Implementación de un sistema remoto para administrar el servicio de agua potable en viviendas de difícil acceso por su ubicación geográfica.

Stiwan Ruiz Henao

Universidad Nacional Abierta Y A Distancia

Escuela De Ciencias Básicas, Tecnología E Ingeniería

Ingeniería Electrónica

Medellín, 2020

Implementación de un sistema remoto para administrar el servicio de agua potable en viviendas de difícil acceso por su ubicación geográfica.

Stiwan Ruiz Henao

Trabajo de Grado

Presentado como requisito para optar por el título de Ingeniero Electrónico

Asesor:

Santiago Rúa Pérez

Universidad Nacional Abierta Y A Distancia

Escuela De Ciencias Básicas, Tecnología E Ingeniería

Ingeniería Electrónica

Medellín, 2020

Contenido

Resumen

En este documento, se describe un proyecto de teleoperación, el cual tiene como fin identificar, diseñar e implementar una solución al problema que tienen diferentes empresas prestadoras del servicio de agua potable en zonas o viviendas donde el acceso para los operadores del servicio es difícil debido a su ubicación geográfica. Por tal motivo, se plantea la implementación de un sistema de comunicación usando la tecnología LoRa y, así, obtener la medida del micromedidor (consumo) y enviar señal de apertura o cierre a la electroválvula que se puede implementar sobre la tubería de agua potable que llega a la vivienda.

En este documento se refieren cada uno de los componentes que se usan para implementar el sistema de teleoperación, también se da a conocer los códigos de programación de las tarjetas Arduino utilizadas, adicional a esto se realiza una descripción de la forma como se realiza la comunicación LoRa entre los nodos y cuáles son los parámetros para configurar estas tarjetas programables con protocolo LoRa.

Abstract

In this document, a teleoperation project is described, which aims to identify, design and implement a solution to the problem faced by different companies providing drinking water in areas or homes where access for service operators is difficult due to their geographical location. For this reason, the implementation of a communication system using LoRa technology is proposed and, thus, obtain the measurement of the micrometer (consumption) and send signal of opening or closing to the electrovalve that can be implemented on the drinking water pipe that reaches the house.

This document describes each of the components that were used to implement the remote control system, it also gives the programming codes of the Arduino cards used, additional to this is a description of how LoRa communication is performed between the nodes and what were the parameters to configure these cards programmable with LoRa protocol.

Agradecimientos

Quiero agradecerle a mi familia quienes me han tenido tanta paciencia y han sabido entender mi dedicación al estudio y creo saber, que es porque han visto mi deseo tan grande de graduarme como Ingeniero Electrónico y aunque muchas veces el miedo de pensar en perder algún curso me invadía, estaban a mi lado para ayudarme a creer nuevamente en mí y darme muchos ánimos con decirme simplemente que el esfuerzo valía la pena.

Agradezco a mis familiares lejanos ya que, aunque no podía frecuentarlos tanto, ellos entendían que estaba dedicado a mi estudio y a mi trabajo. También quiero agradecerles a los tutores ya que algunos me hicieron sentir lo que siempre he esperado de un profesor y es saber que realmente a ellos les interesa que el estudiante aprenda en vez de crear un terror ilógico al curso.

Lista de figuras

Lista de tablas

Introducción

Este proyecto busca inicialmente la implementación de un sistema de comunicaciones inalámbrico llamado LoRa, el cual trae beneficios muy importantes para sistemas o proyectos donde se requiera tener una medición o un control remoto sobre algún dispositivo de instrumentación, otra de las cosas que se quiere obtener con la implementación de este proyecto es mitigar los problemas que tienen muchos operadores de servicios públicos en Colombia, ya que existen zonas que no tienen acceso al sistema de agua potable debido a que son zonas recónditas o dispersas, o zonas en las ciudades que están en proceso formalización (Acero,2019). Estas zonas son de difícil gestión, corresponden a usuarios que se encuentren ubicados en zonas fuera del área urbana (Maya, 2017). Por lo tanto, no es viable para el prestador de servicio público de agua potable llegar a estas zonas para realizar la respectiva gestión de conexión, interrupción y medición.

Las dificultades de monitores e interconexión de los servicios debido a la problemática expuesta, facilita que el suscriptor del servicio de agua potable opte por usar métodos alternativos de abastecimiento de agua para consumo, los cuales en su mayoría son inadecuados (Mesías et al., 2018). Esta situación de conexiones alternativas hace que se incrementen las pérdidas por fraude o daños en la infraestructura debido al uso de herramientas inadecuadas para la conexión.

Luego de entender la problemática que se presenta y de la investigación sobre el protocolo de comunicación LoRa, se busca la forma de programar estos dispositivos que trabajan en conjunto con Arduino, sin embargo, luego de que la comunicación es efectiva

es necesario entender las limitaciones del sistema de comunicación ya que es un aspecto muy importante a la hora de evaluar una nueva tecnología de comunicación ya que en algunos casos puede volverse un problema que un sistema de comunicación para teleoperar un sistema no responda en algún momento que se deba operar remotamente un sistema.

Una vez que se tiene la comunicación entre los nodos y las posibles limitaciones de esta tecnología que para este caso no son importantes, se procede a realizar la investigación sobre cómo es el sistema de medición que se utiliza para totalizar el consumo de agua y la válvula para corte y apertura del servicio, luego de esto, se busca un sensor de flujo y una electroválvula con características eléctricas que permiten trabajar con Arduino y de esta manera hacer una emulación del sistema propuesto. Luego de tener claro cómo se debe realizar el montaje de los instrumentos, se comienza a investigar cómo programar las tarjetas Arduino para recibir el dato del sensor de flujo y accionar la electroválvula que se decidió implementar, de esta manera se realiza la programación adecuada para que el nodo cliente envíe al nodo servidor el dato que fue procesado del sensor de flujo y como recibir el carácter para abrir y cerrar la electroválvula.

El objetivo principal que se tiene con el diseño e implementación del proyecto es implementar un sistema remoto para la interconexión y medición del servicio público de agua en zonas residenciales de difícil acceso para los operarios, este objetivo se logra una vez que se realiza la implementación de la solución y se obtiene el dato que se recibe del sensor de flujo y la electroválvula, por lo tanto una vez que en la aplicación de Processing se desarrolla se puede observar estos datos del sensor y manipular la electroválvula, la implementación del proyecto termina y surgen así las opciones de mejora que se pueden

implementar.

En este escrito del proyecto se pueden encontrar los diferentes objetivos que se tienen presentes, un marco teórico donde se exponen todos los conceptos tenidos en cuenta en la comunicación LoRa, la programación de las tarjetas Arduino y los shield LoRa, la programación de la aplicación desarrollada en Processing, diagramas de conexión y las opciones de mejora que se pueden tener para el proyecto. Este proyecto contiene cada uno de esos componentes necesarios para entender como fue el diseño y desarrollo del proyecto del sistema remoto para la interconexión y medición del servicio público de agua en zonas residenciales de difícil acceso para los operarios.

Planteamiento y formulación del problema

Teniendo en cuenta la problemática que tienen algunas empresas prestadoras de servicios públicos para prestar el servicio de agua potable a viviendas, donde por la ubicación geográfica no es fácil el acceso. Este proyecto, permitirá realizar mediciones del consumo y el corte o reconexión del servicio de agua potable, garantizando un servicio estable donde se puede intervenir remotamente y de forma oportuna. De esta manera, el prestador del servicio de agua potable podrá tener una buena administración sobre el servicio.

El sistema telecontrolado se llevará a cabo mediante la tecnología LoRa, ya que es capaz de alcanzar rangos superiores a 15 km con un bajo consumo de energía (Robson & Haddad, 2019), lo cual le permite un amplio rango de autonomía evitando la visita de un operador en cortos periodos de tiempo.

Con la tecnología LoRa y una buena línea de vista entre los nodos y el gateway se puede tener una comunicación estable y recopilar datos de un gran número de nodos desplegados en áreas amplias (Rizzi et al., 2017). De esta manera, es posible hacer gestión remota a un gran número de viviendas en un sector.

Este sistema de teleoperación también ayudará a mitigar las pérdidas de agua cuando se presenten daños en la infraestructura, debido a la capacidad de respuesta en el cierre o apertura de las válvulas, pues esta tecnología es configurable de acuerdo con las necesidades, obteniendo respuestas basadas en el ciclo de trabajo configurado.

Justificación

Por medio del presente proyecto se pretende buscar una solución a la problemática que se presenta en muchas zonas de Colombia para acceder a ciertas viviendas que por su ubicación geográfica o por los problemas sociales es difícil intervenir en el sistema de acueducto.

Este proyecto se desarrolla utilizando el sistema de comunicación LoRa en modo punto a punto, dando como resultado un sistema de interconexión de nodos de bajo costo además de brindar estabilidad y seguridad a la hora de intervenir el acueducto de alguna vivienda donde el acceso al operador del servicio no sea fácil o conveniente.

También se busca con la implementación de este proyecto automatizar el proceso de medida del consumo de agua y el proceso de corte o apertura del servicio de agua

potable residencial, de esta forma el prestador del servicio público podrá recibir información y manipular el servicio de forma remota sin necesidad de tenerse que desplazar cada mes a la vivienda en la que se implemente la solución.

Objetivos

Objetivo General

Implementar un sistema remoto para la interconexión y medición del servicio público de agua en zonas residenciales de difícil acceso para los operarios.

Objetivos específicos

- Identificar una solución con base en la teleoperación a través de la comunicación por radio enlace.
- Diseñar un sistema telemétrico con protocolo de comunicación LoRa en tipo de red punto a punto que permita realizar operaciones de control y medición del caudal de agua.
- Implementar el sistema de teleoperación diseñado usando tarjetas de desarrollo, electroválvulas y micromedidores.

Marco conceptual

Las comunicaciones inalámbricas se pueden llevar a cabo mediante diferentes tecnologías de radio enlace, sin embargo, este tipo de comunicación aprovecha las características de largo alcance de la capa física LoRa, lo que permite un enlace de un solo salto entre el dispositivo final y una o varias puertas de enlace-gateways (Risc, 2020). Esta tecnología, funciona en las bandas ISM sin licencia y utiliza la técnica de modulación de radio Chirp Spread Spectrum (CSS) en la capa física, y un protocolo de capa MAC LoRaWAN (Carlsson et al., 2018), donde su banda de frecuencia está en 433 Mhz (Asia), 868 Mhz (Europa) y 915 Mhz (América).

LoRa es un sistema que tiene un muy buen alcance de transmisión, las primeras investigaciones han demostrado que la tecnología es capaz de alcanzar rangos superiores a 15 km con un consumo de energía relativamente bajo (Robson & Haddad, 2019). Este sistema de comunicación es muy estable, sin embargo, se debe estudiar el campo de radio. Es una tecnología que tiene un buen alcance, sin embargo, se debe tener en cuenta la importancia de la línea de vista y la potencia de transmisión de las antenas que se utilicen. Para tener buena estabilidad en el enlace se debe verificar que no haya elementos que interfieran en la línea de vista entre los nodos y el gateway.

Un nodo final LoRa es un dispositivo de bajo consumo y gran potencia de transmisión, que envía o recibe datos a un Gateway, cabe resaltar que se les puede dotar de comunicación entre sí (Risc, 2020). Esta interconexión se realiza configurando ambos dispositivos teniendo en cuenta la dirección de cada nodo y del gateway.

Para el valor del ciclo de trabajo para un nodo LoRA se debe considerar un retraso entre las tramas sucesivas enviadas por el nodo final. Si el valor es 1%, el dispositivo tendrá que esperar 100 veces la duración de la última trama antes de enviar nuevamente en el mismo canal (Carlsson et al., 2018). Teniendo en cuenta el valor que configuremos en nuestro Chip podemos saber cuál será la duración aproximada de la próxima recepción de trama.

LoRaWAN está orientado a eventos y enlaces ascendentes (de manera similar a otras soluciones de IoT). En aplicaciones típicas, se recopilan datos de un gran número de nodos desplegados en áreas amplias (Rizzi et al., 2017). La arquitectura de red LoRaWAN se implementa en topología estrella en la que las puertas de enlace retransmiten mensajes entre dispositivos finales y un servidor de red central

El Gateway es el dispositivo responsable de recibir la información del nodo y llevarla a la red; siendo la antena uno de sus elementos más importantes, ya que esta definirá el alcance de comunicación con los nodos finales (Navarro & et al., 2018). El Gateway puede tener diferentes características como son los puertos disponibles para conexión a internet y/o swiches, la cantidad de nodos finales que soporta para el envío y recepción de información.

Existen multitud de soluciones tecnológicas remotas que se pueden implementar para obtener un dato y/o controlar a distancia. Por esto, el monitoreo inalámbrico ubicuo y el Internet de las cosas, donde la duración de la batería del dispositivo es un factor

limitante, son aplicaciones obvias para LoRa (Robson & Haddad, 2019). Por lo tanto, si se requiere un sistema que tenga poco consuma y gran capacidad LoRa es una muy buena opción.

En el transporte es muy común encontrarse con dispositivos de radio comunicación que sirven para rastrear los vehículos, por lo tanto, LoRa también puede ser una solución para este tipo de innovaciones. Según Adelantado (2017), refiere que el roaming es uno de los desarrollos bajo definición dentro de LoRa Alliance para mejorar la movilidad. Por otra parte, esta tecnología tiene múltiples beneficios y características adecuadas para la localización, como bajo consumo de energía, largo alcance de transmisión y bajo costo de implementación (Ha et al., 2019), además de robustez a la interferencia gracias a una modulación de espectro ensanchado configurable en secuencia y ciclo de trabajo fijo (Croce et al., 2018). También hay que destacar su gran alcance de transmisión, LoRa es beneficioso para el sistema IoT que requiere larga distancia comunicación en comparación con protocolos de corto alcance como Wi-Fi y Bluetooth, (Zourmand, 2019). Por lo tanto, este sistema de comunicación es muy adecuado para implementaciones donde se requiere un sistema de teleoperación o teleoperación que sea estable y de bajo costo.

El uso de tecnologías inalámbricas cada vez es mayor, ya que son soluciones de bajo costo y muy útiles en lugares donde la cobertura de comunicaciones es escasa o incluso inexistente y donde no siempre hay acceso a la energía eléctrica (Hernandez, 2019). Por lo tanto, la tecnología LoRa puede ser una solución para largo alcance o falta de energía eléctrica para las comunicaciones en lugares remotos.

Metodología de la investigación

El desarrollo del proyecto se realiza de una forma muy práctica, ya que las tarjetas de desarrollo que se utilizan, que en este caso son Arduino y Dragino LoRa Shield, deben ser programadas desde el IDE de Arduino e incluir librerías, sin embargo el desarrollo de la programación en Arduino está basada en un esquema de escalización de una señal análoga y el control de una salida de esta tarjeta, pero como se menciona anteriormente, este proyecto se desarrolla utilizando comunicación LoRa en la frecuencia de 915MHz y por lo tanto las tarjetas deben configurarse primero desde su archivo config.h y luego realizar un programa que integre la tarjeta LoRa Shield y Arduinos, que básicamente consiste en recepcionar y/o enviar los datos que se reciben por medio del Shield LoRa.

La idea de este proyecto surge desde el año 2019, en una visita al Municipio Apartadó donde participaba de un proyecto de EPM, el cual consiste en la optimización del acueducto de este municipio, se observa que para Aguas Regionales es un problema llegar a varias viviendas que cuentan con el servicio de agua potable, pero por la ubicación de estas viviendas o la problemática social del sector donde están ubicadas se hace complejo ir a tomar la medida del consumo de estas viviendas; en el momento que se determina esta problemática surgen varias ideas como solución a este problemática, teniendo en cuenta que existen varios protocolos de comunicaciones por radiofrecuencia que pueden dar solución a esta situación y hasta quizás disminuir costos a largo plazo ya que se puede automatizar este proceso de lectura y corte o apertura del servicio.

Luego de que esta idea surge, se inicia la investigación sobre los tipos de comunicación existentes y en algún momento, al escuchar a varias personas, una de ellas comenta sobre un nuevo protocolo llamado LoRa, se presentan entonces mayores dudas y también más iniciativa de investigar sobre aquel protocolo. Cuando comienza la investigación sobre este protocolo se evidencia que es un protocolo que ya se viene usando en países como Chile, México, Estados Unidos, entre otros y que existen varios dispositivos en el mercado de las comunicaciones que ya están utilizando este protocolo de comunicación, por lo tanto, al conocer dicha información se decide realizar el proyecto de grado con este tipo de sistema de comunicación.

Teniendo claro que el objetivo es realizar un proyecto enfocado a la solución de este problema de acceso a las viviendas con una ubicación geográfica de difícil acceso, se decide que se va a utilizar el protocolo de comunicación LoRa para dar solución a esta problemática. Desde este momento se adquieren unas tarjetas de Dragino LoRa y se comienzan a realizar ensayos de comunicación entre 2 tarjetas que se envían un “Hola Mundo”; al comienzo es un poco frustrante ya que la comunicación no se logra, con desespero y sin saber lo que pasa, se cambian las antenas al pensar que pueden haber vendido las tarjetas de 915MHz con antenas de una frecuencia distinta a las preprogramadas de la tarjeta, pero finalmente se determina que este no es el problema porque se cambian y aun así no se da la comunicación. Luego de hacer varios intentos y con ayuda de foros en la web de Dragino se observa que no se ha verificado el archivo de configuración con el que las tarjetas trabajan una vez el código de programación es cargado.

De esta manera se muestra la forma como se realiza este proyecto y se puede entender que ha sido en forma ensayo error y con la ayuda de foros, adicionalmente a esto después de investigar, se comprueba que para realizar este proyecto es necesario entender cómo funciona LoRa, toda esta información se recopila de la base de datos que maneja la UNAD, así que al investigar en aplicaciones como Mendeley y bibliotecas como son IEEE, EBSCO, ebooks, entre otras; se desarrolla este proceso de aprendizaje y el logro de este maravilloso proyecto.

Marco metodológico

Teniendo en cuenta el relato en el marco de investigación, el desarrollo del proyecto se realiza por etapas, teniendo como guía el orden de los objetivos plateados, por lo tanto, a continuación, se expone la forma como se desarrolla cada uno de ellos.

Inicialmente se identifica una solución con base en la teleoperación a través de comunicación por radio enlace, teniendo en cuenta que en la biblioteca de la UNAD hay diferentes bases de datos, se revisan diferentes referencias bibliográficas para adquirir bases conceptuales y, de acuerdo con los conocimientos que estas referencias aportan, se identifican las referencias en las cuales se desarrollan sistemas de teleoperación con tecnología LoRa y topología estrella.

Luego se procede a diseñar un sistema telemétrico con protocolo de comunicación LoRa en topología estrella, que permite controlar el consumo y el abastecimiento de agua potable residencial en zonas de difícil acceso. Para el desarrollo del sistema basado en la

teleoperación se realiza un diagrama con la arquitectura de control y comunicaciones, donde se tienen en cuenta los diferentes niveles de control para poder clasificar los instrumentos instalados en campo (tubo de agua), los dispositivos de control, los dispositivos de comunicación y por último la central de visualización y tratamiento de datos obtenidos del micromedidor. Adicionalmente se realizan planos de conexión entre los dispositivos de control y comunicaciones, donde se pueden observar las señales del instrumento de medida (micromedidor) y de control (electroválvula). Una vez se tiene clara la arquitectura de control y comunicación del sistema se realiza un diagrama de flujo donde se puede observar el comportamiento que debe tener el sistema.

Por último se implementa el sistema de teleoperación diseñado para controlar la electroválvula y tomar la medida del consumo de agua potable mediante un micromedidor(Sensor de flujo), este sistema de teleoperación consta de tarjetas Arduino, shields LoRa y Gateway LoRa que se implementa con RaspberryPi y LoRa HAT que a su vez está configurado como nodo final; este proyecto se plantea desarrollar con los dispositivos electrónicos propuestos y además se implementa un framework desarrollado con el lenguaje de programación Processing donde se puede visualizar el comportamiento del proceso de apertura y cierre de válvula remotamente, además de ver el consumo promedio.

Se realizan pruebas al sistema por partes, es decir, inicialmente se verifica la conexión entre los shield LoRa con el servidor, luego se hacen las pruebas de la recolección de datos y control sobre la electroválvula, por último, se ajusta la parte grafica de la visualización de los datos.

Implementación del sistema de teleoperación

Listado de elementos

En el siguiente listado se describen los elementos electrónicos y eléctricos que se utilizan para el desarrollo del proyecto:

Tabla 1. Elementos utilizados

ELEMENTO	DESCRIPCIÓN
Fuente 110/220 VAC - 5V DC	Alimentación electroválvula y Arduinos
Arduino UNO	Procesador de datos
Shield LoRa	Servidor LoRa
Arduino DUE	Nodo recolector de datos y controlador
Shield LoRa GPS	Cliente LoRa
Electroválvula	Abre/cierra paso de agua
Sensor de flujo	Micromedidor consumo de agua
Caja plástica	Agrupamiento de elementos de cada nodo

Listado de maniobra (Fuente: elaboración propia)

Entradas y salidas de la tarjeta Arduino DUE

En la siguiente lista se describen las salidas y entradas de la tarjeta Arduino en configuración cliente:

Tabla 2. Entradas y salidas utilizadas nodo cliente

PIN	FUNCIÓN
Sensor de Flujo - D49	Recibir pulso del sensor
Activación relé – D50	Apertura y cierre electroválvula
Activación relé -D51	Apertura y cierre electroválvula

Entradas y salidas (Fuente: elaboración propia)

Programación en IDE Arduino

Teniendo en cuenta que las tarjetas a utilizar son Arduino y Shield LoRa de Dragrino, se procede a realizar la programación en el IDE de Arduino y la interconexión de las tarjetas, cabe destacar que las tarjetas LoRa Dragino reciben comandos AT, pero para este proyecto se configuraron sus archivos en extensión .h y luego mediante la programación en Arduino.

En la siguiente figura se puede ver la programación en Arduino del nodo servidor que será el que se conecta al PC o Raspberry Pi y se lograrán observar los datos mediante Processing haciendo uso del puerto serial.

Figura 1 Void setup nodo servidor (Fuente: elaboración propia)

En la figura 1 se puede observar que se incluyen las librerías SPI (Serial Peripheral Interface) que es un bus de comunicación a donde la transmisión de datos se realiza en serie, es decir un bit después de otro. Otra de las librerías que se incluye es la propia de la comunicación LoRa para trabajar con el circuito integrado SX1276/SX1278 que es el integrado del shield LoRa Dragino, en esta librería se encuentra el archivo RH_RF95 en el cual se realiza la configuración de la comunicación LoRa.

Adicional a las librerías que se incluyen, en Void Setup se puede ver que se definen las salidas que se van a tener en nuestro Arduino que en este caso se define el pin 13 de

Arduino como salida y se configura para que indique que se ha enviado o recibido un dato al nodo cliente. También se inicia el puerto serial a 9600 baud rate, se inicia el módulo LoRa y se configura su poder de transmisión en 23 dBm (decibelio -milivatio).

Saliendo del void setup se encuentra la figura 2, en la que se puede ver el proceso que se realiza en void loop que es la rutina o el proceso que se va a ejecutar luego de configuración del código.

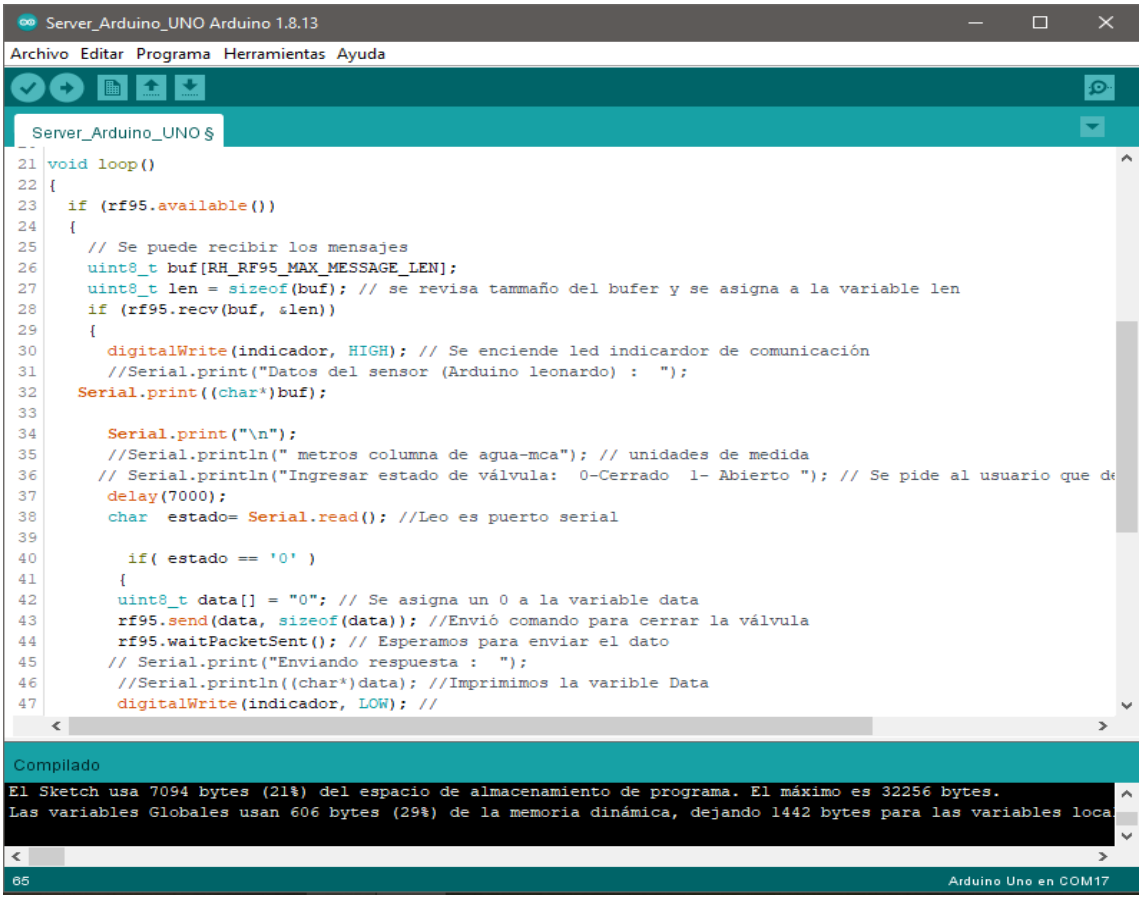

Figura 2 Void loop nodo servidor (Fuente: elaboración propia)

En el *void loop* se puede ver que la primera instrucción que se realiza es la sentencia condicional *if* donde para entrar a este ciclo debe de estar el integrado

SX1276/SX1278 conectado al otro módulo nodo cliente, luego de realizar esta verificación si se cumple se entra en un proceso donde se reciben los datos que envía el nodo cliente y se guarda en la variable *buf* que está definida como *unit8_t* que ayuda a declarar una variable donde solo se puede tener un byte.

Luego de declarar la variable donde se van a almacenar los datos que se reciben, se procede a calcular el tamaño de la variable *buf* y se guarda en *len*, que es otra variable que se declara con el *unit8_t.* luego se hace otra sentencia condicional *if* para saber si se recibe el dato y de esta forma entrar a mostrar por el *indicador* (pin13 Arduino) que se recibe un dato y lo imprime en forma de carácter por el puerto serial y de esta manera se muestra en la aplicación realizada en Processing.

Luego de mostrar el dato que se recibe del nodo cliente, se procede a leer del puerto serial que en la aplicación de Processing es un botón el cual arroja un carácter “1” o “0” para saber si se abre o se cierra la electroválvula.

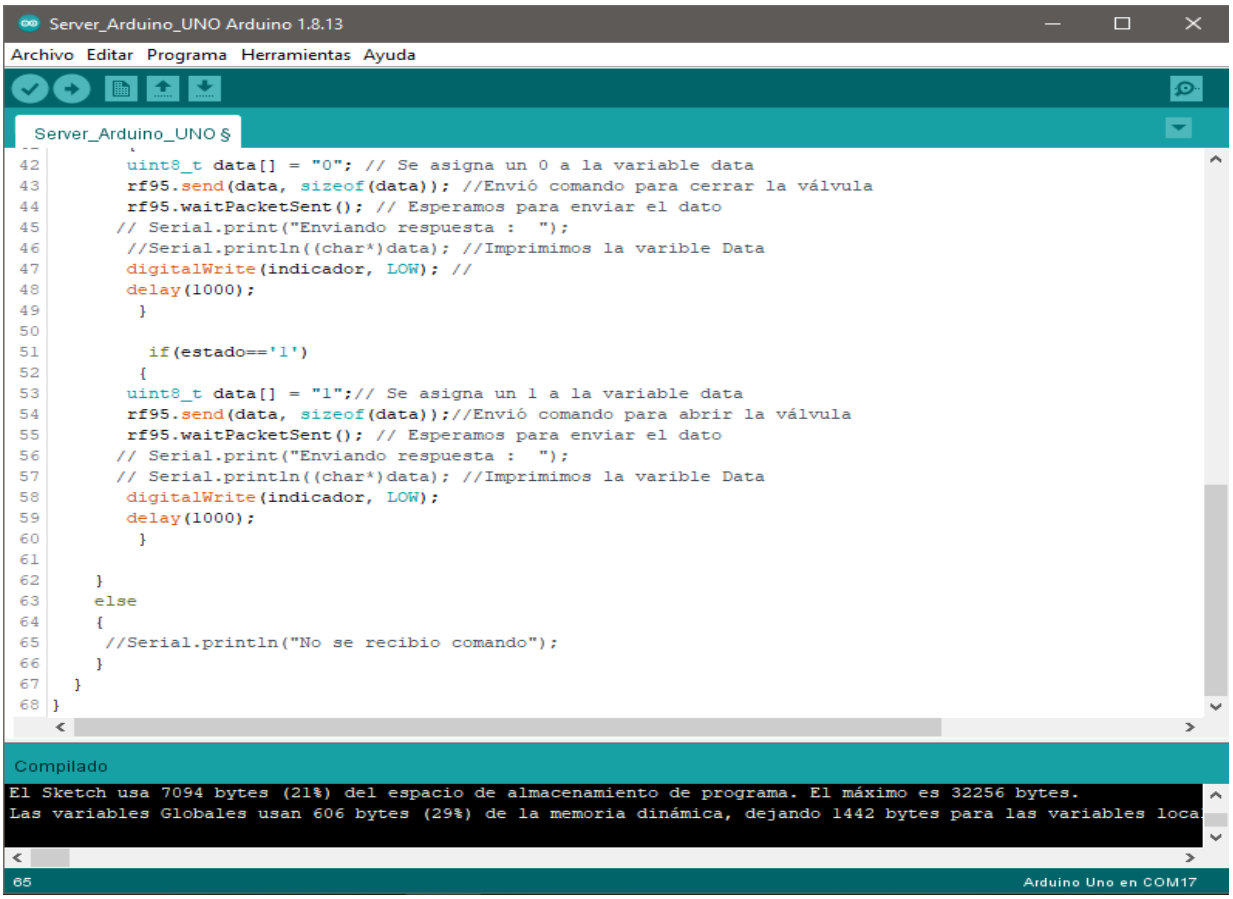

Figura 3 Control de electroválvula nodo servidor (Fuente: elaboración propia)

En la figura 3 se puede ver que se realiza la sentencia condicional *if* para comprobar si se desea abrir o cerrar la electroválvula, también se incluye la instrucción que envía al nodo cliente el dato tipo carácter para abrir o cerrar la electroválvula.

El nodo cliente es el que envía al nodo servidor los datos de consumo que son medidos mediante el sensor de flujo, además de poder controlar la apertura y cierre de la electroválvula. En la siguiente figura se puede visualizar que la programación en el IDE de Arduino es muy similar a la del nodo servidor, sin embargo, se debe tener claro que el nodo servidor es el que recibe la información y mediante el puerto serial del Arduino UNO y con Processing se pueden mostrar los datos que se reciben.

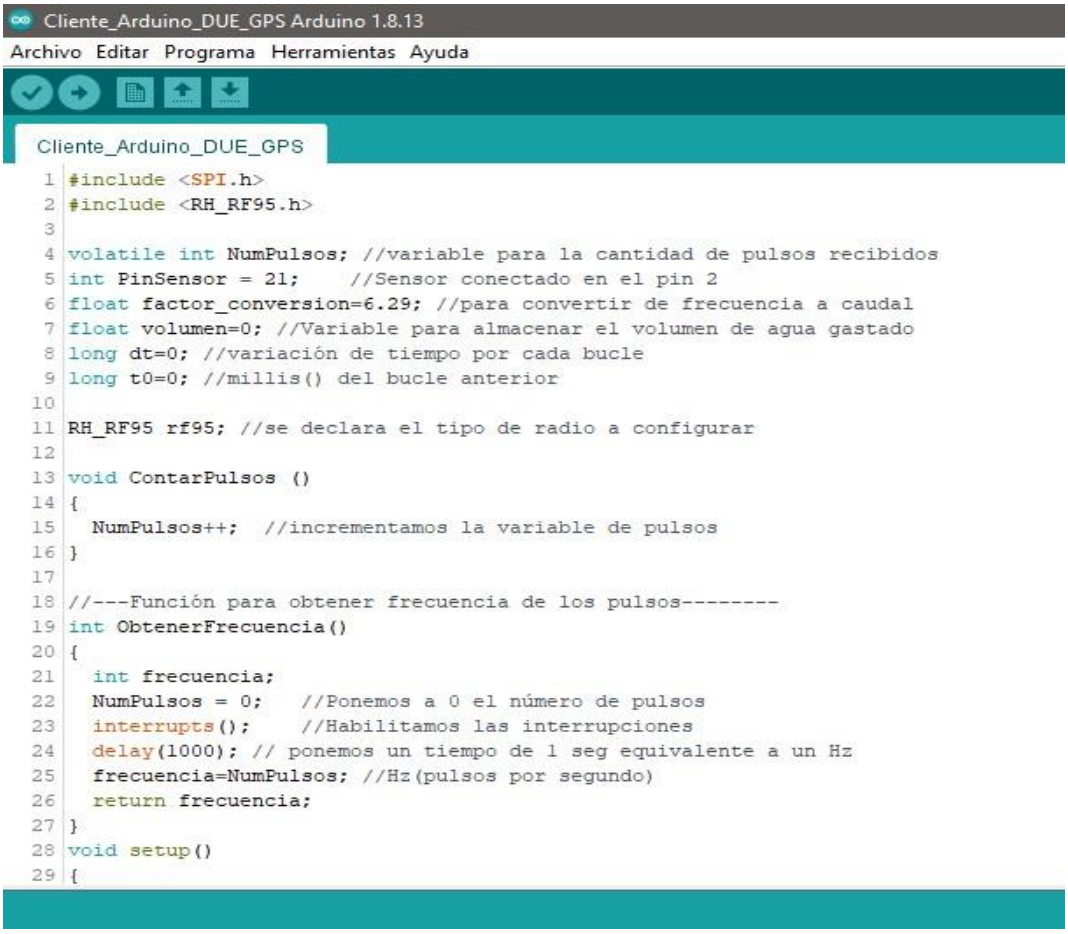

Figura 4 Nodo cliente (fuente: elaboración propia)

Para el nodo cliente también se incluye la librería SPI (Serial Peripheral Interface) y la librería del RH_RF95 en el cual se realiza la configuración de la comunicación LoRa. Al igual que el nodo servidor también se declara el tipo de chip que se utiliza y el poder de transmisión, también se inicia la comunicación y en void setup se declaran los puertos del Arduino a utilizar que en este caso la entrada digital pin D49 y la salida para la electroválvula se da por el pin D50 y D51, sin embargo, esta información de los pines que se utilizan se mostrará en la sección de diagramas.

Cliente_Arduino_DUE_GPS

```
28 void setup()
29 {
30
31   Serial.begin(115200); // Se inicia el puerto serial en 115000 baudios
32   pinMode(PinSensor, INPUT_PULLUP); //Declaramos el puerto para el sensor que hace la interrupción
33   attachInterrupt(digitalPinToInterrupt(PinSensor),ContarPulsos,RISING);//(Interrupción (PIN21),función,Flanco de subida)
34   t0=millis();
35
36   if (!rf95.init())// Se inicia el modulo
37   Serial.println("Inicialización fallida");
38  //Se setea en el archivo RH_RF95 915.0MHz, 13dBm, Bw = 125 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC on
39   rf95.setTxPower(23); // Configuro poder de transmisión
40   //Serial.print("Enviando datos del sensor al Arduino Leonardo GPS :  ");// Envio mensaje al servidor Arduino UNO
41 }
```

Figura 5 void setup nodo cliente (Fuente. elaboración propia)

En la figura 5 se puede ver el void setup, donde se configura la velocidad de conexión del puerto serial que en este caso es de 115000 y de esta manera comienza la comunicación del dispositivo SX1278 con la configuración de la librería RadioHead. También se puede observar que se utiliza attachInterrup que es el comando que se utiliza para leer el pulso de salida de sensor de flujo y cabe mencionar que por el tipo de funcionamiento del sensor(efecto Hall) se debe realizar por interrupciones de hardware que consiste en que si el sensor emite un pulso alto Arduino lo detecta por el puerto que permita interrupciones externas (Arduino Due permite interrupciones en todos sus puertos) y lo lleva a una subrutina donde se cuentan los pulsos y una vez se tenga se realiza la siguiente ecuación para averiguar el factor de conversión.

$$k = \frac{n° \, pulsos}{Volumen * 60}$$

Los números de pulsos se obtienen utilizando la tarjeta Arduino, un recipiente que tiene marcados los litros y un cronómetro, mediante el puerto serial se ven los pulsos que el sensor emite para llenar 10 litros del recipiente que fue utilizado; a continuación, se puede ver el programa que se utiliza para hallar el factor de conversión.

```
volatile long NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 20;    //Sensor conectado en el pin 2


//---Función que se ejecuta en interrupción---------------
void ContarPulsos ()
{
  NumPulsos++;  //incrementamos la variable de pulsos
}

//---Función para obtener frecuencia de los pulsos--------

void setup()
{
  Serial.begin(9600);
  pinMode(PinSensor, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(PinSensor),ContarPulsos,RISING);//(Interrupción 0(Pin2),función,Flanco de subida)
  interrupts();    //Habilitamos las interrupciones
}

void loop ()
{
  //-----Enviamos por el puerto serie----------------
  Serial.print ("Numero de Pulsos = ");
  Serial.println (NumPulsos);
  delay(100);
}
```

Figura 6 Código para hallar los pulsos (Fuente: elaboración propia)

Luego de hallar el factor de conversión se realiza el código pertinente para hallar el consumo total y el caudal en L/m, sin embargo, en la aplicación se envía el dato de caudal que va a ser enviado por comunicación LoRa al nodo servidor, en la figura 7 se puede observar el código que se realiza para obtener estos datos de consumo.

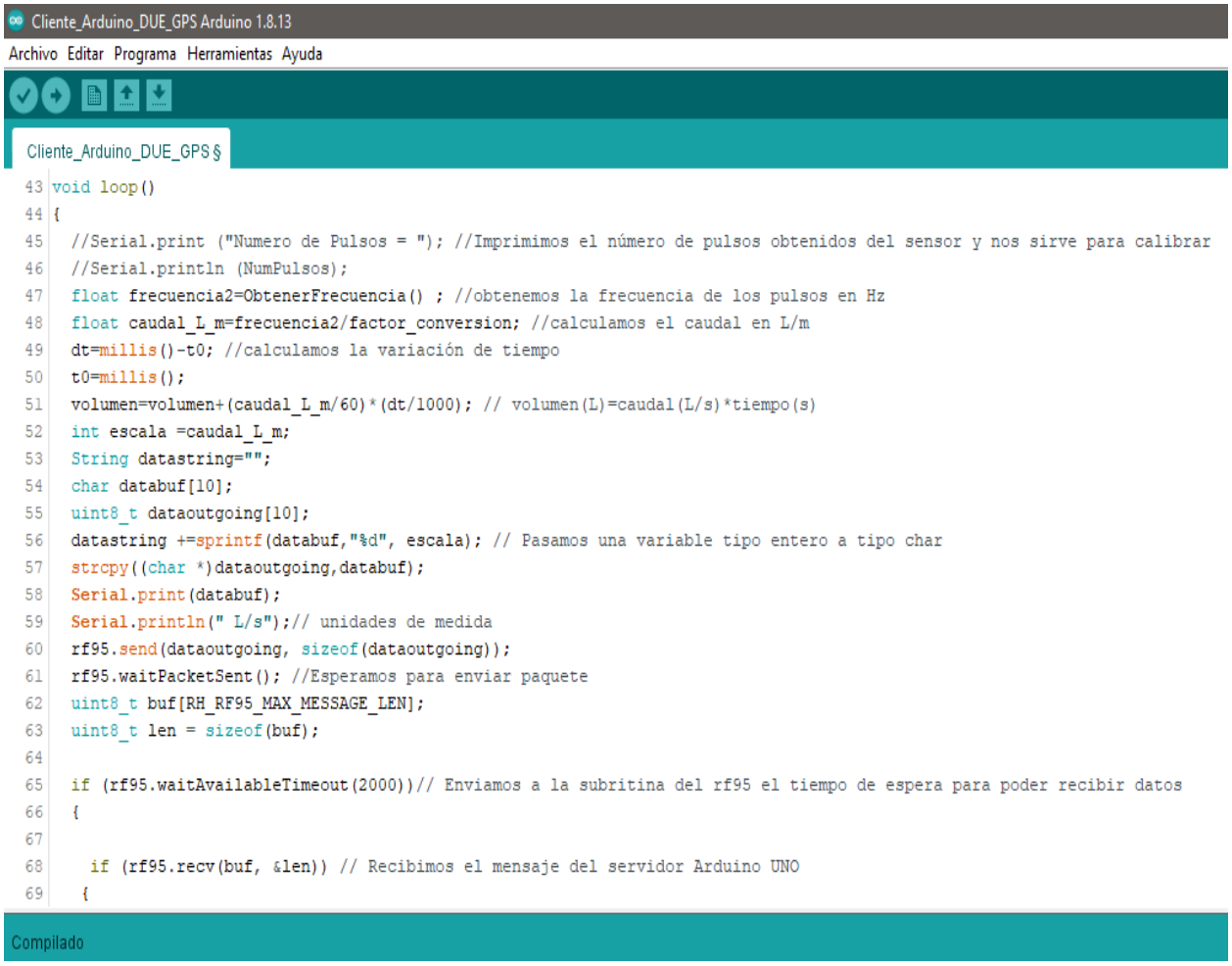

Figura 7 Void Loop nodo cliente (Fuente: elaboración propia)

Para hallar la frecuencia se utiliza una subrutina donde se emplean interrupciones para poder seguir con el envío de los datos ya obtenidos, ya que si no se realiza de esta manera la comunicación no tendría prioridad ya que el programa utiliza attachInterrump para obtener datos del sensor de flujo e ir a una subrutina para contar los pulsos, a continuación, se pueden ver en la figura las subrutinas desarrolladas.

```
13 void ContarPulsos ()
14 {
15   NumPulsos++;  //incrementamos la variable de pulsos
16 }
17
18 //---Función para obtener frecuencia de los pulsos--------
19 int ObtenerFrecuencia()
20 {
21   int frecuencia;
22   NumPulsos = 0;   //Ponemos a 0 el número de pulsos
23   interrupts();    //Habilitamos las interrupciones
24   delay(1000); // ponemos un tiempo de 1 seg equivalente a un Hz
25   frecuencia=NumPulsos; //Hz(pulsos por segundo)
26   return frecuencia;
27 }
28 void setup()
29 {
```

Figura 8 Subrutinas Arduino (Fuente: Elaboración propia)

La última parte del código es donde se envía el dato al nodo servidor y una de las partes más importantes que es el control de apertura y cierre de la electroválvula, en la figura 9 se puede ver que se actúan 2 relevos ya que el control de apertura y cierre es muy similar a como se realiza para el control del sentido de giro de un motor. Además de esto se observa el código donde se recibe la instrucción del nodo servidor en forma de datos de cadena de texto “char” ya que es la manera en la que hay menor pérdida de datos en la comunicación.

Cliente_Arduino_DUE_GPS §

```
64
65  if (rf95.waitAvailableTimeout(2000))// Enviamos a la subritina del rf95 el tiempo de espera para poder recibir datos
66  {
67
68    if (rf95.recv(buf, &len)) // Recibimos el mensaje del servidor Arduino UNO
69   {
70      //Serial.print("Respuesta de Arduino UNO : ");
71      Serial.println((char*)buf);
72
73      char respuesta=buf[0];
74      //Serial.print("Peticion ");
75      Serial.println(respuesta);
76      //Serial.println("Abriendo Válvula");
77      if(respuesta== '1' )
78      {
79
80        digitalWrite(52,LOW);
81        digitalWrite(53,LOW);
82        }
83        //Serial.println("Cerrando Válvula");
84        if(respuesta== '0' )
85      {
86        digitalWrite(52,HIGH);
87        digitalWrite(53,HIGH);
88        }
89       }
90    else
91    {
92      Serial.println("No se recibio comando");
```

Figura 9 Código para abrir o cerrar electroválvula (Fuente: elaboración propia)

Programación en Processing

El entorno de programación en processing es muy similar al IDE de Arduino, aunque maneja unas librerías distintas al Arduino tiene una estructura de programación muy similar, este software se utilizó para realizar la aplicación desde donde se pretende controlar la electroválvula y el sensor de flujo.

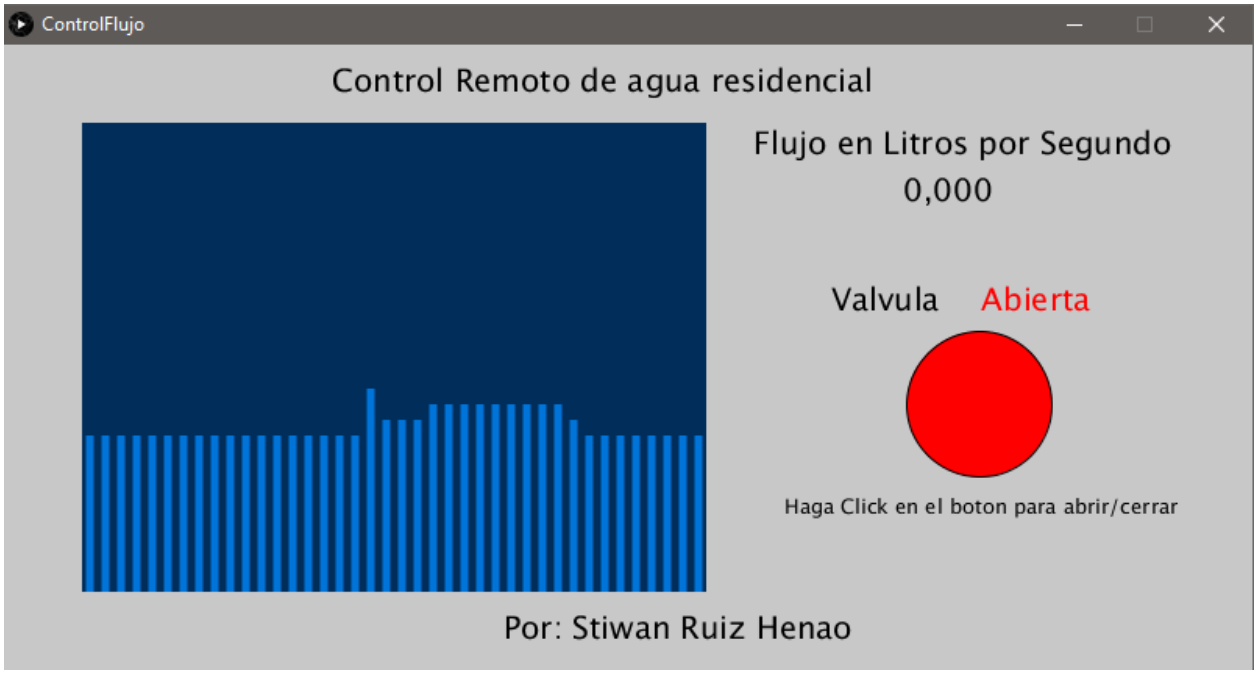

Figura 10 Aplicación en Processing (Fuente: elaboración propia)

Para la elaboración de la aplicación que se muestra en la figura anterior se incluye la librería “controlP5” y el puerto serial processing “ processing.serial”, con estas dos librerías se procede a incluir el botón para abrir y cerrar la válvula, la gráfica que muestra los datos y el dato de los litros por segundo.

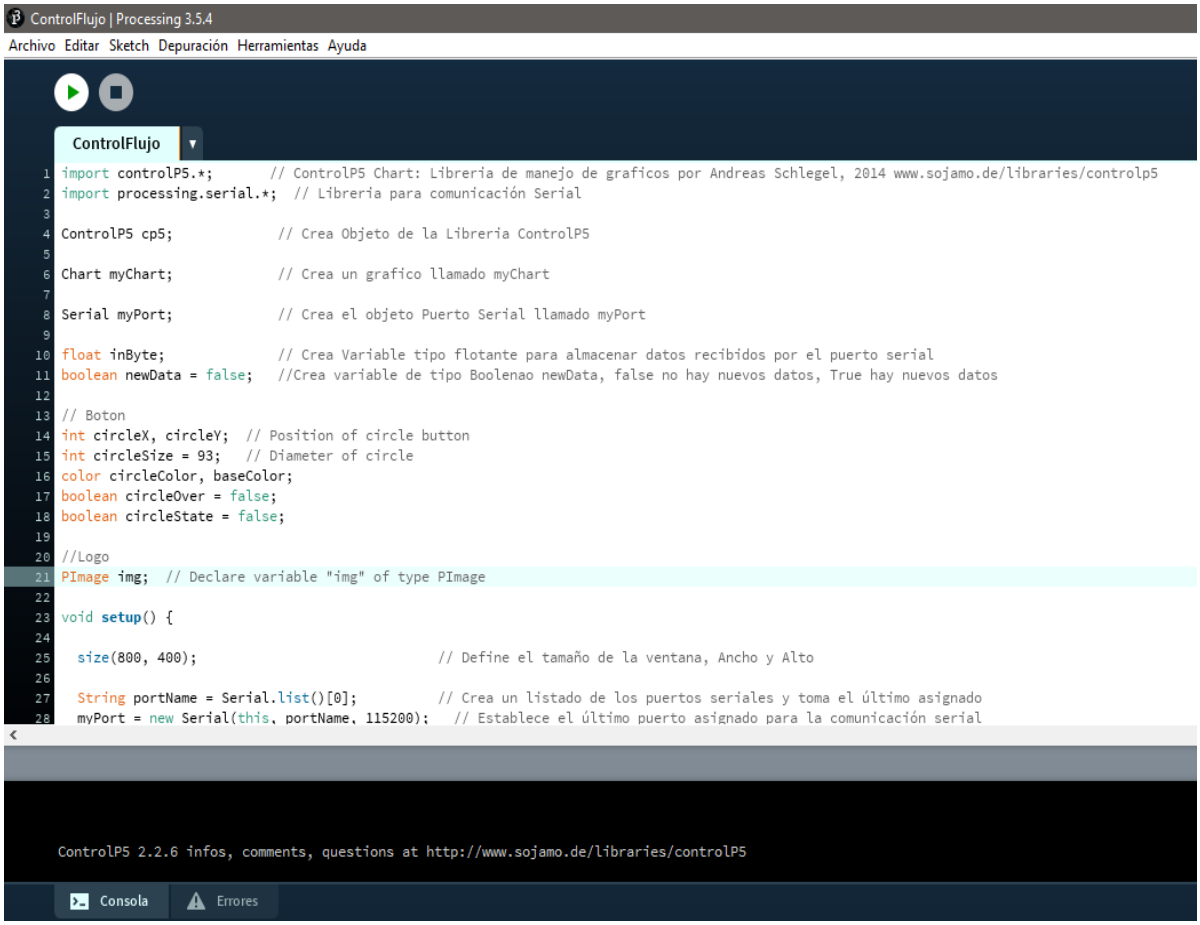

Figura 11 Programación Processing (Fuente: elaboración propia)

Como se puede observar también se declaran las variables donde se recibe el dato del puerto serial y la variable para actualizar el gráfico con los nuevos datos entrantes, adicional a esto, se leen los puertos seriales y se configura el tamaño con el que se muestra la aplicación.

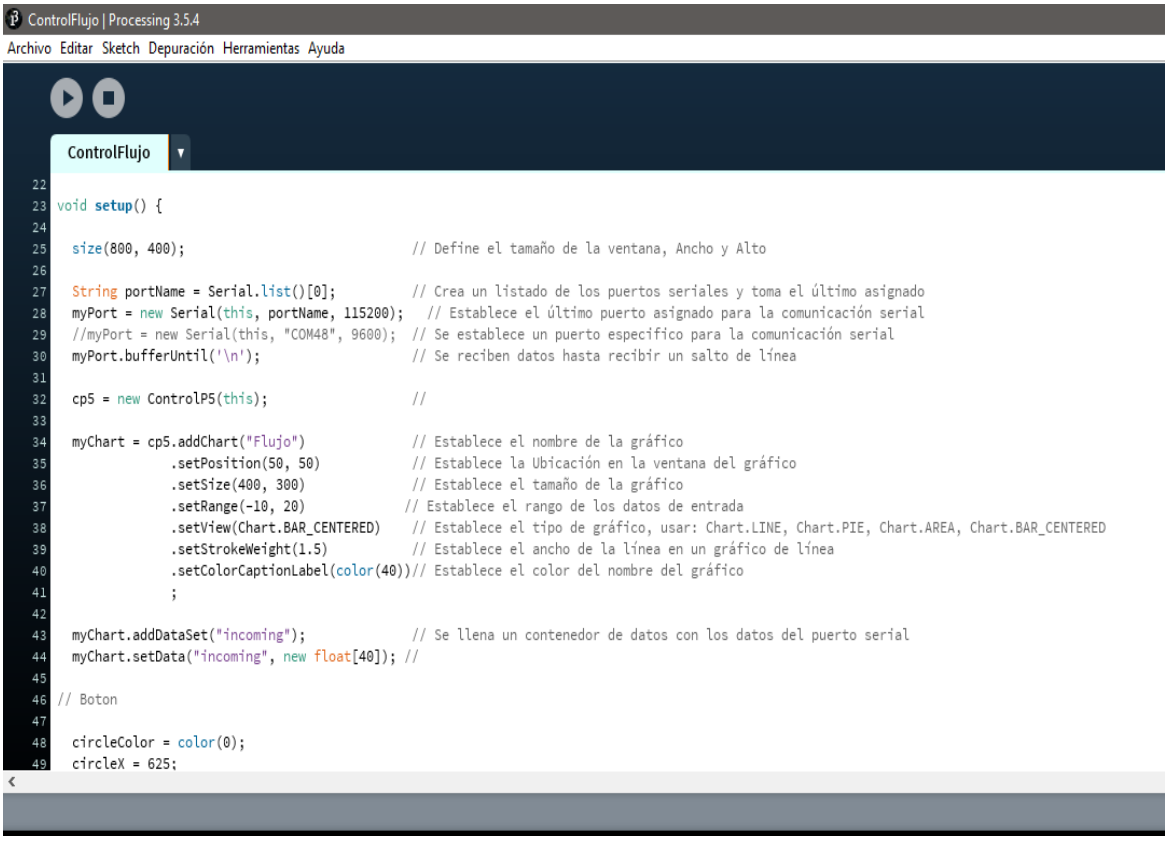

Figura 12 Ajuste de puerto serial y tamaño de aplicación (Fuente: elaboración propia)

En el resto de la programación se realiza el círculo para mostrar el botón, se muestra la gráfica y los diferentes títulos que se deben de poner para poder identificar la variable que se está midiendo.

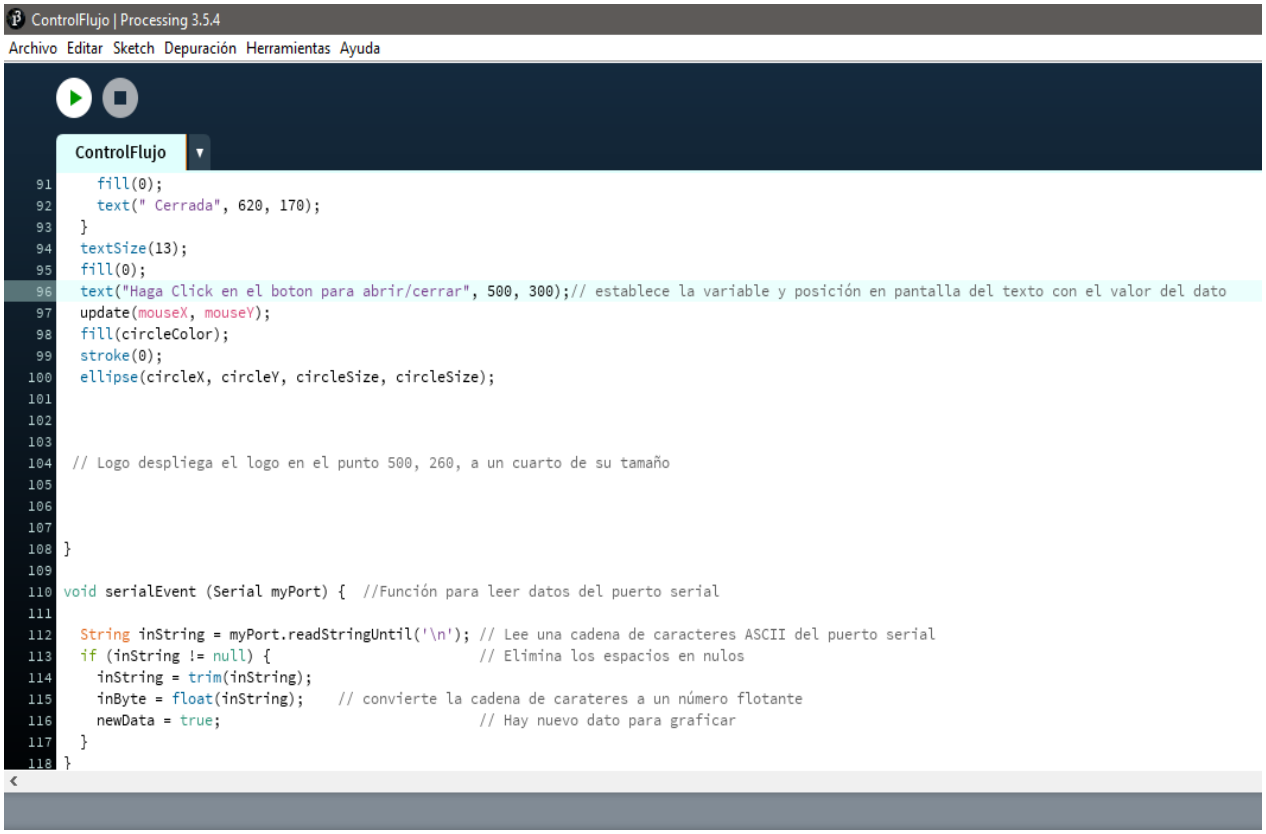

Figura 13 Dato puerto serial Processing (Fuente: elaboración propia)

Implementación y puesta en marcha de la solución

El proyecto consiste en la instalación en el tubo que llega de las derivaciones del tubo “madre” para suministrar agua potable a una vivienda y los elementos que lo componen son un sensor de flujo que permite tomar el consumo de agua de la vivienda y una electroválvula que permite cortar o habilitar el servicio de forma inalámbrica al igual que el dato de consumo. Para poder mostrar el funcionamiento del proyecto se realiza la instalación de los componentes en el tubo de agua del lavadero, a pesar de ser una simulación se puede observar el mismo comportamiento que se tendría si fuera instalado tal

cual está diseñado el proyecto.

Para el montaje de la electroválvula y del sensor de flujo se debe que tener en cuenta la dirección de flujo que muestran los 2 elementos, por lo tanto, se decide primero realizar el montaje de estos elementos con los accesorios de PVC antes de intervenir en el tubo del lavadero, esto con el fin de que los elementos queden alineados de la mejor manera y de esta manera tener una buena presentación de la simulación. En la figura 6 se puede ver que el elemento que en este caso tiene en su cuerpo marcado el sentido de flujo, de esta misma manera viene marcado la dirección en el sensor de flujo, lo cual se debe a que en la entrada de la electroválvula se tiene un filtro metálico para impedir que entren residuos que puedan obstruir la electroválvula y los demás elementos aguas abajo que se tengan instalados.

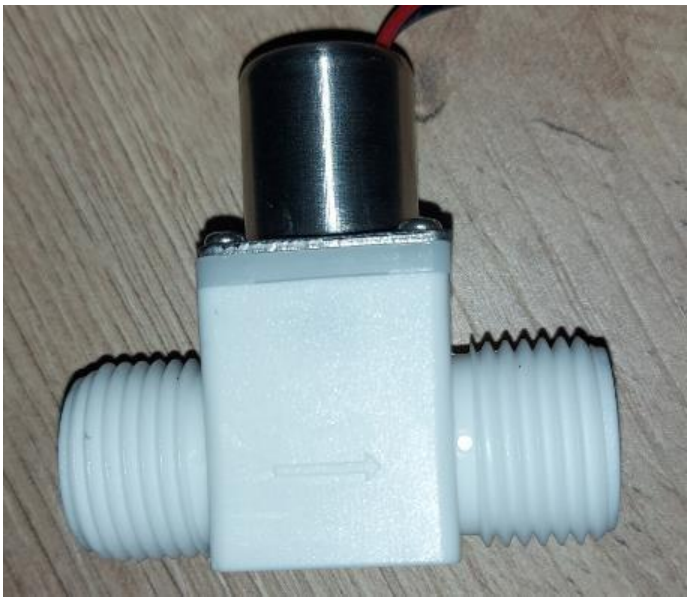

Figura 14 Electroválvula de1/2" (Fuente: elaboración propia)

En la figura 15 se puede apreciar el pre ensamble que se realiza utilizando los accesorios de PVC de ½" para instalar la electroválvula y el sensor de flujo, para esta

instalación se decide

no comprar una unión rosca-rosca ya que se debe dejar una distancia prudente entre el sensor de flujo y la electroválvula para que cuando haya un cierre no se genere impacto tan fuerte en las aspas del sensor de flujo, el cual se mide utilizando los principios del efecto Hall.

Figura 15 Pre ensamble electroválvula y sensor de flujo 1/2" (Fuente: elaboración propia)

Para la instalación de los elementos nivel 0 se corta el agua para poder realizar la conexión, una vez se hace, se decide cambiar el lugar de pruebas para un tubo cerca a la nevera con el cual hay un problema al quitar la canilla que está instalada, pero una vez se repara el tubo que se encuentra dañado se procede a instalar estos elementos y en la

siguiente figura se puede ver como la forma en la cual queda.

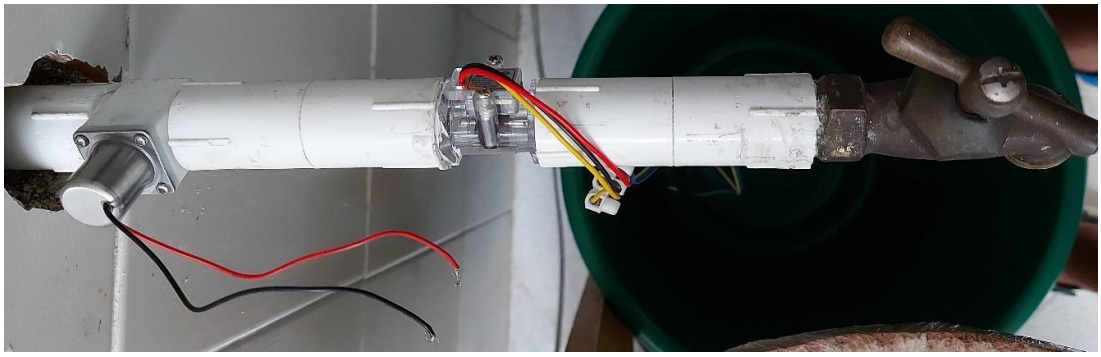

Figura 16 Sensor de flujo y electroválvula (Fuente: elaboración propia)

Luego de realizar la instalación de la electroválvula y sensor de flujo, se procede a realizar el ensamble del nodo cliente en una caja de 20 x 15 cm, en la cual se ubica la tarjeta Arduino DUE con el shield Dragino LoRa/GPS, fuente de 5VDC, relé y bornes de conexión de los cables entrantes, este nodo es el que recibe la señal del sensor de flujo y realiza el control sobre la electroválvula. En la figura 17 se puede visualizar cuales son los componentes que se tienen para el nodo cliente.

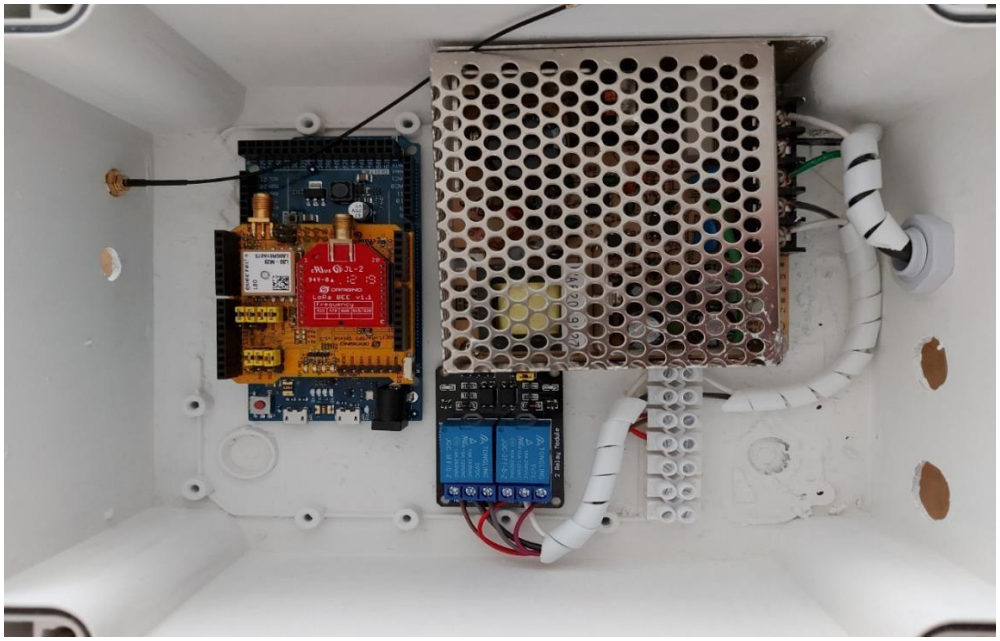

Figura 17 Caja nodo cliente (Fuente: elaboración propia)

Posterior el ensamble por partes, se procede a dejar la caja en el muro cerca al tubo y poner a funcionar los elementos con la alimentación directa de la fuente DC, una vez todo organizado se realizan varias pruebas en cuanto a la estabilidad de la conexión entre los nodos. Una vez que estas pruebas son satisfactorias se deja el sistema trabajando para verificar que no hay falsos pulsos cuando no hay flujo de agua y con esto se puede observar que el proyecto desarrollado cumple con los objetivos planteados y que se puede instalar en un sistema de real funcionamiento cumpliendo el objetivo general del proyecto, cabe mencionar que hay opciones de mejora en cuanto a los elementos utilizados para esta evaluación al sistema diseñado.

Diagramas de conexiones

El nodo cliente se implementa con una tarjeta LoRa/GPS Dragino y con una tarjeta Arduino DUE por sus múltiples puertos, los Shield de Dragino se conectan a Arduino por medio de la comunicación ICSP pero también se utilizan otros pines digitales por lo cual se debe utilizar una tarjeta de Arduino que tiene más entradas que aceptan interrupciones externas ya que las que tiene un Arduino UNO, Leonardo o estos modelos más básicos, estos pines que aceptan interrupciones externas están siendo utilizados por los Shield LoRa Dragino.

En la siguiente figura se puede ver los pines que son utilizados por el Shield Dragino LoRa/GPS:

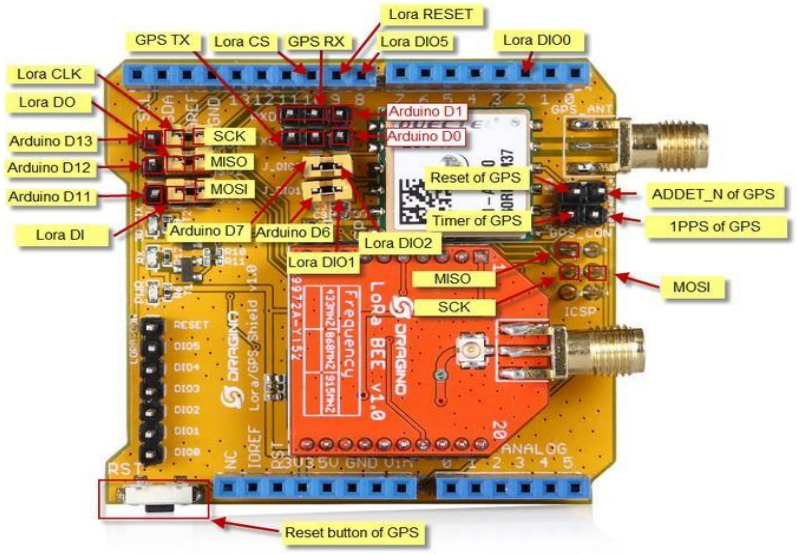

Figura 18 Conexión shield Dragino LoRa/GPS (Fuente: https://wiki.dragino.com/)

Como se mencionó anteriormente, el shield LoRa va encima del Arduino DUE y en la siguiente figura se puede ver la conexión del sensor de flujo y de la electroválvula que se conectan al Arduino DUE.

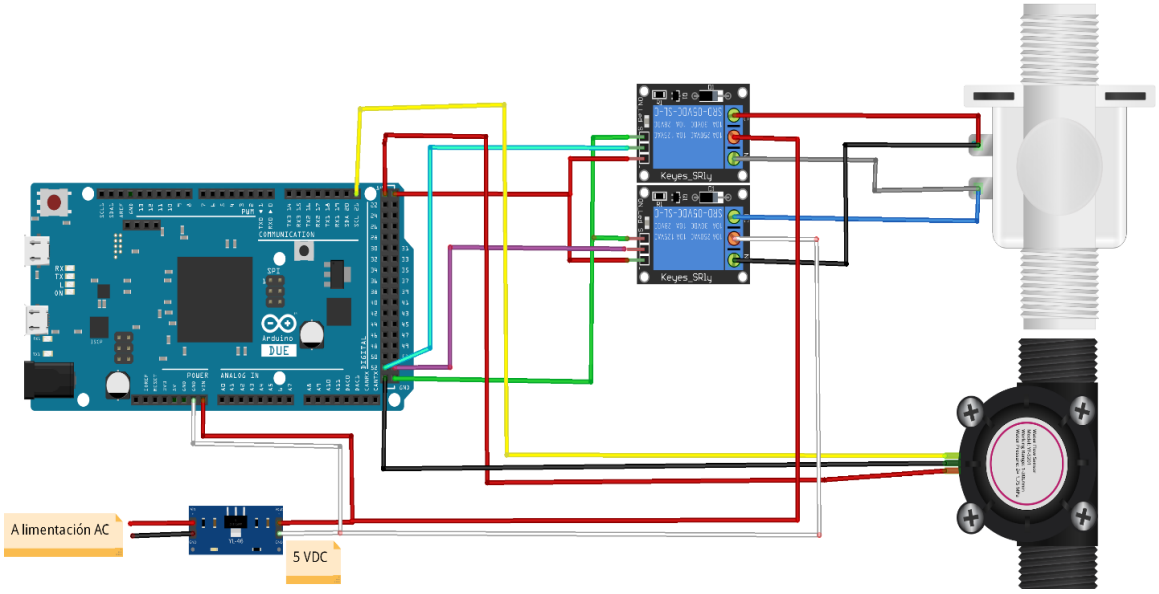

Figura 19 Diagrama de conexión nodo cliente (Fuente: elaboración propia)

El nodo servidor es el encargado de recibir y enviar datos al nodo cliente, este nodo envía la señal de apertura o cierre de la electroválvula y recibe el dato del sensor de flujo, esta tarjeta Arduino UNO se conecta al computador que tenga la aplicación que fue realizada en Processing y mediante el puerto serial es posible ver los datos, bien sea en la propia aplicación o en un software capaz de mostrar los datos del puerto serie de Arduino, en la siguiente figura se pueden ver los pines que son utilizados por este shield con Arduino UNO:

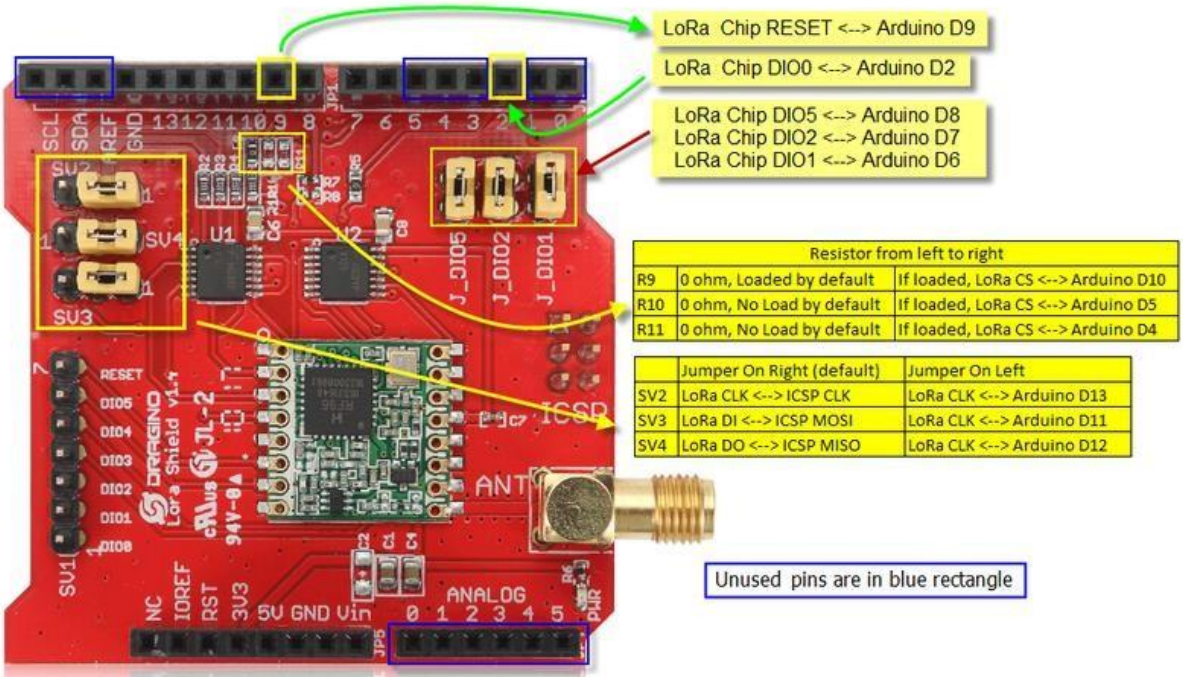

Figura 20 Conexión shield Dragino LoRa (Fuente: https://wiki.dragino.com/)

Presupuesto

Como se observa en el informe, el sistema de teleoperación se lleva a cabo utilizando tarjetas de bajo costo como Arduino y Shield Dragino LoRa, sin embargo, durante el desarrollo e implementación del sistema propuesto se determina que hay muchas

mejoras que se pueden realizar para este proyecto si se fuera a instalar en algún acueducto. A pesar de esto, se considera que es una solución a bajo costo y con muy buenos resultados.

A continuación, se presenta el listado de los materiales y sus precios en el mercado:

Tabla 3. Lista de materiales

Cantidad	Referencia	Valor total
1	Portátil ASUS X509J	$ 2’650.000
1	Dragino Shield LoRa 915 MHz	$ 89.250
1	Dragino Shield LoRa/GPS 915 MHz	$ 157.080
1	Arduino UNO	$ 28.560
1	Arduino DUE	$ 70.210
1	Fuente 110VAC-5VDC	$ 45.500
1	Electroválvula de ½” 5VDC	$ 29.750
1	Sensor de flujo de ½” 5VDC	$ 17.850
1	Reles 5VDC Optocoplado	$ 4.760
2	Caja plástica	$ 45.680
1	Paquete de cable de conexión dupont hembra macho	$ 2.142
1	Paquete de cable de conexión dupont hembra hembra	$ 2.142
4	Metros de cable 3x18	$ 21.400
Total		$ 3’164.324

Listado de materiales (Fuente: elaboración propia)

Resultados obtenidos

Durante el desarrollo del proyecto se presentan varias dificultades en cuanto a la comunicación entre las tarjetas Shield Dragino LoRa y en muchas ocasiones se debe desarrollar un nuevo código para poder tener comunicación entre estas tarjetas, sin embargo luego de enviar un correo al fabricante, se puede determinar que cualquier código que realice para trabajar con estas tarjetas debe tener la librería propia del chip SX1678 y una vez añadida esta librería a la carpeta de Arduino se deben configurar algunos parámetros como es la potencia de transmisión, Bandwidth, Spreading Factor,Coding Rate, Sensitivity Indication, LoRa demodulator. Una vez se tengan estos parámetros configurados se puede tener una comunicación LoRa estable y una de las recomendaciones que se presenta es inicialmente leer el manual del chip SX12XX para saber configurar estos parámetros que se han mencionado.

Para este proyecto se realizó una comunicación LoRa en ambiente interior y como bien se sabe hay más interferencias en este tipo de ambientes de trabajo, pero los resultados obtenidos son muy positivos y se realizaron algunas pruebas donde se trata de probar el sistema de comunicación de una habitación a otra y en ningún momento perdió comunicación, sin embargo cabe mencionar que la antenas con las que venían los shield LoRa fueron reemplazadas por unas de mayor ganancia para tener una mejor comunicación.

El proyecto como ya se mencionó se desarrolla en un ambiente interior donde la calidad de comunicación fue muy buena, por lo tanto, si esta solución se implementara en exterior teniendo una buena línea de vista entre las antenas y en el rango que el fabricante

estipula, se puede tener una buena comunicación y de esta manera poder crear múltiples soluciones a proyectos donde la distancia es un factor que hace difícil medir variables importantes como es el caso de las bocatomas de agua que normalmente están situadas en partes de difícil acceso ya que se busca que la toma del agua del río sea en las partes cercanas al nacimiento del mismo y casi siempre son lugares lejanos a la planta donde se realiza el proceso para convertir el agua entrante en potable.

Posterior al ensamble de las cajas con los nodos, se empezó a realizar pruebas las cuales fueron satisfactorias dando los resultados esperados, en la siguiente figura se muestra el nodo cliente y el nodo servidor.

Figura 21 nodo cliente (Fuente: elaboración propia)

En la figura 21 se puede evidenciar el funcionamiento del nodo cliente, que incluye Arduino DUE, relay 2 salidas y fuente de alimentación.

Figura 22 nodo servidor (Fuente: elaboración propia)

En la figura 22 se puede ver el nodo servidor conectado al computador desde el cual se está ejecutando la aplicación creada en Processing.

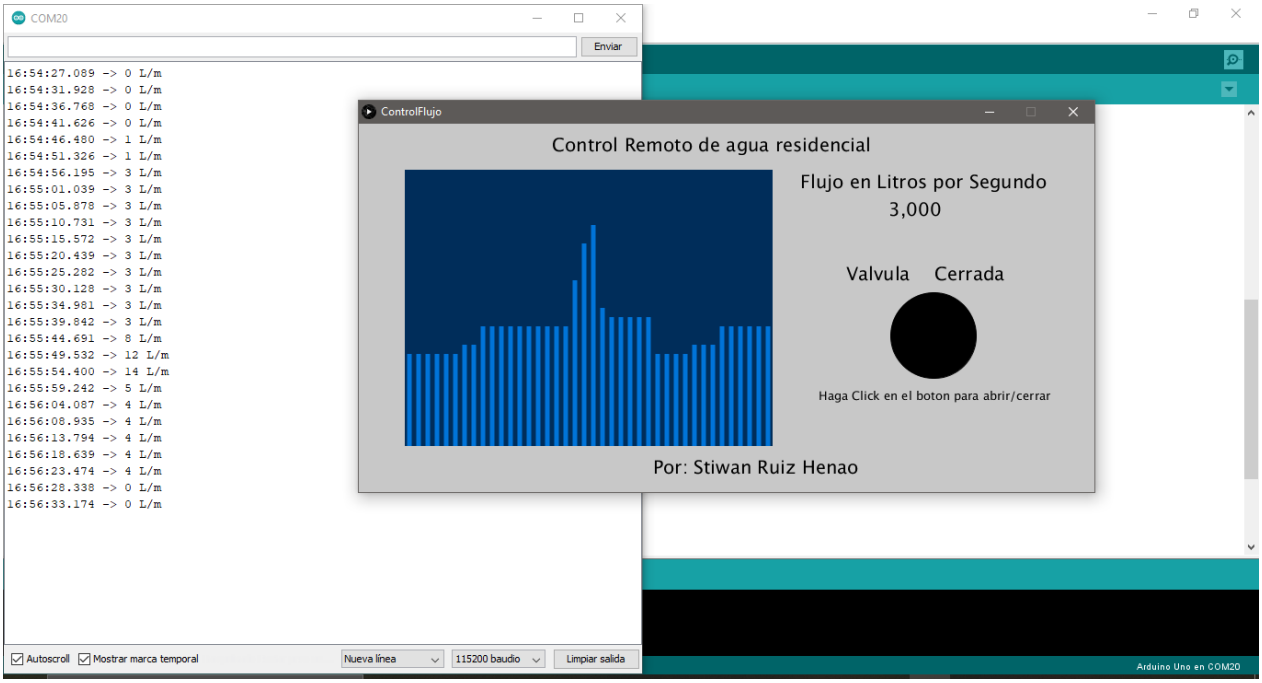

Figura 23 Aplicación processing válvula cerrada (Fuente: elaboración propia)

En la anterior figura se pueden ver los datos que muestra el nodo cliente conectado por el puerto serial en el IDE de Arduino y en la misma figura se aprecia la aplicación de Processing que es la que procesa el dato recibido y envía la señal para abrir y cerrar la válvula.

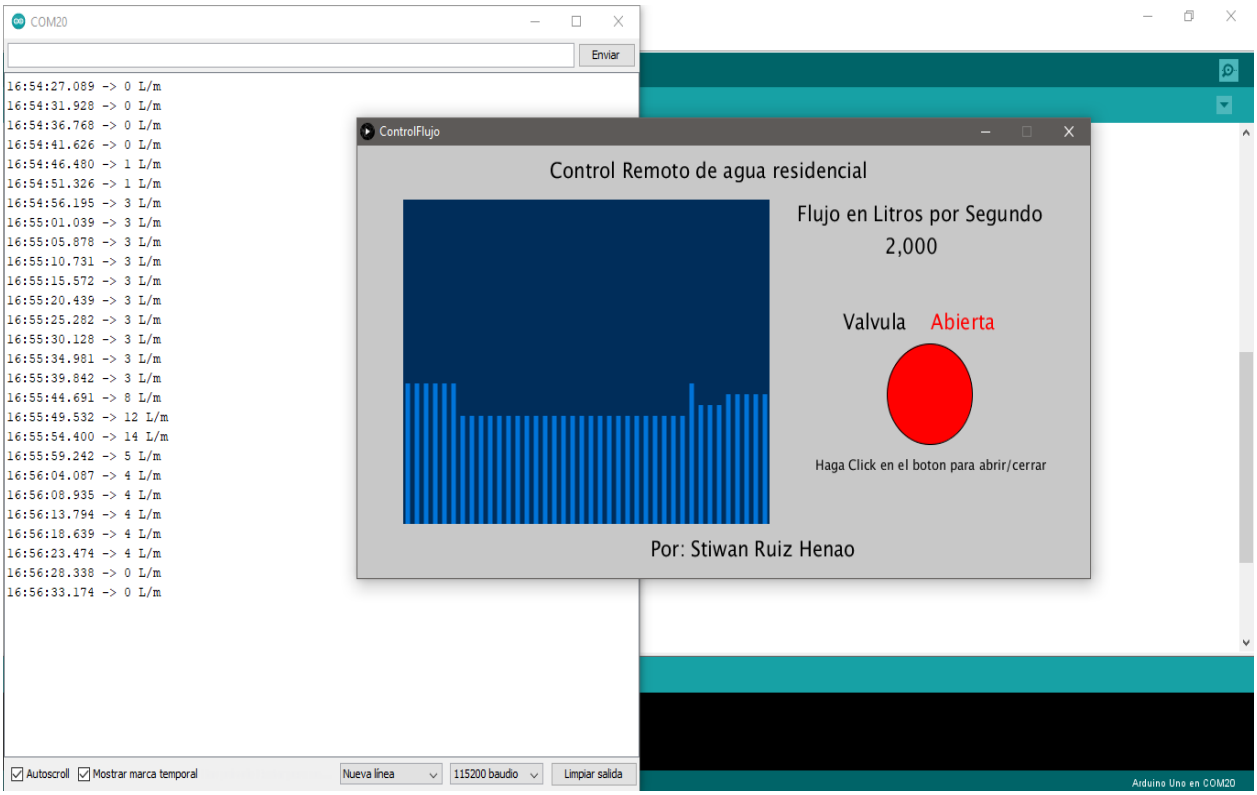

Figura 24 Aplicación processing válvula cerrada (Fuente: elaboración propia)

Trabajo Futuro

El proyecto trabaja con componentes electrónicos de bajo costo, pero siendo estos de muy buena calidad y con un gran rango de cobertura en cuanto a las tarjetas de comunicación Dragino LoRa, cabe mencionar que en el mercado se encuentran más tarjetas electrónicas con el mismo protocolo de comunicación, pero más costosas debido a los posibles periféricos que tengan que pueden ser pantallas, módulos GPS y otros sistemas de comunicación integrados en una misma tarjeta. Para este proyecto se decidió utilizar las tarjetas Arduino debido a que son tarjetas compatibles con los Shield de Dragino y son económicas y proporcionan seguridad para el desarrollo del proyecto, además de las tarjetas Shield LoRa de Dragino que proporciona comunicación de espectro extendido de alcance ultra largo y alta inmunidad a interferencias mientras minimiza el consumo de corriente (Shenzen Dragino,2020).

Cada día que se realizó pruebas de comunicación entre estas tarjetas de Dragino y Arduino, además de entender la importancia que tiene el desarrollo de este proyecto en las poblaciones ya mencionadas se observó que esta aplicación de comunicación combinada con una tarjeta programable, puede servir para el desarrollo de varias ideas donde se puede añadir paneles solares para la alimentación de los componentes y si existe la cobertura se puede añadir un módulo que cuente con comunicación a internet bien sea por medio de una Simcard o por la implementación de algún sistema de ethernet para el sector y de esta manera poder visualizar los datos desde un servidor web en cualquier parte que se desee acceder a esta información.

En el mercado actual hay diferentes dispositivos de medición de flujo que utilizan LoRaWAN para enviar los datos, esto evita tener que usar una tarjeta de comunicación y control aparte ya que el medidor o sensor lo tienen incluido, sin embargo, como se menciona en el párrafo anterior, la aplicación que se diseñó para el proyecto ofrece una mayor integración de dispositivos como pueden ser los totalizadores eléctricos que pueden ser controlados como es el caso de la electroválvula utilizada para este proyecto.

Conclusiones

De acuerdo a los resultados obtenidos se hace mención de la importancia que se le puede dar a este proyecto, ya que es una solución que como se ha mencionado antes no solo sirve para tener control sobre el servicio de agua potable, también se puede incorporar a múltiples soluciones y lo destacado que tiene es el bajo costo de implementación y los diferentes sensores que se pueden utilizar con la tarjeta Arduino.

La implementación del proyecto trae beneficios muy grandes para las empresas prestadoras de servicios públicos y más en las zonas rurales donde en muchas regiones la prestación del servicio de agua potable no es bueno y en muchos casos la empresa prestadora del servicio no le es viable económicamente ofrecer este servicio ya que es difícil estar accediendo a zonas tan remotas o por los problemas sociales que se pueden presentar.

Es importante entender que el proyecto se emuló con dispositivos como el sensor de flujo y la electroválvula que en caso de ser aplicado en el entorno para el cual fue diseñado, se deben de utilizar instrumentos de mayor calidad y exactitud, sin embargo, los resultados obtenidos en esta emulación son muy acordes a lo que se esperaba.

Con el desarrollo del escrito de este trabajo de grado se observó que cuando se realiza un proyecto con esta finalidad, las ideas van surgiendo y cuando se ejecutan a consciencia se aprende cómo contar una gran experiencia por medio del desarrollo y el hallazgo de las diferentes dificultades que se presentan, todo lo anterior, fortalece el saber y

la capacidad de expresar claramente lo efectuado. Los conocimientos y experiencias obtenidas ayudan a entender lo que se debe tener en cuenta para el desarrollo del trabajo.

Glosario

LoRa: es una tecnología de modulación del tipo sread spectrum (amplio espectro), es tecnología inalámbrica y fue desarrollado por la empresa Semtech.

Chirp Spread Spectrum: Chirp según sus siglas “Compressed High Intensity Radar Pulse”, es una señal que la frecuencia aumenta o disminuye con el tiempo. Este tipo de señal es común en radares y sonares.

Shield: son placas de circuitos modulares que se montan unas encima de otras para dar funciones extras a plataformas de desarrollo programables.

Bandwidth: en redes de comunicación el ancho de banda (Bandwidth) corresponde a la cantidad de datos que se pueden transmitir en una unidad de tiempo.

Spreading Factor: es una característica de la comunicación LoRa, este parámetro decide cuantos chirps pueden ser enviados cada segundo.

Coding Rate: En LoRa se debe agregar una corrección de errores hacia adelante en la transmisión de datos, de esta manera se garantiza que la señal LoRa aguante interferencias breves.

attachInterrups: Es un tipo de interrupción por hardware y trabaja cada vez que se detecta un cambio en el pin de interrupción en la entrada de la tarjeta Arduino.

Bibliografía

Adelantado,F y Vilajosana(2017) . Entender los limites de LoRaWAN. Recuperado de: https://doi.org/10.1109/MCOM.2017.1600613

Carlsson, A., B, I. K., Franksson, R., & Liljegren, A. (2018). Measuring a LoRa Network: Performance, Possibilities and Limitations. Springer International Publishing. Recovered from: https://doi.org/10.1007/978-3-030-01168-0

Croce, D., Gucciardo, M., Mangione, S., Santaromita, G., & Tinnirello, I. (2018). Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance.. Recoverd from: https://ieeexplore.ieee.org/document/8267219

Ha, G. Y., Seo, S. B., Oh, H. S., & Jeon, W. S. (2019). LoRa ToA-Based Localization. Recovered from: https://ieeexplore.ieee.org/document/8939702

Maya, J. (2017). Regla de negocio 2017-RN-44. Recuperado de:https://www.epm.com.co/site/Portals/0/Decretos/RN-PROCESO-CXC-Y-GESTION- CARTERA.pdf?ver=2019-03-21-110759-707

Mesías, F. V., Ibarra Guillermo, P., Rodríguez, A. G., Laguna, A., Santana Suárez, M. del C., Granados, L., Guzmán, N., Sacananvoy, P. A., Gómez, M., & Figueroa, M. (2018).

Diagnóstico e identificación de problemas y objetivos, evaluación y selección de la mejor alternativa. Recuperado de: https://cra.gov.co/documents/Documento-AIN-FINAL-EDU- 2018.pdf

Risc, R. (2020). Implementación de Lora y Lorawan como escenario futuro de la industrias 4. 0 en el sector agroindustrial peruano. Recuperado de: https://www.usmp.edu.pe/campus/pdf/articulos/articulo_20.pdf

Rizzi, M., Ferrari, P., Flammini, A., Sisinni, E., & Gidlund, M. (2017). Using LoRa for industrial wireless networks. IEEE International Workshop on Factory Communication Systems - Proceedings

Robson, S., & Haddad, A. M. (2019). On the use of LoRa for Power Line Communication. Recovered from: https://ieeexplore.ieee.org/document/8893538

UNL EPM. (2018). Medidor de velocidad de diámetro nominal igual a 15 mm (½ pulgada) para agua potable. Recuperado de: https://www.epm.com.co/site/Portals/3/documentos/Aguas/ET_AS_ME07_01_Medidor_ de_velocidad_diametro_15_mm_para_agua_potable.pdf?ver=2018-11-28-150208-577

Zourmand, A. (Junio de 2019). Internet of Things (IoT) using LoRa technology. Recovered from: https://ieeexplore.ieee.org/abstract/document/8825008

Matondang, J (Agosto del 2018). Spreading Factor, Bandwidth, Coding Rate and Bit Rate in LoRa. Recovered from: https://josefmtd.com/2018/08/14/spreading-factor-bandwidth- coding-rate-and-bit-rate-in-lora-english/

Semtech(s.f.). Semtech SX1278. Recovered from: https://www.semtech.com/products/wireless- rf/lora-transceivers/sx1278

Llamas, l (Diciembre del 2018). Medir caudal y consumo de agua con arduino y caudalímetro. Recuperado de: https://www.luisllamas.es/caudal-consumo-de-agua-con-arduino-y- caudalimetro/

Electroensaimada(s.f.). SPI Arduino. Recuperado de: http://www.electroensaimada.com/spi.html#:~:text=Se%20entrara%20en%20detalle%20 del,un%20bit%20despu%C3%A9s%20de%20otro

Hernandez,R (Diciembre del 2019). ¿Qué es la tecnología LoRa y por qué es importante para IoT). Recuperado de: https://www.thethingsnetwork.org/community/santa-rosa/post/que- es-la-tecnologia-lora-y-por-que-es-importante-para-iot

Anexos

Anexo 1 Arduino UNO

Figura 25 Arduino UNO (Fuente: mercadolibre.com.co)

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm

Figura 26 Ficha técnica Arduino UNO (Fuente: www.arduino.cc)

Anexo 2 Arduino DUE

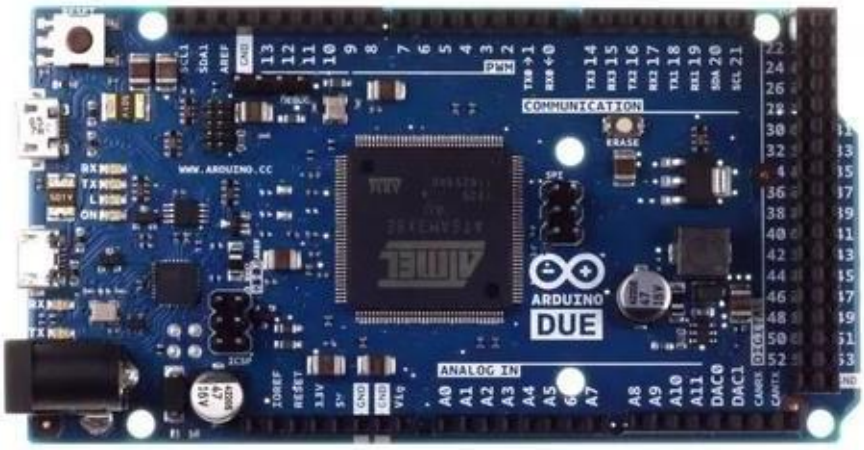

Figura 27 Arduino DUE (Fuente: www.mercadolibre.com.co)

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

Figura 28 Ficha técnica Arduino DUE (Fuente: www.arduino.cc)

Anexo 3 Fuente de alimentación 110VAC – 5VDC

Figura 29 Fuente 110VAC-5VDC (Fuente: www.mouser.mx)

Fabricante:	MEAN WELL
Categoría de producto:	Suministros de energía de conmutación
RoHS:	Detalles
Voltaje de salida-Canal 1:	5 VDC
Número de salidas:	1 Output
Potencia de salida:	25 W
Voltaje de entrada:	85 VAC to 264 VAC, 120 VDC to 370 VDC
Corriente de salida-Canal 1:	5 A
Estilo de montaje:	Chassis
Longitud:	99 mm
Ancho:	97 mm
Altura:	36 mm
Serie:	S-25
Marca:	MEAN WELL
Tipo de producto:	Switching Power Supplies
Cantidad de empaque de fábrica:	45
Subcategoría:	AC-DC Power Supply
Peso de la unidad:	390 g

Figura 30 Ficha técnica fuente 110VAC-5VDC (Fuente: www.mouser.mx)

Anexo 4 Shield Dragino LoRA/GPS para Arduino

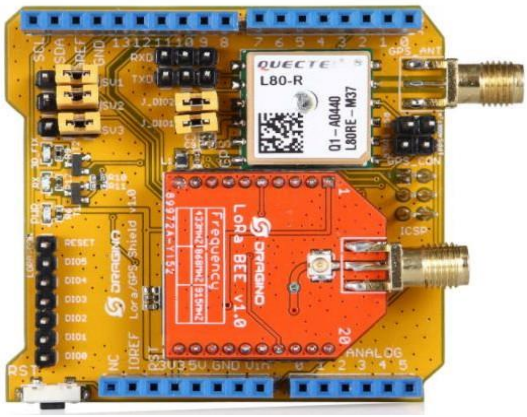

Figura 31 Shield Dragino LoRa/GPS (Fuente: www.dragino.com)

Lora Spec:

- 168 dB maximum link budget.
- +20 dBm - 100 mW constant RF output vs.
- +14 dBm high efficiency PA.
- Programmable bit rate up to 300 kbps.
- High sensitivity: down to -148 dBm.
- Bullet-proof front end: IIP3 = -12.5 dBm.
- Excellent blocking immunity.
- Low RX current of 10.3 mA, 200 nA register retention.
- Fully integrated synthesizer with a resolution of 61 Hz.
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
- Built-in bit synchronizer for clock recovery.
- Preamble detection.
- 127 dB Dynamic Range RSSI.
- Automatic RF Sense and CAD with ultra-fast AFC.
- Packet engine up to 256 bytes with CRC.
- Built-in temperature sensor and low battery indicator.

GPS Spec:

- Power Acquisition:25mA,Power Tracking:20mA.
- Compliant with GPS, SBAS.
- Programmable bit rate up to 300 kbps.
- Serial Interfaces UART: Adjustable 4800~115200 bps,Default: 9600bps.
- Update rate:1Hz (Default), up to10Hz.
- Protocols:NMEA 0183,PMTK.
- Horizontal Position Accuracy:Autonomous <2.5 m CEP.
- TTFF@-130dBm with EASY™:Cold Start <15s,Warm Start <5s,Hot start <1s;TTFF@-130dBm.
- without EASY™:Cold Start <35s,Warm Start <30s,Hot Start <1s.
- Timing Accuracy:1PPS out 10ns, Reacquisition Time <1s.
- Velocity Accuracy Without aid <0.1m/s,Acceleration Accuracy Without aid 0.1m/s².
- Sensitivity Acquisition -148dBm, Tracking -165dBm, Reacquisition -160dBm.
- Dynamic Performance Altitude Max.18000m, Maximum Velocity Max.515m/s, Maximum Acceleration 4G.
- L1 Band Receiver(1575.42MHz) Channel 22 (Tracking) /66 (Acquisition).

Figura 32 ficha técnica shield Dragino LoRa/GPS (Fuente: www.dragino.com)

Anexo 5 Shield Dragino LoRa para Arduino

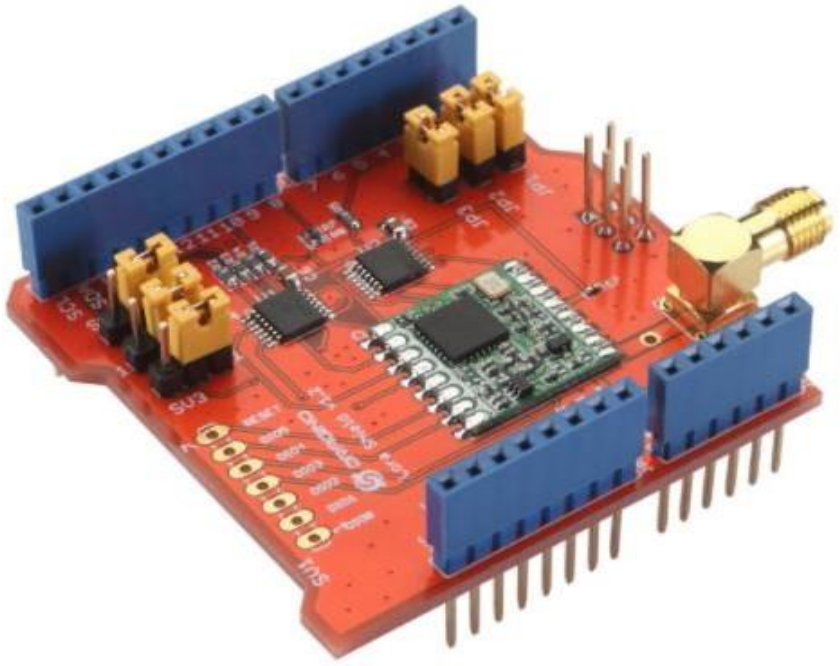

Figura 33 Shield Dragino LoRa (Fuente: www.dragino.com)

Summary

- 168 dB maximum link budget.
- +20 dBm - 100 mW constant RF output vs.
- +14 dBm high efficiency PA.
- Programmable bit rate up to 300 kbps.
- High sensitivity: down to -148 dBm.
- Bullet-proof front end: IIP3 = -12.5 dBm.
- Excellent blocking immunity.
- Low RX current of 10.3 mA, 200 nA register retention.
- Fully integrated synthesizer with a resolution of 61 Hz.
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
- Built-in bit synchronizer for clock recovery.
- Preamble detection.
- 127 dB Dynamic Range RSSI.
- Automatic RF Sense and CAD with ultra-fast AFC.
- Packet engine up to 256 bytes with CRC.
- Built-in temperature sensor and low battery indicator.

Figura 34 Ficha técnica shield Dragino LoRa (Fuente: www.dragino.com)

Anexo 6 Sensor de flujo de agua de ½”

Figura 35 Sensor de flujo (Fuente: https://www. botland.com.pl)

- Voltaje de funcionamiento: 5VDC ~ 24VDC máximo
- Corriente de funcionamiento: máximo 15mA (DC 5V)
- Rango de flujo: 1 ~ 30 L/min
- Presión del agua: ≤ 1.75MPa
- Pulso: F(Hz)=(5.0 x Q) ±3%, Q=L/min
- Load capacity: ≤ 10mA (DC 5V)
- Temperatura de funcionamiento: ≤ 80°C
- Temperatura del líquido: ≤ 120°C
- Humedad (operación): 35% ~ 90% RH
- Temperatura de almacenamiento: - 25°C ~ 80°C
- Humedad (almacenamiento): 25% ~ 95% RH

Figura 36 Ficha técnica sensor de flujo (Fuente: https://www.seeedstudio.com)

Anexo 7 Ficha técnica electroválvula plástica de ½”

Figura 37 Electroválvula plástica de ½” (fuente: www.didacticaselectronicas.com)

- Voltaje de funcionamiento: 3.6-6.5V
- Resistencia de la bobina: 9Ω
- Presión de funcionamiento: 0.05-1.0MPa
- Ancho de pulso: 30ms
- Corriente de funcionamiento: 500mA
- Rosca G1/2
- Conexión DN15mm macho
- Material plástico
- Formas de trabajo: válvula de apertura de pulso positivo, válvula de apagado de pulso negativo
- Vida útil: 30 millones de veces

Figura 38 Ficha técnica electroválvula de ½” (Fuente: didacticaselectronicas.com)

Anexo 8 Relé 2 salidas 5VDC optocoplado

Figura 39 Relé 5VDC optocoplado (Fuente: www.bigtronica.com)

Tabla 4 Ficha técnica rele 2 salidas 5VDC

Voltaje de Alimentación	5VDC
Salidas	2
Corriente contactos	10A
Voltaje contactos rele	12VDC, 110VAC, 220VAC
Señal de activación	En bajo (GND)

Tabla 4 ficha técnica rele 2 salidas 5VDC