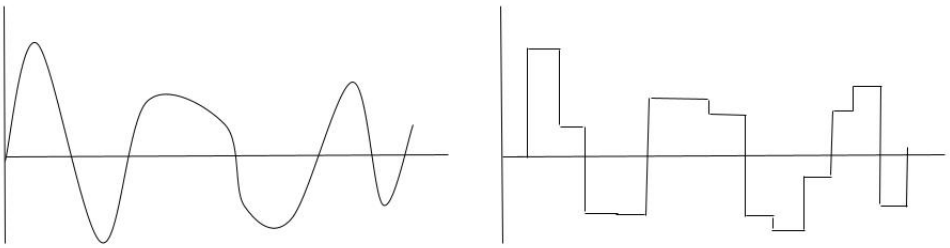


Figure 16: Analog versus digital

Even though analog signals are more accurate, digital ones are easier to manipulate and store. This is the main reason why the last ones are required for digital systems in computer processing.

The process of varying some properties of a carrier signal such as its amplitude or frequency depending on a data signal that usually carries the information is called modulation and it provides a way of swapping from analog to digital and viceversa, thus allowing the transmission of a signal through space as a radio wave or several channels of information.

4.1.1 Pulse Code Modulation

Pulse Code Modulation (PCM) is a signal modulation method used to digitally represent analog signals. This process is carried out in three stages:

1 **Sampling:** In this first step the amplitude of the analog signal is measured at certain instants of time. These values, called samples, are extracted at regular time intervals and together they form a discrete signal named PAM, that stands for Pulse Amplitude Modulation.

The number of samples taken per second is called the Sampling frequency or rate and denoted F_s. Clearly, higher sample rates lead to better discrete representations of the analog signal since the loss of information is smaller.

Theorem 4 (Sampling theorem)**.** *F_s must be at least twice the frequency of the input signal, F_{in}, in order to be able to recover it back.*

$$F_s \geqslant 2F_{in}$$

This rate is defined as the Nyquist rate.

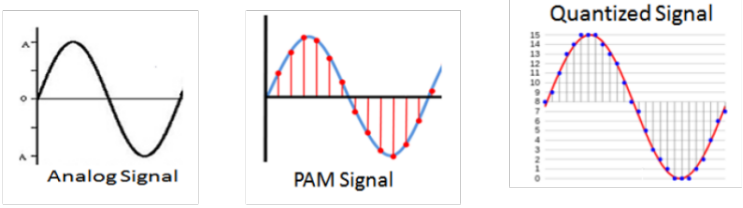
If an analog signal is sampled at a frequency N times higher than the Nyquist rate, it is said to be oversampled by a factor of N.

2 **Quantization:** Once the continuous signal has been discretized, the range of possible voltage values is divided into 2^N different levels called quantization intervals. The parameter N is called the bit-depth or resolution.

3 **Encoding:** Each quantization interval is associated with a distinct N-bit binary number applying an encoding method (two's complement, binary representation, Gray Codes). Given any sample, the closest interval is identified and it is assigned the corresponding numerical value.

That way, any continuous variable must be converted into a digital version using only a limited number of bits, which brings out the so called quantization error. This error can be calculated by adding the differences between the actual amplitude and the encoded one for every sample.

Notice that bigger resolutions decrease the quantization error.



The PCM process is implemented on analog-to-digital converters (ADCs). Those circuits are often referred to as N-bit ADCs, where N represents the bit-depth. In terms of the sampling rate, ADCs can be classified as Nyquist rate ADCs or oversampling ADCs.

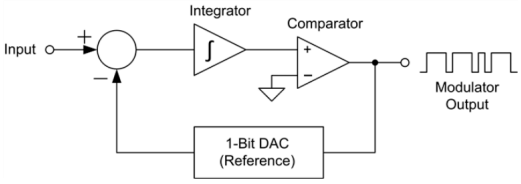
4.1.2 Pulse Density Modulation

Similarly to PCM, Pulse Density Modulation (PDM) transforms an analog signal into a binary one by first sampling it. However, since this method uses a 1-bit quantizer, each sample is assigned a unique bit, either 0 or 1 taking into account whether the signal is increasing or decreasing in comparison to the previous sample. This choice is the result of applying $\Delta\Sigma$ Modulation.

$\Delta\Sigma$ Modulation employs oversampling to reduce the noise at low frequencies and increase it at higher ones where it can be filtered out (noise shaping).

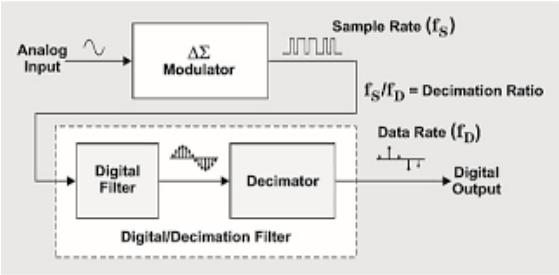
The aim of this modulator is to transmit the changes (Δ) in amplitude between consecutive samples, rather than the actual amplitude of the samples themselves. Consequently, when the amplitude of the input signal is increased it generates more 1s, and if it approaches the lowest values, more 0s.

In order to achieve this, every $\Delta\Sigma$ Modulator includes at least a difference amplifier, an integrator, and a comparator with feedback loop that contains a 1-bit DAC.



$\Delta\Sigma$ ADC

$\Delta\Sigma$ ADCs are an example of oversampling converters which consist of a delta-sigma modulator, followed by a decimation filter. As such, this kind of ADCs first encode the input signal in analog format as a bit stream using high-frequency delta-sigma modulation, and then apply a digital decimation filter that averages and downsamples it, producing a higher-resolution but lower sample-frequency digital output.

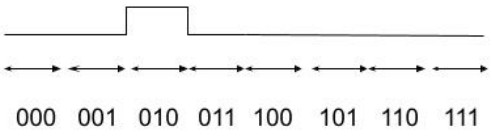


4.1.3 Pulse Position Modulation

Pulse Position Modulation (PPM) is another signal modulation method where binary messages are represented as binary signals.

With PPM a message of k bits is encoded as a pulse train of 2^k clock periods, corresponding each one to one of the possible binary sequences of length k. Only one HIGH pulse is transmitted in the position assigned to the input message.

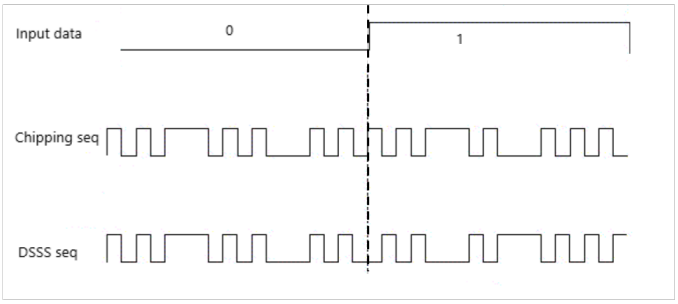
Let us show an example when $k = 3$ and the message we want to send is 010.



4.1.4 DSSS

Direct Sequence Spread Spectrum is a modulation technique based in spread spectrum technology, which has the objective of increasing the bandwidth of the transmitted signal.

In order to do so, each bit of the input data is XOR-ed with a PN-sequence, which receives the name of chipping sequence, resulting in a signal with shorter periods than the original one. Hence, 0-bits are encoded as the chipping sequence, and 1-bits as its inverse.



CDMA or Code Division Multiple Access is a method used in communications where there is only one channel shared for multiple simultaneous communications which uses DSSS to eliminate generated interferences.

Each transmitter is given a different chipping sequence to modulate the message. Ideally, if all the sequences are orthogonal, i.e, uncorrelated, the receiver is able to distinguish between the different sent signals whenever he gets a combination of them. In order to do this, the received signal is correlated with the different chipping sequences giving as output the original transmitted signal. This requires all users to be synchronized (synchronized CDMA). When synchronization between the devices is not possible, orthogonal sequences have bad correlation properties as a result of the delay between them. In this case, a set of chipping sequences with small correlation is the best alternative (asynchronized CDMA). This can be acheived with PN-sequences or Gold codes.

4.2 Short Distance Communication Protocols

Three common protocols for short distance communications in embedded systems are SPI, I^2C and I^2S. These are all serial protocols, meaning the data is transferred one bit at a time along a single wire. For this project we will focus in I^2S.

Let us start introducing the Two's complement Representation a method used in computers which is part of the I^2S protocol.

4.2.1 Two's Complement Representation

The two's complement scheme gives a binary encoding of signed integers. Using N bits, the following set of integers $\{-2^{N-1}, ..., 2^{N-1}-1\}$ can be represented.

The two's complement of a N-bit number is the complement with respect to 2^N, i.e. the number and its two's complement must add 2^N. It can be calculated by inverting all its digits and adding 1.

Following this, with this notation non-negative numbers are encoded by its ordinary binary representation whereas negative ones are represented by the two's complement of their absolute value.

The MSB represents the sign of the integer. If the leading bit is 1, the number is negative.

Remark. *One typical operation for integers given in two's complement format is to increase the number of bits used for its representation. This conversion is called sign extension.*

If $b_0 b_1 ... b_{r-1}$ is the two's complement encoding of an integer l using r bits and $s > r$, then l can be encoded with s bits in the following manner:

$$\begin{cases} l = \underbrace{0 \dots 0}_{s-r} b_0 b_1 ... b_{r-1}, \text{ if } b_0 = 0 \\ l = \underbrace{1 \dots 1}_{s-r} b_0 b_1 ... b_{r-1}, \text{ if } b_0 = 1 \end{cases}$$

4.2.2 I^{2}S protocol

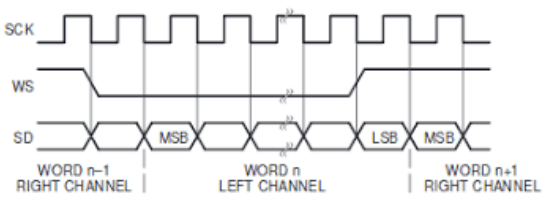
I^{2}S is used to transmit PCM data from one device, the transmitter, to another one, the receiver. For this to happen, both devices must share the same clock signal. The device which controls the clock system is called the master, while the other one will get synchronized to it acting as a slave. The master role can be taken either by the transmitter or by the receiver.

This protocol is usually implemented in audio interfaces, where there are two channels, left and right. It requires three digital signals for the communication:

- **Serial Clock/Bit Clock Line (SCK/BCK)**: At each bit-clock cycle one bit of data is sent. The frequency of this clock depends on three factors:

$$\text{freq}_{BCK} = \text{Sample Rate} \times \text{Bits per channel} \times \text{Number of channels}$$

- **Word Select/Left-Right Clock (WS/LRCK)**: Used to swap from one channel to another, the frequency of WS clock is the sample rate.
- **Serial Data (SD)**: The Serial Data encoded in two's complement with the most significative bit (MSB) first is transmitted through this signal. It is important to bear in mind that the WS line changes one bit-clock period before the MSB is transmitted.



Arduino IDE includes an I^{2}S library which implements this protocol on SAMD21 based boards.

5 Set-up

We will implement the indoor positioning system using Arduinos, an ADC and the ultrasonic sensors HC-SR04.

5.1 Components

All the instruments used in the project can be easily purchased online. DigiKey and Mouser are possibly the most common suppliers of electronic components.

5.1.1 Arduino

Arduino is an open-source hardware and software tool used for electronic projects. In terms of hardware, Arduino consists of a circuit board or microcontroller which can be programmed using a specific software called Arduino IDE. With this programming environment code can be written in a computer and loaded to the board by simply connecting an USB.

The microcontroller is internally attached to some entries called pins which can have multiple functionalities such as connecting the board to other devices through jumping wires. As in other embedded systems, the microcontrollers have a low memory capacity.

Arduino is very intuitive, which makes it a good option for electronics beginners. Besides, it has a huge community online so there is a great number of resources such as examples, tutorials and forums. Another plus point is its compatibility with a lot of devices. Nonetheless, Arduino is not used for commercial purposes due among other things to its cost and a better performance of other microcontrollers. These reasons explain why this embedded system is not the best choice when producing real life products. Instead, it is mainly used for developing prototypes.

Hardware

There are a number of different Arduino boards available on the market but we are using Arduino UNO R3 and Arduino MKRFOX1200.

Any Arduino board has the following elements:

- microcontroller: microchip
- digital Pins which can be declared as input or output, some of which are PWM.
- analog input Pins: can be used as digital pins
- system clock

The Arduino UNO R3 is a simpler board containing the ATmega328 microchip. The Arduino MKRFOX1200, instead, has the SAMD21 microcontroller which includes some improvements with respect to the previous one, such as compatibility with I^2S protocol. For this purpose, pins A6, 2 and 3 of this board integrate the three digital lines: SD, BCK and LRCK. These lines are required to communicate the Arduino board with the chosen AD Converter with I^2S as shown in 17. Consequently, one Arduino MKRFOX1200 will be used as the target.

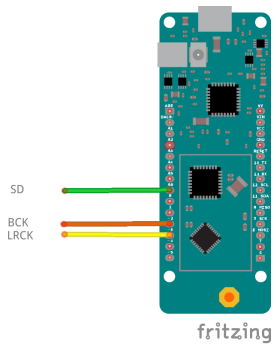


Figure 17: MKRFOX1200

Software

The programming language employed in Arduino IDE is a version of C++ which includes special methods and functions.

Specific methods for Arduino:

- `pinMode(pin, INPUT/OUTPUT)` sets the type of the specified pin.
- `digitalRead(pin)` reads the tension going through the specified digital pin and returns HIGH or LOW. For boards operating at 5V (respectively 3.3V) if tension is higher than 3V (respectively 2V) it returns HIGH and LOW, otherwise.
- `digitalWrite(pin, HIGH/LOW)` writes a HIGH (3.3/5V) or a LOW (0V) value to the specified digital pin.
- `analogWrite(pin, value)` generates a square wave of the specified duty cycle. Only valid for PWM pins.
- `delay(time)` makes the processor wait the specified time in miliseconds until next instruction is executed.

Any program in Arduino receives the name of sketch and it must contain at least a setup function and a loop function. Only called once at the beginning of the sketch, as its name suggests, the setup function is where initial settings are established. On the other hand, the loop function is run repeatedly until the microcontroller is turned off.

Once the sketch is uploaded to the board, it is stored in the microcontroller and there is no longer need to connect it to a computer. Only a power supply such as an external battery is required for operation.

5.1.2 Ultrasonic sensor

HC-SR04 is an ultrasonic sensor designed for distance measurement. It has two transducers (a transmitter and a receiver), 4 pins (Vcc, Trig, Echo, Ground) and a 40 KHz frequency-clock. It can measure distances in a range of 2 to 400 cm with accuracy of 3mm. It has a beam angle of 30 degress.

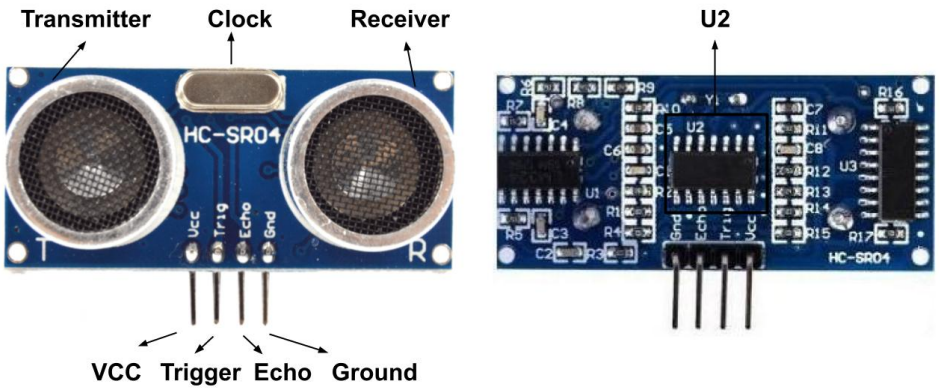


Figure 18: HC-SR04

Pin Trigger, which is connected to the transmitter, acts as input, whereas Echo is connected to the receiver and acts as output.

After pin Trig gets a pulse of at least 10 μs, the transmitter sends 8 pulses which travel through the air. At the same time pin Echo turns HIGH and waits until either the signal is reflected on some object and goes back to the receiver, or the time exceeds, which means there is no object nearby.

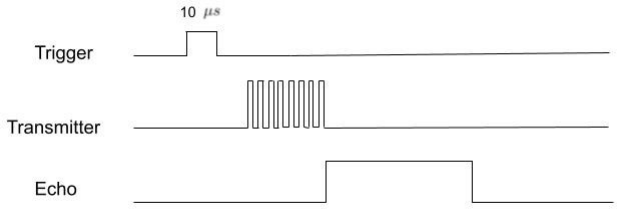


Figure 19: Operating mode of HC-SR04

In the first case, as soon as the signal is received Echo turns LOW. Thus, the width of the pulse of Echo can be used to calculate the distance to the reflected object.

$$2d = v \cdot t$$

We are interested in finding the distance to more than one object, but this design of the sensor only allows us to find how far it is from the closest one. For this reason, with the help of an oscilloscope, we have found where is the analogical signal processed internally by the module of the sensor before getting the digital output. This occurs in pin 10 of U_2.

5.1.3 Analog-to-digital Converter

The Texas Instruments PCM1808EVM is an evaluation module designed to test the PCM1808, a $\Delta\Sigma$ ADC with 24-bit resolution and sampling rate varying from 8 kHz to 96 kHz.

The ADC can be chosen to work in master or slave mode by the control pins MD0 and MD1. When operating in master mode, the evaluation board internal clock will be the one derived for the communication. This specific model is manufactured with a system clock of 12.28 MHz, which at most allows us to sample at 48 kHz. However, due to Theorem 4 (Nyquist theorem), we are interested in sampling rates of at least 80 kHz since the Ultrasonic sensor is working at 40 kHz. This can be attained with a quicker system clock of for example 24.576 MHz.

SAMPLING RATE (kHz)	SYSTEM CLOCK FREQUENCY (f_{SCLK}) (MHz)		
	$256f_s$	$384f_s$	$512f_s$
48	12.288	18.432	24.576
64	16.384	24.576	32.768
88.2	22.5792	33.8688	45.1584
96	24.576	36.864	49.152

Table 2: Relation between Sampling Frequency and System Clock Frequency

Table 2 shows the sampling rate depending on the frequency of the system clock, that can be found in the specifications. Rather than the original clock we will be using the ECS-245.7-20-5PX-TR, another crystal oscillator which having similar attributes to the previous one, but a higher frequency of 24.576MHz, lets us obtain the desired sampling rate.

On the other hand, the transmitted data can be sent in two different formats: 24-bit I^2S or 24-bit Left-justified which can be decided setting the FMT pin.

The module has two analog input pins, one for each channel (left and right) and 4 pins for the I^2S protocol which can be either output or input depending on the operating mode (master/slave). In Master mode, both LRCK and BCK are provided as an output by the ADC to transmit the serial data to an external circuit and BCK is set to 64 times the sampling frequency.

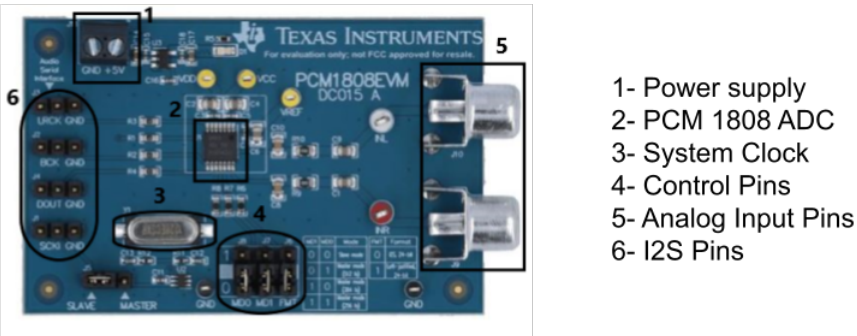


Figure 20: PCM1808EVM

More specifications about this evaluation module can be found in its datasheet (See Tex [2006]). Apart from this there is little documentation available online regarding this evaluation board.

5.1.4 Oscilloscope

An oscilloscope is a device that allows us to graphically visualize electrical signals which vary on time. The x-axis represents the time and the y-axis the voltage.

As an example, the following two figures show how the BCK and LRCK signals of the evaluation module are seen in the oscilloscope screen. This equipment lets us visualize and measure some properties of the signals such as their frequency. In this case, the down-middle tab $\frac{1}{\Delta x}$ indicates LRCK and BCK have respectively frequencies of approximately 96KHz and 6.144MHz, that are the expected values.

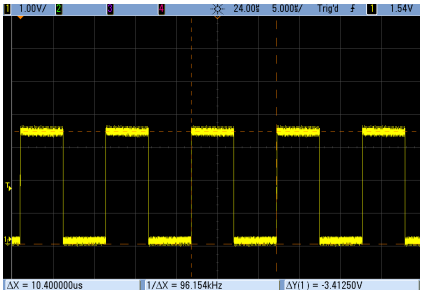


Figure 21: LRCK Signal

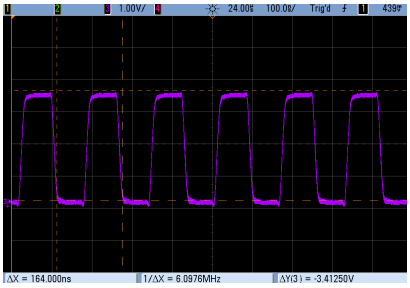


Figure 22: BCK Signal

5.2 Final Prototype

In this section, we build a prototype to test the use of the Gold sequences using Pulse Position Modulation (PPM). For the emission of an encoded message, each bit is assigned one LOW and one HIGH pulse, not necessarily in this order, so that the receiver is able to recover it back by identifying the position of the HIGH pulse in the slot-interval. Our code takes as input a specific sequence and a duration for each interval, but those values could be changed.

The simplest experiment is composed of just two nodes, one acting as a target and the other acting as a reference point.

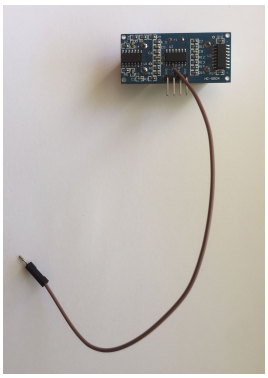


Figure 23: Shielded HCSR-04

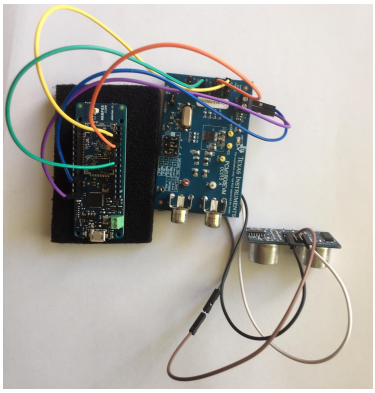


Figure 24: Target

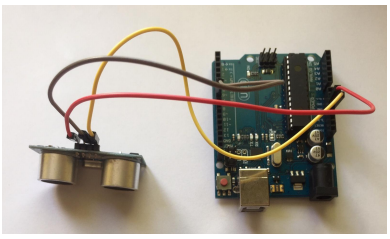


Figure 25: Reference points

Each type of node plays a different role:

- Reference Point: The node is composed by an Arduino UNO and a HCSR-04. A ultrasonic PPM signal is emitted, the positions of the pulses depending on the sequence selected. Possible sequence lengths are 7, 15 and 31, but the code is fully customizable (Figure 25).
- Target: The node is composed by an Arduino MKRFOX 1200, an ADC and one HCSR-04 (as there are two channels, two sensors could be used as in Figure 26). In this configuration the target should be able to accurately calculate the distance to the reference point (See Figure 24). To perform this calculations, the analog signal that contains the pulses is required. Following the project posted in Instructable-Circuits, we have welded the pin 10 of U2 from the ultrasonic sensor of the target to a jumper wire. This allows us to use the analog signal from the receiver sensor (Figure 23).

  This analog signal is then fed to the ADC, that communicates through the I2S protocol, where the ADC board takes the role of master. This had a sampling rate limitation of at most 48kHz,

so we removed the original clock of our ADC and shielded the ECS-245.7-20-5PX-TR instead in order to reach a sampling rate of 96 KHz.

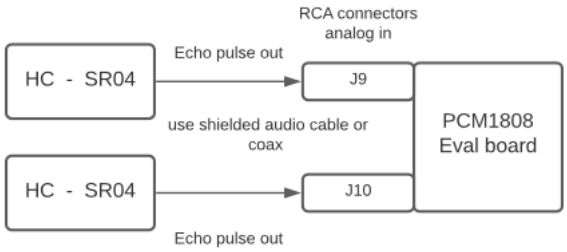


Figure 26: Target schema for ADC and two ultrasound receivers configuration

Next diagram illustrates all the connections between the components.

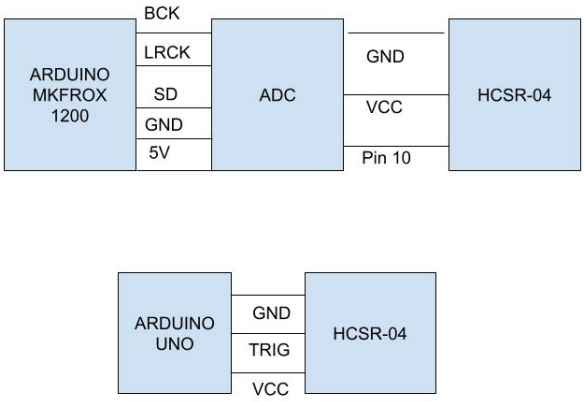


Figure 27: Connecting schema

The necessary pieces of code for both the transmission and reception stages have been written (see Code). We have implemented a LFSR generator which lets us obtain the corresponding m-sequences given a seed and a characteristic polynomial. This is is then plugged in other pieces of code which generate preferred pairs of m-sequences and Gold or Kasami families. This is useful for the target since the memory in the microcontroller is limited and the multiple chipping sequences may be very long.

Our experiment consisted of a single reference point emitting a message and the target being able to retrieve it. As the expected result was not obtained, we decided to step back and see where the failure was. For that we focused on studying the behavior of the ADC and the I^2S library. We connected the ADC to a current generator and sent through it a sinusoid from 0 to 3V. The signal that was being recovered was not as expected and also stopped at a certain moment despite still being receiving information.

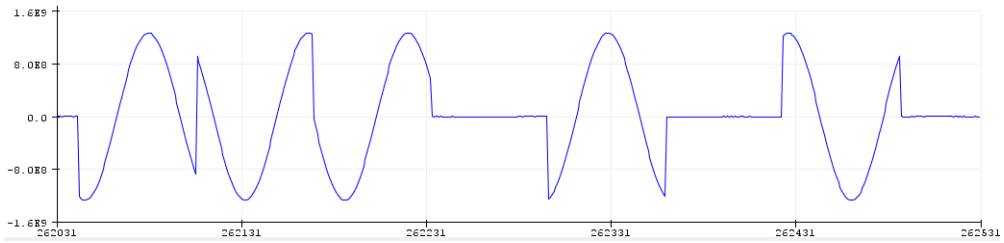


Figure 28: Received signal at Arduino MKFROX 1200

From the data available, we deduce the following:

- The problem observed may be due to the incompatibility of formats between the ADC board and the Arduino I2S library. The ADC represents the amplitude of a sample of the signal with 24 bits, but there is no such variable in Arduino. On the other hand, they do exist for 8, 16 and 32 bits.
- Another possible issue might be related with a loss of synchronization, a situation explained in section 7.3.3 of Tex [2006].

To fix the first problem we tried to identify what kind of transformation was internally being performed by Arduino between the 24 input bits and the 32 output bits. We found that the last byte of each sample was being padded with 0s. We tried removing them and interchanging the byte order without any success. Regarding the loss of synchrony, we managed to avoid it by restarting the I2S protocol, but this fix could lead to data loss.

As a result, these problems may be attributed to the lack of compatibility between the Arduino and the ADC, apart from the scarcity of examples and documentation of the evaluation board. We ordered a different ADC (the evaluation module ADS8353Q1EVM-PDK) that handles the output signal and integrates all the process in a single board. It has a better interface and the sampling rate is higher to avoid operating too close to the Nyquist rate. Since it did not arrive on time due to external circumstances, it is left as future work.

5.3 Conclusions and future work

The present master thesis subject contains the steps needed to build a proof-of-concept for an indoor localization system (IPS) using ultrasonic technology, that allows for non expensive prototyping. The technology used was based on Arduino devices and the communication channel is the 40 Khz frequency band, although the framework, theoretical background and code are suitable for implementation in UWB.

A non-exhaustive list of the results of this Master Thesis is:

- Researching on the different ranging techniques, both passive and active, and their geometrical formulation;
- Mathematical study of the properties of the sequences for different ranging and communication techniques;
- Implementations in Arduino of several primitives for LFSR generation, correlation of sequences, Pulse Position Modulation and Direct Sequence Spread Spectrum;
- Preparation and Testing of the ultrasonic module and the Analog Digital converter;

Most of the work regarding the sequences has been already done and implemented. Also, code on transmission has been tested and it is working as expected. Due to unexpected circumstances, it has not been possible to finish the reception part as the ADC was not able to recover the transmitted signal. For this reason, future work covers testing a different ADC module, running integration tests on the system and evaluating the system in real conditions, including several reference points, presence of noise, etc.

A Code

```
/*
   LFSR generator
    * regist represents the state of the register
    * pol represents the characteristic polynomial
*/

String seq;
String regist;
String pol;

void setup() {
   Serial.begin(9600);
   regist = "111";  // Example (Initial seed has to be different to zero)
   pol="011";  // Example (1+ x^2 + x^3)
   seq="";
   for(int i=pow(2,regist.length()); i>0; i--) {
      Serial.println(i);
      seq+=regist[regist.length()-1];
      regist=lfsr_next(regist); //Calculates the following state of the register
   }
   Serial.println(seq);
}

// Auxiliary function to produce the next state:
String lfsr_next(String seq) {
  String rotate=seq.substring(0,seq.length()-1);
  int sum=0;
  for(int i=pol.length();i>0;i--){
    sum=sum^(String(seq[i]).toInt());
    }
  if (sum){
    rotate="1"+rotate;}
  else{
    rotate="0"+rotate;}
  return rotate;
}

void loop() {}
```

```
/*
   SHIFT CALCULATOR:
   Calculates the correlation between a known sequence (seq1) and all the phases
   of a received sequence (seq2).
   The maximum correlation corresponds to the shift.
*/

String seq1;
String seq2;

void setup() {
  seq1="1001011"; //Example
  seq2="1011100"; //Example
  Serial.begin(9600);
  Serial.println(desplazamiento(seq1,seq2)); //Visualize the shift
}

//AUXILIARY FUNCTIONS:

//Correlation between two sequences:
int correlacion(String seq1,String seq2){
  int sum;
  sum=0;
  for (int i=0;i<=seq1.length()-1;i++){
    sum+=pow(-1,String(seq1[i]).toInt())*pow(-1,String(seq2[i]).toInt());
    }
   return sum;
  }

//Shift a string k positions:
String my_rotate(String seq, int k){
  String rotate=seq.substring(seq.length()-k)+seq.substring(0,seq.length()-k);
  return rotate;
}

//Calculate the shift between two sequences:
int desplazamiento(String seq1,String seq2){
  int corr_max=-1;
  int pos_max=0;
  for (int i=0;i<seq1.length();i++){
    int cor=correlacion(seq1,my_rotate(seq2,i));
    Serial.println(my_rotate(seq2,i));
    Serial.println(cor);
    if (cor>corr_max){
      corr_max=cor;
      pos_max=i;
      }
    }
  return pos_max;
  }

void loop() {}
```

```
/*
   PREFERRED PAIR GENERATOR:
   Generates a preferred pair of sequences given a m-seq (seq) of period 2^n-1
   and an integer (q) such that:
            * n not a multiple of 4
            * q=2^k-1 ó q=2^(2k)-2^k+1
            * mcd(n,k)=1 if n is odd or 2 if n is even (n=2 mod 4)
*/

String seq;
int q;
String decimada;

void setup() {
  Serial.begin(9600);
  seq="1011100";  //Example
  q=3;            //Example
  decimada=decimar(seq,q);
  Serial.println(decimada);  //Visualize the decimation

  //Check correlation values are -1,-t(n) y t(n)-2 (in this case -1,-5,3)
  for (int i=0;i<seq.length();i++){
    Serial.println(correlacion(my_rotate(seq,i),decimada));
    Serial.println(correlacion(decimada,my_rotate(seq,i)));
    }
}

//AUXILIARY FUNCTIONS:

//Decimated sequence:
String decimar(String seq,int decimacion){
  String decimada="";
  for (int i=0;i<seq.length();i++){
    decimada+=seq[(i*decimacion)%seq.length()];
    }
  return decimada;
  }


//Calculate the correlation
int correlacion(String seq1,String seq2){
  int sum;
  sum=0;
  for (int i=0;i<=seq1.length()-1;i++){
    sum+=pow(-1,String(seq1[i]).toInt())*pow(-1,String(seq2[i]).toInt());
    }
   return sum;
  }

//Shift a string k positions:
String my_rotate(String seq, int k){
  String rotate=seq.substring(seq.length()-k)+seq.substring(0,seq.length()-k);
```

```
  return rotate;
}

void loop() {}
```

```
/*
    Gold Codes generator:
        * seq represents a maximal sequence generated with a primitive
          polynomial of deg n
        * decim represents the qth decimation of seq

          * Requirements: (seq,decim) are a prefered pair
             * n not a multiple of 4
             * q=2^k-1 or q=2^(2k)-2^k+1
             * mcd(n,k)=1 if n is odd or 2 if n is even (n=2 mod 4)

        * Gold[2^n+1] represents an array with all the Gold codes
*/
String seq;
String decim;

void setup() {
  Serial.begin(9600);
  seq="1110100";  //Example
  decim="1001011"; //Example
  String Gold[seq.length()+2];

  //Generate Gold family:
  Gold[0]=seq;
  Gold[1]=decim;
  for (int i=0;i<seq.length();i++){
    Gold[i+2]=string_xor(my_rotate(decim,i),seq);
  }

  //Visualize Gold family:
  for (int i=0;i<seq.length()+2;i++){
    Serial.println(Gold[i]);
    }
}

//Auxiliary functions:

//Rotate a string k positions:
String my_rotate(String seq, int k){
  String rotate=seq.substring(seq.length()-k)+seq.substring(0,seq.length()-k);
  return rotate;
}

//Xor of two sequences:
String string_xor(String seq1, String seq2){
  String result="";
  for (int i=0;i<seq1.length();i++){
    result+=String(String(seq1[i]).toInt()^String(seq2[i]).toInt());}
  return result;
  }

void loop(){}
```

```
/*
   KASAMI SEQUENCE generator
      *Given a m-sequence(seq)of length 2^n-1, with n even, the decimation
       w=u[2^(n/2)+1] (decim) is calculated.
      *Kasami family is generated :
       u, u+w, u+Tw, ..., u+T^(2^(n/2)-2))w
*/

String seq;
String decim;
int n;
String Kasami[4];  //Example

void setup() {
  Serial.begin(9600);

  //Example 1:
  //generated by x^6+x^5+x^3+x^2+1 (length 63)
  //seq="110001001000011100000101111110010101000110011110111010110100110";
  //n=6;

  //Example 2:
  //generated by x^4+x+1 (length 15)
  seq="100100011110101";
  n=4;
  decim=decimar(seq,pow(2,n/2)+1);

  //Generate Kasami family:
  Kasami[0]=seq;
  for (int i=0;i<=pow(2,n/2)-1;i++){
    Kasami[i+1]=string_xor(seq,my_rotate(decim,i));
    }

  //Visualize family:
  for (int i=0;i<=pow(2,n/2);i++){
    Serial.println(Kasami[i]);
    }
}

//AUXILIARY FUNCTIONS:

//Decimated sequence:
String decimar(String seq,int decimacion){
  String decimada="";
  for (int i=0;i<seq.length();i++){
    decimada+=seq[(i*decimacion)%seq.length()];
    }
  return decimada;
  }

//Shift a string k positions:
String my_rotate(String seq, int k){
```

```
  String rotate=seq.substring(seq.length()-k)+seq.substring(0,seq.length()-k);
  return rotate;
}

//Xor of two sequences:
String string_xor(String seq1, String seq2){
  String result="";
  for (int i=0;i<seq1.length();i++){
    result+=String(String(seq1[i]).toInt()^String(seq2[i]).toInt());}
  return result;
  }

void loop() {}
```

```
/*
 SONAR SEQUENCES generator:
    * Given a m-sequence of period 2^n-1 with n even a sonar sequence is built
 */

int m=3; //Example m=n/2
String mseq="110001001000011100000101111110010101000110011110111010110100110";

void setup() {
  Serial.begin(9600);
  sonar_array(m,mseq);
}

//AUXILIARY FUNCTIONS:

//--Secuencia sonar: (Following Games article)
void sonar_array(int m,String mseq){
  int filas=pow(2,m)-1;
  int columnas=pow(2,m)+1;
  //Calculate the first column of the matrix
  String col1="";
  String zeros="";
  for (int j=0;j<filas;j++){
      col1=col1+mseq[columnas*j];
      zeros=zeros+"0";}
  //Calculate the remaining columns and their delay from the first one
  String col;
  for (int i=1;i<columnas;i++){
    col="";
    for (int j=0;j<filas;j++){
      col=col+mseq[i+columnas*j];}
      if (col==zeros){
        Serial.println("Infinity");
        }
       else{
        Serial.println(desplazamiento(col,col1));}
    }
  }

//--Calculate correlation between two sequences:
int correlacion(String seq1,String seq2){
  int sum;
  sum=0;
  for (int i=0;i<=seq1.length()-1;i++){
    sum+=pow(-1,String(seq1[i]).toInt())*pow(-1,String(seq2[i]).toInt());
    }
   return sum;
  }

//--Shift a string k positions:
String my_rotate(String seq, int k){
  String rotate=seq.substring(seq.length()-k)+seq.substring(0,seq.length()-k);
```

```
  return rotate;
}

//--Calculate the shift between two sequences:
int desplazamiento(String seq1,String seq2){
  int corr_max=-1;
  int pos_max=0;
  for (int i=0;i<seq1.length();i++){
    if (correlacion(my_rotate(seq1,i),seq2)>corr_max){
      corr_max=correlacion(my_rotate(seq1,i),seq2);
      pos_max=i;
      }
    }
  return pos_max;
  }

void loop() {}
```

```
/*
   PULSE POSITION MODULATION:
   Modulates a given sequence (seq) using PPM taking 2*mytime as
   the slot-duration and sends it through the ultrasonic module.
*/

String seq;             // Message
int disparador = 6;     // Trigger of sensor
int entrada = 5;        // Echo of sensor
float mytime;           // Half of slot-duration

void setup() {
  pinMode(disparador, OUTPUT);
  pinMode(entrada, INPUT);
  seq="10010"; //Example
  mytime=2.5    //Example
  Serial.begin(9600);
  modulacion(seq);
}

void modulacion(String seq){
  for (int i=0;i<seq.length();i++){
    if (String(seq[i])=="0"){
      digitalWrite(disparador, HIGH);
      delayMicroseconds(mytime);
      digitalWrite(disparador, LOW);
      delayMicroseconds(mytime);
    }
    else{
      digitalWrite(disparador, LOW);
      delayMicroseconds(mytime);
      digitalWrite(disparador, HIGH);
      delayMicroseconds(mytime);
      }
    }
  }

void loop() {}
```

```
/*
    2D TRILATERATION :
    If the distances of three anchors to the central arduino are known
    as well as their position, we can identify the
    position of the arduino.
*/

float coord1[2],coord2[2],coord3[2],r1,r2,r3; //Circles
float posx,posy; //Position of the target

void setup() {
  Serial.begin(9600);
  //Declaring the known variables:

  //Anchor 1
  coord1[0]=5;  //Example coordinate x
  coord1[1]=3;  //Example coordinate y
  r1=4;         //Example radius

  //Anchor 2
  coord2[0]=0;  //Example coordinate x
  coord2[1]=1;  //Example coordinate y
  r2=3;         //Example radius

  //Anchor 3
  coord3[0]=1;  //Example coordinate x
  coord3[1]=2;  //Example coordinate y
  r3=2;         //Example radius

  //Determine posx, posy:
  trilateracion(coord1,coord2,coord3,r1,r2,r3);
  Serial.println(posx);
  Serial.println(posy);
}

//Auxiliary function for Trilateration:
void trilateracion(float* coord1,float* coord2, float* coord3,
                   float r1, float r2, float r3){
  float V1=(pow(coord1[0],2)-pow(coord2[0],2)+pow(coord1[1],2)
           -pow(coord2[1],2)+pow(r2,2)-pow(r1,2))/2;
  float V2=(pow(coord3[0],2)-pow(coord2[0],2)+pow(coord3[1],2)
           -pow(coord2[1],2)+pow(r2,2)-pow(r3,2))/2;
  posx=(V1*(coord2[0]-coord3[0])-V2*(coord2[0]-coord1[0]))/
       ((coord1[1]-coord2[1])*(coord2[0]-coord3[0])-
       (coord3[1]-coord2[1])*(coord2[0]-coord1[0]));
  posy=(posx*(coord1[1]-coord2[1])-V1)/(coord2[0]-coord1[0]);
  }

void loop(){}
```

Bibliography

Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A. Al-Ammar, and Hend S. Al-Khalifa. Ultra-Wide Band Indoor Positioning technologies:analysis and recent advances. *Sensors*, 2016.

Arduino. I2s reference documentation, 2021-07-16. URL https://www.arduino.cc/en/Reference/I2S.

Hari Balakrishnan, Roshan Baliga, Dorothy Curtis, Michel Goraczko, Allen Miu, Nissanka B Priyantha, Adam Smith, Ken Steele, Seth Teller, and Kevin Wang. Lessons from developing and deploying the Cricket indoor location system. *Preprint*, 2003.

William Cherowitzo. Linear feedback shift registers. URL http://www-math.ucdenver.edu/~wcherowi/courses/m5410/m5410fsr.html.

François Despaux, Adrien Van den Bossche, Katia Jaffrès-Runser, and Thierry Val. N-twr: An accurate time-of-flight-based n-ary ranging protocol for Ultra-Wide Band. *Ad Hoc Networks*, 79:1–19, 2018.

Avraham Freedman and Nadav Levanon. Any two n× n costas signals must have at least one common ambiguity sidelobe if n> 3—a proof. *Proceedings of the IEEE*, 73(10):1530–1531, 1985.

Y. Fukuju, M. Minami, H. Morikawa, and T. Aoyama. Dolphin: an autonomous indoor positioning system in ubiquitous computing environment. In *Proceedings IEEE Workshop on Software Technologies for Future Embedded Systems. WSTFES 2003*, pages 53–56, 2003. doi: 10.1109/WSTFES.2003.1201360.

Richard A. Games. An algebraic construction of sonar sequences using m-sequences. *SIAM Journal on Applied Mathematics*, 8(4), 1987.

Solomon W. Golomb. *Shift Register Sequences*. Aegean Park Press, USA, 1967. ISBN 0894120484.

Solomon W. Golomb and Guang Gong. *Signal design for good correlation: For wireless communication, cryptography, and radar*, volume 9780521821. 2005. ISBN 9780511546907. doi: 10.1017/CBO9780511546907.

Solomon W. Golomb and Herbert Taylor. Constructions and Properties of Costas Arrays. *Proceedings of the IEEE*, 72(9):1143–1163, 1984. ISSN 15582256. doi: 10.1109/PROC.1984.12994.

DH Green and SK Amarasinghe. Families of sequences and arrays with good periodic correlation properties. *IEE Proceedings E (Computers and Digital Techniques)*, 138(4):260–268, 1991.

Ana I. Gómez, Domingo Gómez, and Andrew Tirkel. Generalised gmw sequences. *IEEE ISIT*, 2020.

Victor Pardo Gómez. *CURSO ARDUINO AVANZADO*. Inven, Santander, Spain, 2016.

Instructable-Circuits. Enhanced Ultrasonic Range Finder. URL https://www.instructables.com/Enhanced-Ultrasonic-Range-Finder/.

Cung Lian Sang, Michael Adams, Timm Hørmann, Marc Hesse, Mario Porrmann, and Ulrich Rückert. Numerical and experimental evaluation of error estimation for two-way ranging methods. *Sensors*, 19(3):616, 2019.

Mark Liffiton. Acordeón arduino. URL http://platea.pntic.mec.es/~mhidalgo/documentos/05_LenguajeResumen_Arduino.pdf.

Svet Maric, Ivan Seskar, and Edward L Titlebaum. On cross-ambiguity properties of welch-costas arrays. *IEEE Transactions on Aerospace and Electronic Systems*, 30(4):1063–1071, 1994.

Rainer Mautz. *Indoor positioning technologies*. PhD thesis, ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, 2012.

Carlos Medina, José Carlos Segura, and Angel De la Torre. Ultrasound Indoor Positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. *Sensors*, 13(3): 3501–3526, 2013.

David Munoz, Frantz Bouchereau Lara, Cesar Vargas, and Rogerio Enriquez-Caldera. *Position location techniques and applications*. Academic Press, 2009.

I2S bus specification. Philips Semiconductors, 2nd edition edition, 1996.

Nissanka Bodhi Priyantha. *The Cricket indoor location system*. PhD thesis, Massachusetts Institute of Technology, 2005. URL http://cricket.csail.mit.edu/.

Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey, 1996.

Antonio Ramón Jiménez Ruiz and Fernando Seco Granja. Comparing ubisense, Bespoon, and Decawave UWB location systems: Indoor performance analysis. *IEEE Transactions on instrumentation and Measurement*, 66(8):2106–2117, 2017.

Dilip V. Sarwate and Michael B. Pursley. Crosscorrelation properties of pseudorandom and related sequences. *IEEE*, 1980.

Richard G Swan. Factorization of polynomials over finite fields. *Pacific Journal of Mathematics*, 12 (3):1099–1106, 1962.

PCM1808 Single-Ended, Analog-Input 24-Bit, 96-kHz Stereo ADC datasheet. Texas Instruments, 2006. URL https://www.ti.com/lit/ds/sles177b/sles177b.pdf?ts=1626082216515.

Jan Van Sickle. *GPS for land surveyors*. CRC press, 2008.

Jan Van Sickle. The GPS signal, 2017-07-15. URL https://www.e-education.psu.edu/geog862/node/1407.

Albert Viaplana Ventura. Indoor positioning using fake ble beacons. Master's thesis, Universitat de Barcelona, 2016.

Hong-Peng Wang and Gang Han. The research on responding time and precision of centralized controlled Cricket indoor location system. In *2009 WRI International Conference on Communications and Mobile Computing*, volume 1, pages 349–352. IEEE, 2009.

Lloyd R. Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information theory*, 20(3):397–399, 1974.

Tan F. Wong. Spread spectrum and code division multiple access. URL http://wireless.ece.ufl.edu/twong/notes1.html.

Miodrag Živković. A table of primitive binary polynomials. *Mathematics of Computation*, 62(205): 385–386, 1994.