

CONTROL LOCAL DEL MODULO DE MONITOREO Y CONTROL DE
INSTRUMENTACIÓN M2CI POR COMPUTADOR

EDUINSON YESID SALAS GONZÁLEZ
WILSON BERNARDO VILLEGAS ARIAS

UNIVERSIDAD NACIONAL ABIERTA Y DISTANCIA UNAD
ESCUELA DE LAS CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA HUILA
2015

CONTROL LOCAL DEL MODULO DE MONITOREO Y CONTROL DE
INSTRUMENTACIÓN M2CI POR COMPUTADOR

EDUINSON YESID SALAS GONZÁLEZ
WILSON BERNARDO VILLEGAS ARIAS

PROYECTO EXPERIMENTAL PARA OPTAR POR EL TITULO DE INGENIERO
ELECTRÓNICO

PEDRO TORRES SILVA
DIRECTOR DE PROYECTO
INGENIERO ELECTRÓNICO

UNIVERSIDAD NACIONAL ABIERTA Y DISTANCIA UNAD
ESCUELA DE LAS CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA HUILA
2015

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Neiva Huila (27, Septiembre, 2015)

DEDICATORIA

Este proyecto lo dedicamos a nuestros padres que con su esfuerzo y sacrificio, hemos culminado con éxito nuestros estudios universitarios, y al ingeniero Pedro Torres quien dedico parte de su parte de su tiempo en asesorarnos en el planteamiento del proyecto, en el diseño electrónico del sistema de control.

AGRADECIMIENTOS

Primero que todo queremos agradecerle a Dios creador del universo y dueño de nuestras vidas por habernos dado la fortaleza, sabiduría y oportunidad para llevar a cabo el desarrollo de este proyecto.

De igual forma queremos agradecer a nuestros familiares por su apoyo incondicional y a los ingenieros DIEGO FERNANDO SENDOYA LOSADA Y PEDRO TORRES SILVA, por ofrecernos la oportunidad de ser parte de este proyecto en el cual logramos aportar todos nuestros conocimientos en el desarrollo de sus actividades, y a aquellas personas que de una u otra forma, colaboraron o participaron en la realización de este proyecto, hacemos extensivo nuestros más sinceros agradecimientos

Contenido

LISTA DE TABLAS	8
LISTA DE DIAGRAMAS	8
LISTA DE GRÁFICOS	8
GLOSARIO	9
RESUMEN.....	10
ABSTRACT.....	10
PALABRAS CLAVE:	10
INTRODUCCIÓN	11
2. OBJETIVOS	12
2.1 OBJETIVO GENERAL.....	12
2.2 OBJETIVOS ESPECÍFICOS.....	12
3. PLANTEAMIENTO DEL PROBLEMA.....	13
3.1 DEFINICIÓN DEL PROBLEMA	13
3.2 JUSTIFICACIÓN.....	14
4. MARCO TEÓRICO	16
¿Qué es Arduino?	17
¿Por qué Arduino?	17
Arduino UNO	18
Información general.....	19
Programación.....	19
Advertencias	20
Diferencias con otras tarjetas.....	20
Alimentación	20
Memoria	21
Entrada y Salida	21
Comunicación	22
Reset Automático (Software).....	23
Referencia rápida para programar Arduino.....	23
Estructuras	23
Variables	23
Operadores booleanos.....	24

Operadores de comparación.....	24
Operadores matemáticos	24
Estructuras de control	24
Condicionales	25
Ciclos.....	25
Funciones.....	25
Funciones digitales.....	25
Funciones analógicas	26
5. MATERIALES Y MÉTODOS.....	27
5.1 MATERIALES.....	27
Dispositivos electrónicos.....	27
ARDUINO UNO	27
ETHERNET SHIELD	28
5.2 METODOLOGÍA	29
6. DESARROLLO DEL PROYECTO.....	30
Selección de Arduino Uno como interfaz entre Labview y tarjeta de control.....	30
Utilizando Arduino UNO para controlar 256 salidas digitales.....	30
Control de 8 LED utilizando comunicación serial	30
Prueba de funcionamiento sobre Planta de Nivel.	39
6. DESARROLLO DEL PROYECTO.....	41
6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO	41
6.2 CRONOGRAMA.....	41
CONCLUSIONES	42
RECOMENDACIONES.....	43
ANEXOS	45
Anexo A: circuito de simulación en Proteus.....	45
Anexo B: Montaje físico	
Anexo C: Link de descarga	46
Anexo D: Práctica de llenado de tanque y control de nivel	47

LISTA DE TABLAS

Tabla 1. Conexiones para la práctica 1. Planta piloto Nivel.....	39
Tabla 2. Listado de actividades	41
Tabla 3. Teach-in Rampa descendente ($A_1 > A_2$).....	47

LISTA DE DIAGRAMAS

Diagrama 1. Esquemático de la propuesta.....	35
Diagrama 2. Arduino en Proteus	36
Diagrama 3. Circuito conversor Serial-Paralelo	36
Diagrama 4. Circuito Demultiplexor.....	37
Diagrama 5. Circuito de memoria y registro	38
Diagrama 6. Diagrama circuital del control de nivel.....	39
Diagrama 7. Conexionado propuesto para el control de la planta de Nivel.....	40
Diagrama 8. Hardware general del proyecto.....	45
Diagrama 9. Diagrama de bloques del VI de la práctica 1.....	52

LISTA DE GRÁFICOS

Gráfico 1. Distribución de los pines en el ARDINO UNO.....	21
Gráfico 2. Uso del ARDINO para encender LED	31
Gráfico 3. Software de comunicaciones.....	31
Gráfico 4. Ejemplo de inserción y reconocimiento de un dato.....	33
Gráfico 5. Ejemplo de reconocimiento e inserción de más datos.....	33
Gráfico 6. Prueba con un dato mayor	34
Gráfico 7. Diagrama frontal de VI de la práctica 1. Ejemplo de llenado tanque 2.....	48
Gráfico 8. Panel frontal práctica 1. Ejemplo tanque 1 vacío.	49
Gráfico 9. Panel frontal práctica 1. En funcionamiento.	50
Gráfico 10. Panel frontal práctica 1. Llenado tanque 2 con electroválvula 2 accionada. ...	51
Gráfico 11. Exportación de datos de la práctica 1.	51

GLOSARIO

ACTUADOR: Es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.

ARDUINO: Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

CONMUTACION: conexión que realizan los diferentes nodos que existen en distintos lugares o espacios para lograr conectarlos entre sí.

ETHERNET SHIELD: "Arduino Ethernet Shield". Puede funcionar como servidor que acepta peticiones remotas o como cliente. Se permiten hasta cuatro conexiones simultáneas.

SHIELDS para Arduino: Las denominadas placas "Shield" son extensiones para tu placa Arduino (generalmente Arduino UNO o Arduino Pro 328) con lo que puedes expandir las posibilidades de la placa. Las hay de todo tipo, para módulos GSM, relés, comunicación con XBee y mucho más.

SISTEMAS: Es una combinación de componentes que actúan juntos y realizan un objetivo determinado.

RESUMEN

Para el desarrollo del proyecto se planteara varias fases con la finalidad de lograr de una manera más veras la realización y ejecución del proyecto, comenzando por una fase de planteamientos y diseños, en la cual servirán para estructurar de manera general cada uno de los procedimientos y procesos que se llevaran a cabo durante la realización del proyecto, en otras fases se realizaran la parte de ejecución, operación y puesta en marcha de los dispositivos e instrumentos del módulo M2CI, especificando que para este proyecto se trabajara sobre la planta de monitoreo y control de nivel, en la cual de manera experimental se realizara el sistema de control que nos permita controlar los diferentes instrumentos necesarios para realizar un laboratorio de control de nivel, como son los actuadores, sensores, borneras de alimentación tarjeta de adquisición de datos y acondicionamiento de potencia en una siguiente fase se realizara el diseño lógico de la sistema de conmutación con relevos que permitirá la interconexión entre los instrumentos ya mencionados y en una última fase se integrara todas las fases anteriores donde se probaran y se pondrán en marcha el funcionamiento las plantas del módulo mediante una interfaz gráfica, visualizadas y controlada desde un computador.

ABSTRACT

For the project several phases in order to achieve a more truly way the realization and implementation of the project, beginning with a phase of approaches and designs, which will serve to structure generally each of the procedures are raised and processes that were carried out during the project, at other stages the part of implementation, operation and commissioning of the devices and instruments M2CI module will be made, specifying that for this project to work on the plant monitoring and control level, in which experimentally control system that allows us to control different tools to perform a laboratory level control is carried out, such as actuators, sensors, power terminal card data acquisition and conditioning power in a next phase the logical design of the system switching relays that allow the interconnection between the instruments mentioned above and in the last phase all previous phases where they prove to be integrated and will launch operation plants will be held module using a graphical, visual interface and controlled from a computer.

PALABRAS CLAVE:

Interfaz, comunicación, control, conmutación, planta, automatizar, Arduino, virtualización, laboratorios remotos, Labview.

INTRODUCCIÓN

Para el desarrollo del proyecto se planteara varias fases con la finalidad de lograr de una manera más veras la realización y ejecución del proyecto, comenzando por una fase de planteamientos y diseños, en la cual servirán para estructurar de manera general cada uno de los procedimientos y procesos que se llevaran a cabo durante la realización del proyecto, en otras fases se realizaran la parte de ejecución, operación y puesta en marcha de los dispositivos e instrumentos del módulo M2CI, en una siguiente fase se realizara el diseño lógico de la interfaz gráfica que permitirá el control y en una última fase se integrara todas las fases anteriores donde se probaran y se pondrán en marcha el funcionamiento las plantas del módulo mediante una interfaz gráfica, visualizadas y controlada desde un computador.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e implementar un modelo experimental de control local a la planta de nivel del módulo de Monitoreo y control de instrumentación que permita operar y controlar la planta mediante un computador, y que en futuros proyectos sea el modelo para el control de todo el modulo

2.2 OBJETIVOS ESPECÍFICOS

1. Identificar la estructura y funcionamiento del módulo M2CI.
2. Plantear un sistema de conmutación que nos permita encender y controlar de forma individual la planta de nivel que integra el módulo mediante computador.
3. Crear un sistema de mediante el uso de circuitos integrados y de tarjetas desarrolladas con módulos de Arduino que me permita realizar el control del sistema de conmutación.
4. Diseñar el sistema lógico y la representación gráfica del funcionamiento del módulo mediante el uso de la herramienta LABVIEW.

3. PLANTEAMIENTO DEL PROBLEMA

3.1 DEFINICIÓN DEL PROBLEMA

Unos de los cambios culturales y de vida en este siglo, es el ritmo laboral y social que las personas deben afrontar para cumplir con sus diversas actividades de cotidianidad, los cuales han estado acompañado de los avances tecnológicos y de comunicaciones, y han hecho que el hombre sienta la necesidad de ser más polifacético y no dedicarse solo a una actividad determinada, sino tener herramientas que le permitan dinamizar y simplificar las actividades a realizar.

Basado en lo anterior y en la modalidad de estudio de la UNAD [1], de manera virtual y a distancia y del desarrollo de futuros proyectos como es la virtualización y control remoto del módulo M2CI derivados a partir de este proyecto que se va desarrollar en la cual se pretende es automatizar el encendido, apagado y control de las plantas del módulo de monitoreo sin necesidad de recurrir a cableados manuales para poder operar las diferentes plantas, tal como se realiza actualmente, donde se requiere conectar de manera manual las diferentes alimentaciones de voltaje, las salidas y entradas de los instrumentos del módulo. Lo cual sería un problema para el futuro proyecto de virtualización ya que se va a requerir de mantener cableado todas las plantas, lo cual generaría inconvenientes con las practicas que se requieren realizar de manera presencial, pues se tendría que mantener las conexiones de manera permanente, configuradas para el acceso remoto, lo cual sería un gran problema tanto para las personas que quieran trabajar tanto de manera local, como de forma remota, ya que cualquier cambio local afectaría el control y acceso remoto.

3.2 JUSTIFICACIÓN

Basado en la descripción del problema y en la misión de la UNAD de ser una universidad de innovaciones didácticas y de equidad social, se ha planteado y se ha desarrollado este proyecto con lo cual se pretende fomentar la investigación y afianzar los semilleros de investigación como es el caso del SYSCON que pertenece al programa de electrónica y donde se han desarrollado muy pocos proyectos debido a recursos económicos y de tiempo que estudiantes y tutores no tienen.

La finalidad de este proyecto más que por la necesidad de adquirir y afianzar nuestros conocimientos, surge de la necesidad de ser precursores de un proyecto que permitirá compartir las herramientas y los instrumentos de manera equitativa con otros estudiantes que no tienen posibilidad de acceso con estos instrumentos para prácticas de electrónica como lo permite el M2CI, en la cual se puede desarrollar sistema de control de temperatura, de nivel de agua, de control de velocidad y posicionamiento y muchos otros sistemas de control que son utilizados en el área industrial, por lo que nuestra propuesta de proyecto a desarrollar estará basada en implementar un sistema de control que me permita activar, desactivar y controlar las diferentes plantas desde un computador sin necesidad de realizar cableados manuales, lo cual será la base para desarrollar futuros proyectos por parte de los nuevos alumnos y tutores que deseen virtualizar y remotizar el módulo M2CI [2]

Según la misión de la UNAD y teniendo en cuenta la metodología de estudio de manera virtual y que es el pilar de la educación de este siglo; desarrollar nuevos instrumentos o nuevas formas de adquirir y afianzar el saber, mediante el uso de las tecnologías de la información y la comunicación es lo que constantemente se está buscando para la universidad, en el ámbito comercial y académico pues va a permitir que haya un mayor interés por los personas que desean ingresar a la universidad, como es de conocimiento en la actualidad la innovación y el desarrollo tecnológico es lo que busca las empresas modernas pues la economía se ha hecho dependiente de la tecnología y la UNAD como empresa que ofrece los servicios de educación a distancia a priorizado este proyecto como forma de gestionar una educación de calidad y no excluyente.

Para desarrollar este proyecto se ha buscado el apoyo directo de la universidad quien ha gestionado en su mayoría el recurso económico mediante la adquisición de los instrumentos y herramientas como son el módulo M2CI [3], el software de simulación Labview, también ha dispuesto el recurso humano con la asignación de tutores que orientaran las diferentes fases del proceso que se realizara. Teniendo a disposición lo anterior se optara por plantear las diferentes ideas que se podrán trabajar sobre el instrumento principal que es el M2CI, pues debido a que este cuenta con diferentes plantas de simulación para el control de manera integrada e independiente que se puede trabajar con todas las plantas a la vez o de manera individual, pero la idea del proyecto es poder controlar desde un computador este módulo y poder interactuar con cada una y sobre esta realizar las practicas allí

asignadas por el tutor de acuerdo al curso que se encuentre viendo el estudiante de una manera más segura, ya que la operación de las plantas se realizara desde un PC.

Por último y como se mencionaba anteriormente se desea que este proyecto sea la plataforma de la virtualización total del módulo de monitoreo y control de instrumentación - M2CI.

4. MARCO TEÓRICO

A pesar de que nada puede compararse con la interacción real con las plantas físicas (laboratorios reales y presenciales), afortunadamente existen otras opciones que proporcionan a los alumnos la impresión de que están interactuando con plantas reales. Así, normalmente se entiende por laboratorio remoto aquél que existe y puede ser manipulado de forma remota a través de Internet, haciendo uso de Webcams, hardware específico para la adquisición local de datos y software para dar una sensación de proximidad con el equipamiento y por laboratorio virtual (local o remoto), o utilizando software informático genérico o específico para recrear el comportamiento de plantas de experimentación que sólo existen en ordenadores usados para la simulación.

La interacción directa con el equipamiento del laboratorio aporta una experiencia difícil de igualar dado que, además de las variables medidas, los alumnos perciben los experimentos con los cinco sentidos (vista, tacto, oído, olfato e incluso, a veces, gusto). También resulta de alto interés didáctico la resolución de problemáticas asociadas a la puesta a punto de los equipos, la configuración de los experimentos y la medida de las variables de interés, etc. Sin embargo es muy frecuente que los laboratorios permanezcan infrutilizados debido a diversas razones (por ejemplo, horario limitado de acceso al laboratorio, ausencia de personal encargado o mantenimiento del equipamiento en estado inadecuado). En esta situación, el acceso remoto restringido a un equipamiento real puede consistir en una solución muy interesante si se consigue paliar la ausencia del contacto directo con el equipamiento con tecnología: Webcams, micrófonos, hardware de adquisición de datos, etc.

Un referente muy mediato y semejante a la propuesta de proyecto que hemos planteado es el proyecto que ha realizado algunos integrantes del semillero de investigación de instrumentación y control que pertenece al CCAV Neiva, donde se realizó el proyecto de grado en el año 2013 llamado: Acceso y Control Remoto de la Planta de Temperatura del módulo de monitoreo y Control de Instrumentación donde solo se trabajó el acceso remoto a la planta de temperatura para controlar la temperatura del horno mediante un instrumento virtual desarrollado en Labview y que era la interfaz gráfica que manipulaba el usuario para realizar el control remoto y permitía configurar un punto de ajuste para alcanzar la temperatura ideal o deseada.

Otro proyecto donde se ha utilizado el mismo modulo ha sido el desarrollado por la red de emprendedores de la fundación Bavaria cuyo sistema está basado en una comunicación entre la instrumentación y un aplicativo realizado en ambiente grafico de Labview, y donde una parte de la comunicación se realiza por medio de la tarjeta de adquisición de datos USB además incorpora también un PLC. El aplicativo se basa en realizar monitoreo y control, local y remoto de las variables anteriormente mencionadas. En dicha instrumentación se caracterizan los instrumentos de forma virtual de tal manera que el estudiante puede analizar su

comportamiento dentro del proceso. Aunque este proyecto ha consistido en ser un entrenador de procesos para que los estudiantes conozcan cómo funcionan procesos similares en la industria.

¿Qué es Arduino?

Arduino es una plataforma [4] para el desarrollo de prototipos que se basa en hardware y software fáciles de usar y además es de código abierto. Las tarjetas Arduino son capaces de leer una entrada, como por ejemplo la luz en un sensor, el dedo sobre un botón, o un mensaje de Twitter, y convertirla en una salida – activación de un motor, encendido de un LED, publicación de algo en línea. El usuario puede decirle a la tarjeta qué hacer mediante el envío de un conjunto de instrucciones al microcontrolador instalado en la misma. Para ello se utiliza el lenguaje de programación de Arduino (basado en Wiring), y el software de Arduino (IDE), basado en Processing.

A través de los años Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta instrumentos científicos complejos. Una comunidad mundial – estudiantes, aficionados, artistas, programadores y profesionales – se ha reunido alrededor de esta plataforma de código abierto y sus contribuciones han añadido una increíble cantidad de conocimiento que puede ser de gran ayuda tanto para principiantes como para expertos.

Arduino nació en el Ivrea Interaction Design Institute como una herramienta sencilla para prototipado rápido, dirigida a estudiantes sin experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la tarjeta Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta desde simples tarjetas de 8 bits hasta productos para aplicaciones de IoT (Internet of Things), impresión 3D y entornos embebidos. Todas las tarjetas Arduino son de código completamente abierto, permitiendo a los usuarios construirlas de forma independiente y, eventualmente, adaptarlas a sus necesidades particulares. El software también es de código abierto, y está creciendo gracias a los aportes de los usuarios en todo el mundo.

¿Por qué Arduino?

Gracias a su sencilla y accesible experiencia de usuario, Arduino se ha utilizado en miles de diferentes proyectos y aplicaciones. El software de Arduino es fácil de usar para los principiantes, y lo suficientemente flexible para los usuarios avanzados. Se ejecuta en Mac, Windows y Linux. Los profesores y estudiantes lo utilizan para construir instrumentos científicos de bajo costo, para demostrar principios de química y física, o para empezar con la programación y la robótica. Los diseñadores y arquitectos construyen prototipos interactivos; los músicos y artistas lo utilizan para experimentar con nuevos instrumentos musicales. Arduino es una herramienta clave para aprender cosas nuevas. Cualquier persona – niños, aficionados, artistas, programadores – puede comenzar a experimentar

simplemente siguiendo paso a paso las instrucciones de un kit, o compartiendo ideas en línea con otros miembros de la comunidad de Arduino.

Hay muchas otras plataformas de microcontroladores disponibles para la computación física. Parallax Basic Stamp, BX-24 de Netmedia, Phidgets, Handyboard del MIT, y muchas otras ofrecen una funcionalidad similar. Todas estas herramientas toman los detalles confusos de la programación de microcontroladores y los envuelven en un paquete fácil de usar. Arduino también simplifica el proceso de trabajar con microcontroladores, pero ofrece algunas ventajas sobre otros sistemas para los profesores, estudiantes y aficionados interesados:

- **Asequible:** Las tarjetas Arduino son relativamente baratas en comparación con otras plataformas de microcontroladores. La versión menos costosa del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino pre-ensamblados cuestan menos de 50 dólares.
- **Multiplataforma:** El software de Arduino (IDE) se ejecuta en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los sistemas de microcontrolador se limitan a Windows.
- **Entorno de programación simple:** El software de Arduino (IDE) es fácil de usar para los principiantes, pero lo suficientemente flexible para que los usuarios avanzados lo aprovechen también. Para los profesores, se basa convenientemente en el entorno de programación Processing, por lo que los estudiantes que aprenden a programar en ese entorno estarán familiarizados con cómo funciona el Arduino IDE.
- **Código abierto y software extensible:** El software de Arduino está publicado como una herramienta de código abierto, disponible para la extensión por programadores experimentados. El idioma se puede ampliar a través de bibliotecas C++, y la gente con ganas de entender los detalles técnicos puede dar el salto desde Arduino al lenguaje de programación AVR C en el que se basa. Del mismo modo, puede agregar código AVR-C directamente en sus programas de Arduino si quiere.
- **Código abierto y hardware extensible:** Los planos de las tarjetas Arduino se publican bajo una licencia Creative Commons, por lo que los diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, ampliándolo y mejorándolo. Incluso los usuarios con poca experiencia pueden construir la versión prototipo del módulo con el fin de entender cómo funciona y ahorrar dinero.

Arduino UNO

Arduino UNO es la mejor tarjeta para empezar con la electrónica y la codificación, ya que es la placa más robusta con la que se puede empezar a experimentar, y es la más usada y documentada de toda la familia Arduino.

Información general

Arduino UNO es una tarjeta de desarrollo basada en el ATmega328P. Cuenta con 14 pines de entrada/salida digitales (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, un conector ICSP y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; simplemente se conecta a un computador con un cable USB, o se alimenta con un adaptador AC-DC, o con una batería. Se puede experimentar con Arduino UNO sin preocuparse demasiado por hacer algo mal, en el peor de los casos se puede reemplazar el chip por unos pocos dólares y empezar de nuevo.

"UNO" significa uno en italiano y fue elegido para conmemorar el lanzamiento del software de Arduino (IDE) 1.0. La tarjeta UNO y la versión 1.0 del software de Arduino (IDE) fueron las versiones de referencia de Arduino, que ahora ha evolucionado con nuevos lanzamientos.

Especificaciones técnicas

Microcontrolador	ATmega328P
Voltaje de operación	5 V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (límite)	6-20 V
Pines I/O digitales	14 (de los cuales 6 entregan salida PWM)
Pines I/O digitales PWM	6
Pines de entradas analógicas	6
Corriente DC por pin I/O	20 mA
Corriente DC para pin 3.3 V	50 mA
Memoria Flash	32 KB (ATmega328P) de los cuales 0.5 KB son usados por bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
Largo	68.6 mm
Ancho	53.4 mm
Peso	25 g

Programación

La tarjeta Arduino UNO se puede programar con el software de Arduino (IDE). El microcontrolador ATmega328 de la tarjeta viene pre-programado con un bootloader que permite cargar un nuevo código sin el uso de un programador de hardware externo. Se comunica mediante el protocolo STK500. También se puede

pasar por alto el bootloader y programar el microcontrolador a través del puerto ICSP (In-Circuit Serial Programming) utilizando el Arduino ISP.

Advertencias

La tarjeta Arduino UNO tiene un multi-fusible que protege a los puertos USB de su computador contra corto-circuitos y sobre-corrientes. Aunque la mayoría de los computadores tienen su propia protección interna, el fusible proporciona una capa extra de protección. Si se aplican más de 500 mA al puerto USB, el fusible rompe automáticamente la conexión hasta que el corto-circuito o la sobrecarga se eliminen.

Diferencias con otras tarjetas

La tarjeta Arduino UNO es diferente de todas las placas anteriores ya que no utiliza el chip controlador de USB a serial FTDI. En lugar de ello, cuenta con el Atmega16U2 (Atmega8U2 hasta la versión R2) programado como un convertidor de USB a serial.

Alimentación

La tarjeta Arduino UNO puede ser alimentada a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente. La alimentación externa (no USB) puede venir de un adaptador de AC-DC o de una batería.

La tarjeta puede funcionar con una fuente externa de 6 a 20 voltios. Sin embargo, si se suministra menos de 7V, el pin de 5V puede suministrar menos de cinco voltios y la tarjeta puede ser inestable. Si se utiliza más de 12 V, el regulador de voltaje se puede sobrecalentar y dañar la tarjeta. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son:

- **VIN.** Voltaje de entrada a la tarjeta Arduino UNO cuando se utiliza una fuente de alimentación externa (en lugar de los 5 voltios de la conexión USB u otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o, si se suministra tensión a través del conector, se accede a él a través de este pin.
- **5V.** Este pin saca 5V del regulador de la tarjeta. La tarjeta puede ser alimentada ya sea desde el conector de alimentación DC (7 – 12 V), el conector USB (5V), o por el pin VIN (7 – 12 V). El suministro de tensión a través de los pines de 5V o 3.3V no pasa por el regulador, y puede dañar su tarjeta.
- **3V3.** Un suministro de 3.3 voltios generados por el regulador de la tarjeta. El suministro máximo de corriente es de 50 mA.

- Serial: 0 (RX) y 1 (TX). Se utilizan para recibir (RX) y transmitir (TX) datos TTL en serie. Estos pines están conectados a los pines respectivos USB-Serial TTL del chip ATmega8U2.
- Interrupciones externas: 2 y 3. Estos pines pueden configurarse para activar una interrupción con un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Para mayor información se recomienda ver la función `attachInterrupt()`.
- PWM: 3, 5, 6, 9, 10, y 11. Proporcionan una salida PWM de 8 bits con la función `analogWrite()`.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan comunicación SPI utilizando la librería SPI.
- LED: 13. Hay un LED incorporado manejado por el pin digital 13. Cuando el pin es HIGH, el LED está encendido, cuando el pin es LOW, está apagado.
- TWI: Pin A4 o SDA y pin A5 o SCL. Soportan comunicación TWI utilizando la librería Wire.

La tarjeta Arduino UNO tiene 6 entradas analógicas, etiquetadas como A0 hasta A5, cada una proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto se miden desde 0 a 5 voltios, aunque es posible cambiar el límite superior de su rango usando el pin AREF y la función `analogReference()`.

Hay otra pareja de pines en la tarjeta:

- AREF. Tensión de referencia para las entradas analógicas. Se utiliza con `analogReference()`.
- Reset. Cuando esta línea es LOW se reinicia el microcontrolador. Normalmente se utiliza para añadir un botón de reinicio a las placas shield que bloquean la tarjeta.

Comunicación

La tarjeta Arduino UNO tiene una serie de facilidades para la comunicación con un computador, otra tarjeta Arduino, u otros microcontroladores. El ATmega328 ofrece comunicación serial UART TTL (5V), que está disponible en los pines digitales 0 (RX) y 1 (TX). Un ATmega16U2 en la tarjeta canaliza esta comunicación serial a través del USB y aparece como un puerto COM virtual para el software del computador. El firmware 16U2 utiliza los controladores USB COM estándar, y no se necesita ningún controlador externo. Sin embargo, en Windows, se requiere un archivo .inf. El software de Arduino (IDE) incluye un monitor serial que permite que datos de texto simples sean enviados hacia y desde la tarjeta. Los LED RX y TX de la placa parpadean cuando se están transmitiendo datos a través del chip USB-Serial y la conexión USB del computador (pero no para la comunicación serial en los pines 0 y 1).

La librería `SoftwareSerial` permite la comunicación serial en cualquiera de los pines digitales del Arduino UNO. El ATmega328 también permite la comunicación

I2C (TWI) y SPI. El software de Arduino (IDE) incluye una librería Wire para simplificar el uso del bus I2C. Para la comunicación SPI, se utiliza la librería SPI.

Reset Automático (Software)

En lugar de requerir que se presione el botón de reinicio antes de la carga del código, la tarjeta Arduino UNO está diseñada de manera que permite que sea reiniciada por el software de Arduino (IDE). Una de las líneas de control de flujo de hardware (DTR) del ATmega8U2/16U2 está conectada a la línea de reset del ATmega328 través de un condensador 100 nF. Cuando esta línea es LOW, la línea de reset cae lo suficiente para restablecer el chip. El software de Arduino (IDE) utiliza esta capacidad para permitir que se cargue el código con sólo pulsar el botón de carga en la barra de herramientas de interfaz.

Referencia rápida para programar Arduino

Arduino se programa en el lenguaje de alto nivel C/C++ y generalmente tiene los siguientes componentes para elaborar el algoritmo:

- Estructuras
- Variables
- Operadores matemáticos, lógicos y booleanos
- Estructuras de control (Condicionales y ciclos)
- Funciones

Estructuras

Son dos funciones principales que debe tener todo programa en Arduino:

```
void setup() {  
}
```

Código de configuración inicial, solo se ejecuta una vez.

```
void loop() {  
}
```

Esta función se ejecuta luego del setup(), se mantiene ejecutándose hasta que se apague el Arduino.

Variables

Es un dato o conjunto de datos que cambia su valor con la ejecución del programa.

Booleano	Entero	Carácter
true o false	Valor entero	Almacena un ASCII

<code>boolean encendido=true;</code>	<code>int conta=5;</code>	<code>char letra= 's';</code>
--------------------------------------	---------------------------	-------------------------------

Estos son algunos tipos de variables. Para conocer más tipos se recomienda visitar: arduino.cc/en/Reference/HomePage

Operadores booleanos

Usados generalmente dentro del condicional if. Por ejemplo, if (a || b)

- && (y)
- || (o)
- ! (negación)

Operadores de comparación

Usados generalmente dentro del condicional if y sobre el for y el while. Por ejemplo, if (a == b)

- == (igual a)
- != (diferente de)
- < (menor que)
- > (mayor que)
- <= (menor o igual)
- >= (mayor o igual)

Operadores matemáticos

Se aplican al manejo de variables, condicionales y ciclos. Por ejemplo, int valor = valor +5

- = (asignar)
- % (módulo)
- + (suma)
- - (resta)
- * (multiplicación)
- / (división)

Estructuras de control

Son instrucciones que nos permiten tomar decisiones y hacer diversas repeticiones de acuerdo a unos parámetros, dentro de las más importantes se destacan:

- if
- switch/case

- for
- while

Condicionales

Ideales para tomar decisiones luego de evaluar las condiciones lógicas:

if (Si)	switch/case (Casos)
<pre> if (entrada < 500) { // acción A } else { // acción B } </pre>	<pre> switch (var) { case 1: // acción A break; case 2: // acción B break; default: // acción C } </pre>

Ciclos

Ideales para repetir lo que se encuentre dentro de ellos.

for (Para)	while (Mientras)
<pre> for (int a = 0; a > 10; a++) { // acción a repetir } </pre>	<pre> while (var < 200) { // acción a repetir var++; } </pre>

Funciones

Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento. Las funciones son utilizadas para descomponer grandes problemas en tareas simples y para implementar operaciones que son comúnmente utilizadas durante un programa y de esta manera reducir la cantidad de código.

Cuando una función es invocada se le pasa el control a la misma, una vez que ésta finalizó con su tarea el control es devuelto al punto desde el cual la función fue llamada.

Funciones digitales

Orientadas a revisar el estado y la configuración de las entradas y salidas digitales.

pinMode()	digitalRead()	digitalWrite()
Permite configurar un pin	Leer un pin digital (0 ó 1)	Escribir un pin digital con 1 ó 0
pinMode(pin,modo)	<code>digitalRead(pin)</code>	<code>digitalWrite(pin,estado)</code>
pinMode (13,OUTPUT); pinMode (a,INPUT);	<code>int a = digitalRead (13);</code>	<code>digitalWrite (13,HIGH);</code> <code>digitalWrite (13,LOW);</code>

Funciones analógicas

Ideales para la lectura y escritura de valores analógicos.

analogRead()	analogWrite() → PWM
Leer un valor análogo 0 a 1023 analogRead(pin)	Escribir un valor análogo 0 a 255 <code>analogWrite(pin,valor de PWM)</code>
int a = analogRead (A0);	<code>analogWrite (9, 134);</code>

5. MATERIALES Y MÉTODOS

5.1 MATERIALES

1. Software
2. Labview 2012 versión estudiante
3. Componentes electrónicos

Cant	componentes	descripción
6	74HC04.	El circuito integrado 7404 cuenta con 6 inversores independientes con tecnología TTL. Cada inversor puede ser usado sin la necesidad de conectar los demás. Su salida es el estado inverso a su entrada
2	C.I 74HC154.	Demultiplexor de 4 a 16 líneas
64	C.I 74HC4066	4 interruptores bilaterales análogos
32	C.I 74HC573	Es un dispositivo que contiene 8 flipflops de tipo D los cuales se encuentra en una misma pastilla. Tiene alta velocidad, es un dispositivo CMOS con el detalle de ser compatibles con la tecnología TTL (LSTTL).
1	C.I 74HC574	Este consiste en una memoria de ocho bits síncrona con salida 3-state, compuesto por ocho flipflops que almacenan temporalmente el dato ingresado en la transición de niveles bajo a alto del reloj, y un pin de control output enable negado.
2	C.I 74HC595	Se define como un "registro de desplazamiento de 8 bits con entrada serie, salida serie o paralelo con latch ("Registro guardado") de 3 estados."Entonces puede usarse para controlar 8 salidas simultáneas usando unos pocos pines del microcontrolador (SERIAL).

Dispositivos electrónicos

ARDUINO UNO

Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños.

ETHERNET SHIELD [5]

Un Shield en Arduino son placas de circuitos modulares que se apila sobre el Arduino o sobre otro Shield, de forma que nos permite ampliar el hardware/capacidades de Arduino.

Las Shield se pueden comunicar con el Arduino bien por algunos de los pines digitales o analógicos o bien por algún bus como el SPI, I2C o puerto serie, así como usar algunos pines como interrupción. Además estas Shield se alimentan generalmente a través del Arduino mediante los pines de 5V y GND.

En muchos casos nos encontraremos que los Shield vienen sin soldar las patillas o en algunos casos sin algunos componentes soldados.

Cada Shield de Arduino debe tener el mismo factor de forma que el standard de Arduino con un espaciado de pines concreto para que solo haya una forma posible de encajarlo.

RELE SHIELD x 16 [6]

Tarjeta para el manejo de dieciséis relés, este módulo cuenta con una fuente suicheada, optoacopladores para cada relé y un circuito integrado ULN

Características:

- Modulo con relé de 250VAC/10A, 30VDC/10A
- Voltaje de funcionamiento 5~12VDC.
- Indicador de encendido (verde), e indicador de estado por cada canal.

Si fuese necesario se pueden desarrollar otras tarjetas que no dependan de la familia de Shield de Arduino cuya base sean solamente Relés [7] convencionales que permitan mayores manejos de corriente entre sus contactos.

Otros materiales

Cables: Diferentes configuraciones desde uses hasta jumper.
Diseño y construcción de los circuitos impresos.

5.2 METODOLOGÍA

Con la finalidad de ser coherentes en la búsqueda de alcanzar los objetivos propuestos, este proyecto se realizara en varias etapas:

- a. Reconocimiento de la estructura y funcionamiento de cada una de las plantas del módulo M2CI, donde se configurara sus sistemas de conexiones de alimentación (voltaje, corriente) sus datos de entrada y de salida, con la finalidad de tener el esquema de funcionamiento que servirá para desarrollar la automatización.
- b. Partiendo de que necesitamos conectar diferentes puntos, se optó por diseñar un sistema de conmutación mediante el uso de dispositivos electrónicos como los relés, que tiene la capacidad de controlar el encendido automatizado a las diferentes plantas de manera independiente con solo recibir un pulso de activación proveniente de una orden dada por un usuario desde una interfaz como se mencionara más adelante.
- c. Basado que necesitamos realizar las respectivas activaciones de cada uno de los relevos, se planteó que la forma que se podía lograr, era mediante un módulo de control basado en dispositivos que me pudieran codificaran las diferentes orden dada desde la interfaz de usuario, que nos tomaran esos datos y no los asignaran a cada uno de los relevos que se desean activar. Para lograr esto se seleccionó un multiplexor el cual nos permitirá de un conjunto de datos enviados, seleccionar el dato deseado y asignarlo en una salida, debido a que necesitamos almacenar ese dato deseado para que nos mantenga activado el relevo o el interruptor en la matriz de conexiones de los diferentes dispositivos, se optó por utilizar un latch tipo Data.
- d. Posterior a la realización del sistema de control se diseñó el sistema lógico y estructural mediante la herramienta de programación grafica Labview que nos permitirá desarrollar todo el funcionamiento del módulo pero de manera automatizada, y este será el interfaz de comunicación y de acceso que tendrá el usuario desde su computador, para acceder a las diferentes plantas.
- e. Para comprobar que el diseño tanto de la interfaz gráfica como el sistema de control y el de conmutación por relevos y por interruptores ha sido desarrollado de manera satisfactoria y cumple con los requisitos y objetivos de encender y controlar las diferentes plantas, se harán pruebas de manera individual a la planta de nivel del M2CI mediante la aplicación de una práctica básica de laboratorio, el cual será el indicador de que el proyecto propuesto en su fase de ejecución se ha realizado de manera satisfactoria.

6. DESARROLLO DEL PROYECTO

Para realizar la ejecución del proyecto estos fueron los puntos más significativos que se desarrollaron.

Selección de Arduino Uno como interfaz entre Labview y tarjeta de control

De acuerdo al diseño y a los parámetros necesarios para comunicarme con Labview y con un dispositivo de control en la que es necesarios solo enviar paquetes de datos pequeños como son los de 8 bits, el Arduino Uno fue el más adecuado debido al bajo costo, su diseño standard que lo hace fácil de utilizar, además tiene la cantidad necesarias de entradas y salidas para nuestro proyecto.

Utilizando Arduino UNO para controlar 256 salidas digitales

Como se ha observado hasta el momento, Arduino UNO tiene 14 pines que se pueden programar como entradas o salidas digitales. Sin embargo, para este proyecto se requiere el control de 256 salidas digitales, las cuales deben entregar un valor 0 o 1, dependiendo de la información suministrada de forma serial al Arduino. A continuación se presentará el diseño inicial y su correspondiente evolución para lograr este propósito.

Control de 8 LED utilizando comunicación serial

El primer intento para controlar varias salidas digitales, utilizando la información recibida de manera serial se presenta a continuación.

En este caso se utiliza el Arduino UNO para controlar el encendido y el apagado de 8 LED, dependiendo de la información recibida desde el Monitor Serie del software de Arduino (IDE). La comunicación serial del Arduino se realiza a través de los pines digitales 0 (RX) y 1 (TX). Para emular la comunicación serial se emplea el software "Free Virtual Serial Ports", el cual permite crear una pareja de puertos seriales virtuales. En este caso se crea la pareja COM2<->COM4. El puerto COM2 se conecta al Arduino UNO, y el puerto COM4 se conecta al Monitor Serie del software de Arduino (IDE). La configuración de los puertos seriales virtuales se muestra a continuación:

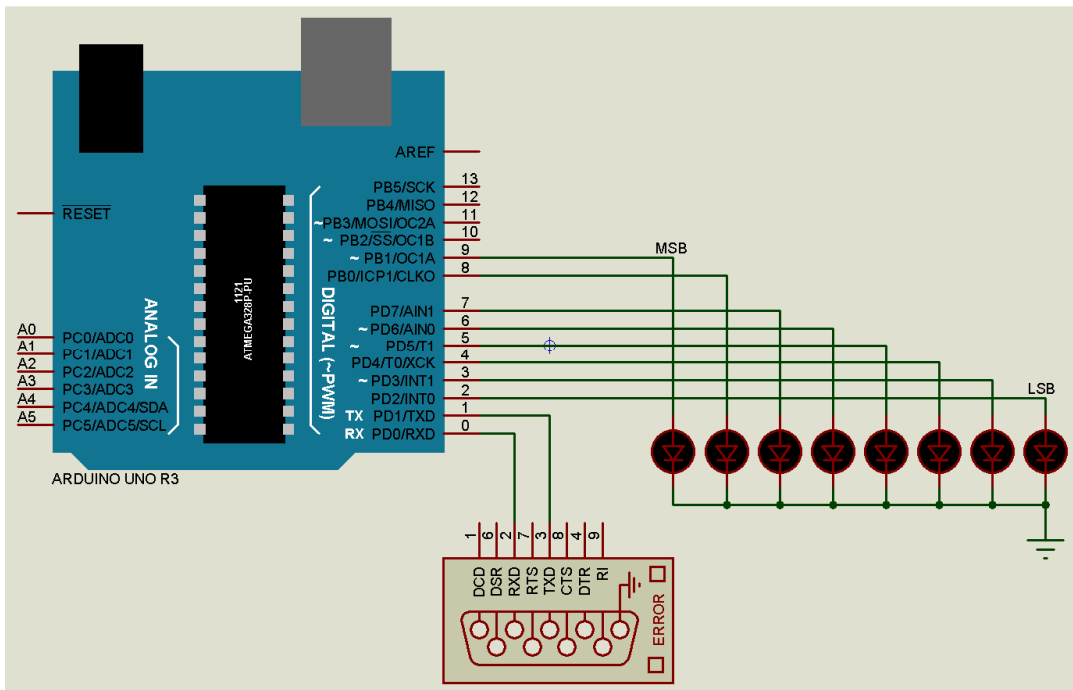


Gráfico 2. Uso del ARDUINO para encender LED

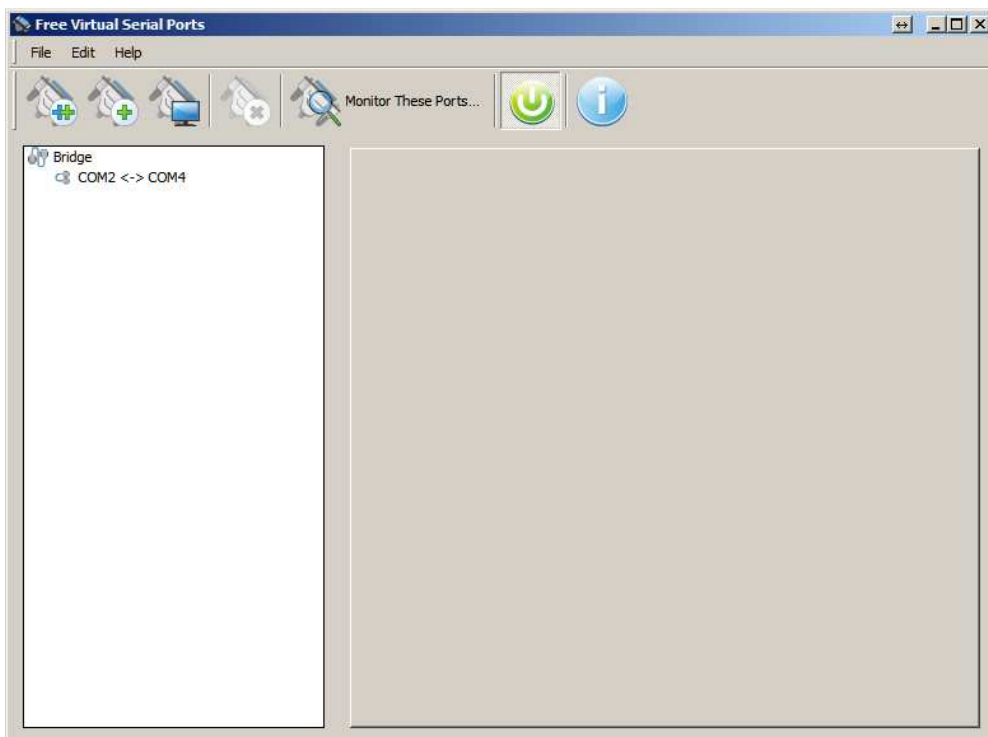


Gráfico 3. Software de comunicaciones

Los 8 LED se conectan a los pines digitales 2, 3, 4, 5, 6, 7, 8 y 9 del Arduino UNO, de manera que el bit de mayor peso (MSB) es el pin digital 9, y el bit de menor

peso (LSB) es el pin digital 2. A continuación se presenta el código Arduino que permite tomar la información recibida de forma serial y enviarla codificada en binario a los 8 LED de salida.

```
/*
 Este sketch utiliza la función parseInt() para leer, desde
 El Monitor Serie, un dato entre 0 y 255. Se utilizan 8 LED
 conectados a los pines 2, 3, 4, 5, 6, 7, 8 y 9 para visualizar
 la información en binario.
*/

void setup() {
 // se abre la comunicación serial con una velocidad de 9600
 Serial.begin(9600);
 // como se utilizará comunicación serial no se pueden usar
 // los bits 0 y 1, por lo tanto se establecen los pines
 // 2 a 9 como salidas. Siendo 9 el MSB y 2 el LSB.
 for (byte i = 2; i <= 9; i++) {
   pinMode(i, OUTPUT);
 }
}

void loop() {
 // se queda en espera de que aparezca algún dato por el puerto
 // serial, para ser leído
 while (Serial.available() > 0) {
   // el valor leído se almacena en la variable dato
   byte dato = Serial.parseInt();
   // si se presiona ENTER se saca el valor hacia los LED
   // y se muestra el valor en el Monitor Serie
   if (Serial.read() == '\n') {
     // se hace un ciclo activando cada LED de acuerdo al bit
     // correspondiente
     for (byte i = 0; i <= 7; i++) {
       // se verifica cada bit de la variable dato, iniciando por
       // el menos significativo. Si el bit es 1, la salida se
       // coloca HIGH. Si el bit es 0, la salida será LOW.
       digitalWrite(i + 2, bitRead(dato, i));
     }
     // el valor de la variable dato se muestra en binario
     // en el Monitor Serie
     Serial.println(dato, BIN);
   }
 }
}
```

También se presentan algunos resultados de las pruebas realizadas. Inicialmente se envía el dato 1:

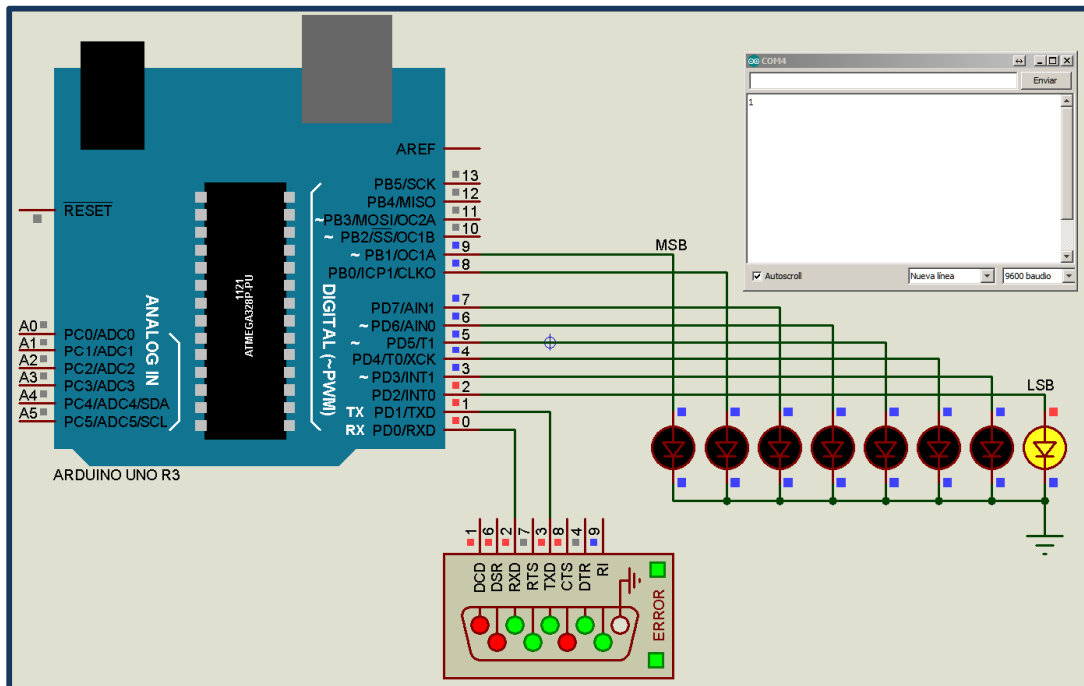


Gráfico 4. Ejemplo de inserción y reconocimiento de un dato

Luego se hace la prueba con el dato 20:

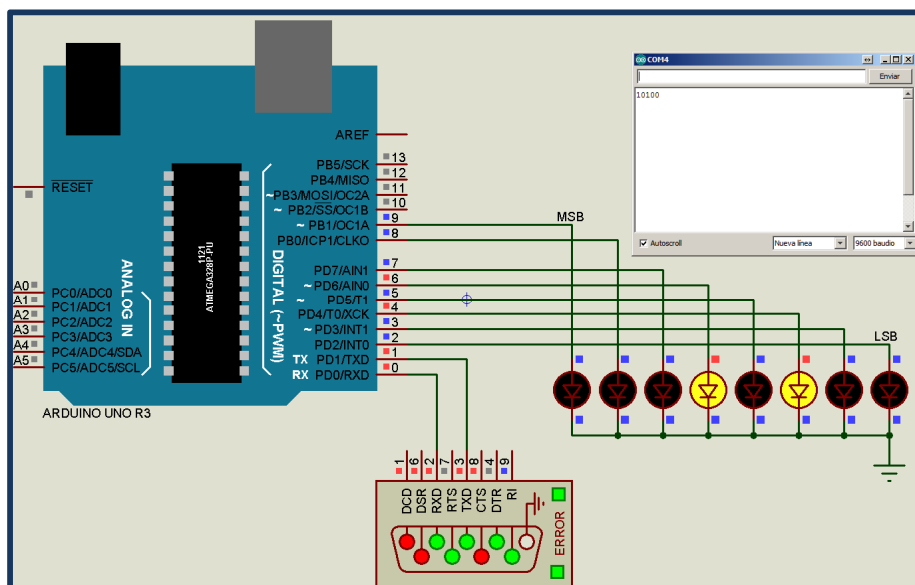


Gráfico 5. Ejemplo de reconocimiento e inserción de más datos

Finalmente, se hace la prueba con el dato 240:

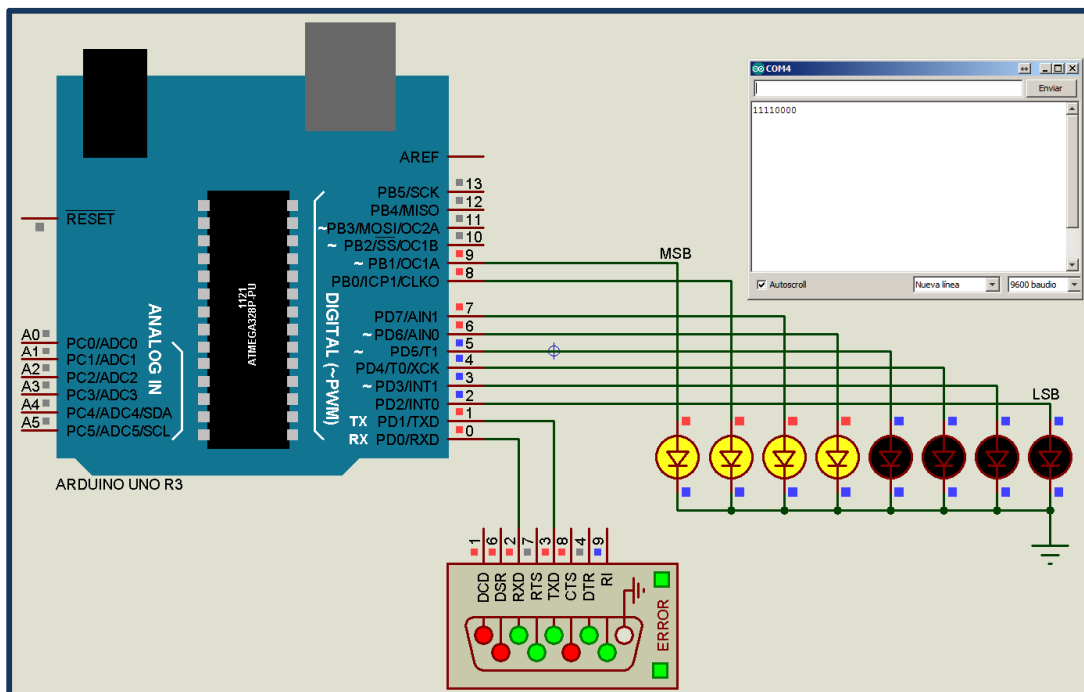


Gráfico 6. Prueba con un dato mayor

Se puede apreciar el correcto funcionamiento del diseño, sin embargo se quiere controlar 256 salidas digitales, y el Arduino UNO sólo tiene 14 pines que se pueden configurar como salidas. En este momento surge la pregunta, ¿cómo se puede utilizar la tarjeta Arduino UNO para controlar una mayor cantidad de salidas digitales?, la utilización de un registro de desplazamiento serie a paralelo de 8 bits puede ser la solución. En el mercado existen varias opciones, sin embargo vamos a optar por el 74HC595.

Incorporación de la Ethernet Shield como interfaz de comunicación remoto o local

Debido a que era necesario comunicar el Arduino con un ordenador en la que se tenga instalado el Labview, pensando en un futuro proyecto de automatización y remotización del M2CI se optó por implementarle como interfaz de comunicación la Ethernet Shield, ya que esta nos realiza la conexión por conexión red, tanto para un computador conectado mediante red LAN o también nos permita conexión mediante la internet y de su fácil conexión al Arduino.

Conexión Arduino a tarjeta de control

Como disponemos de un reducido número de salidas en el Arduino y por diseño y complejidad es más conveniente trabajar con menos conexiones, pero para este caso los puntos finales a controlar son numerosos, se hace necesario disponer

muchas salidas, para esto ampliaremos los pines de salida digital mediante la utilización del circuito integrado 74HC595 que permite convertir la salida serie del Arduino el cual envía 8 bits y esta las convierte en 8 salidas diferentes el cual nos amplía el número de salidas necesarias.

Ahora necesitamos seleccionar ese registro de datos para que seleccione los datos que queremos que queden habilitados como salidas es decir queden con un 1, para esto utilizaremos el circuito integrado 74HC154, que es un Demultiplexor, que nos permitirá habilitar o deshabilitar las salidas que deseamos.

Como ya tenemos habilitadas y deshabilitadas unas salidas, ahora será necesario que esas salidas me queden en estos estados, y me mantenga los relevos o los interruptores activados o desactivados hasta que se le dé una nueva orden para activar otro dispositivo

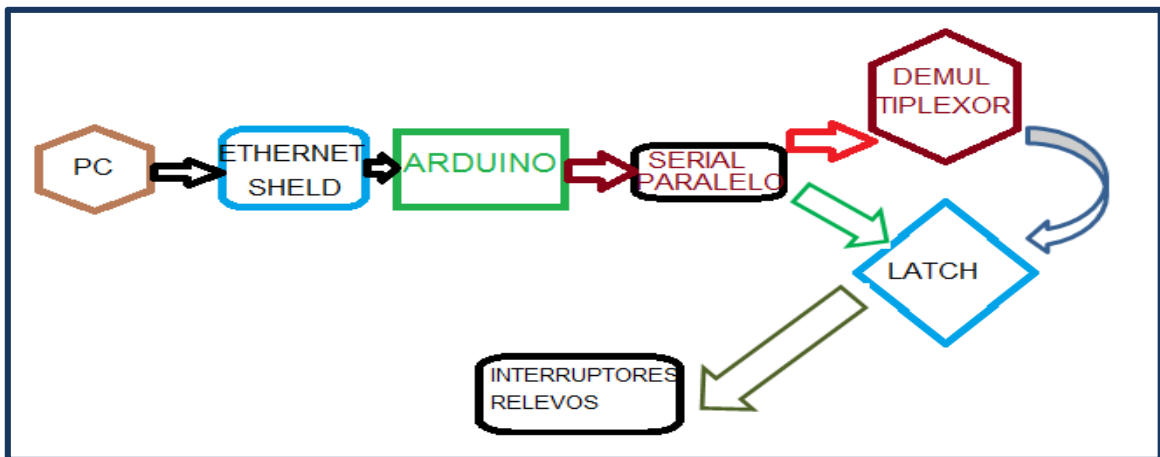


Diagrama 1. Esquemático de la propuesta.

Diseños de circuitos

Conexión de la Board Arduino Uno [8]

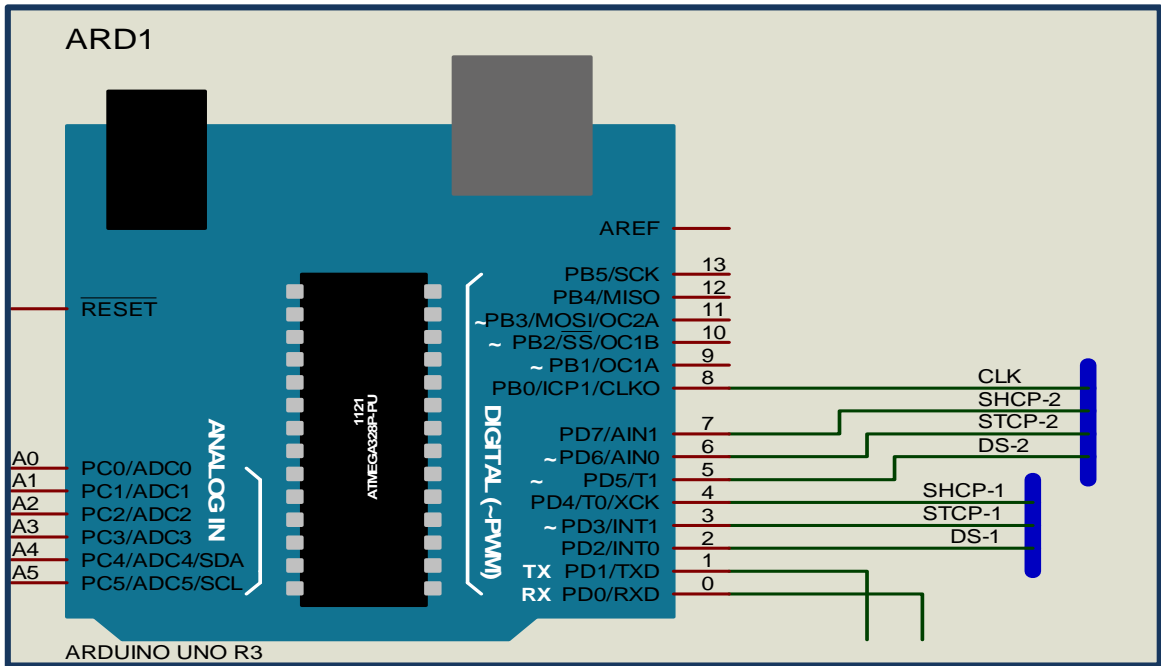


Diagrama 2. Arduino en Proteus

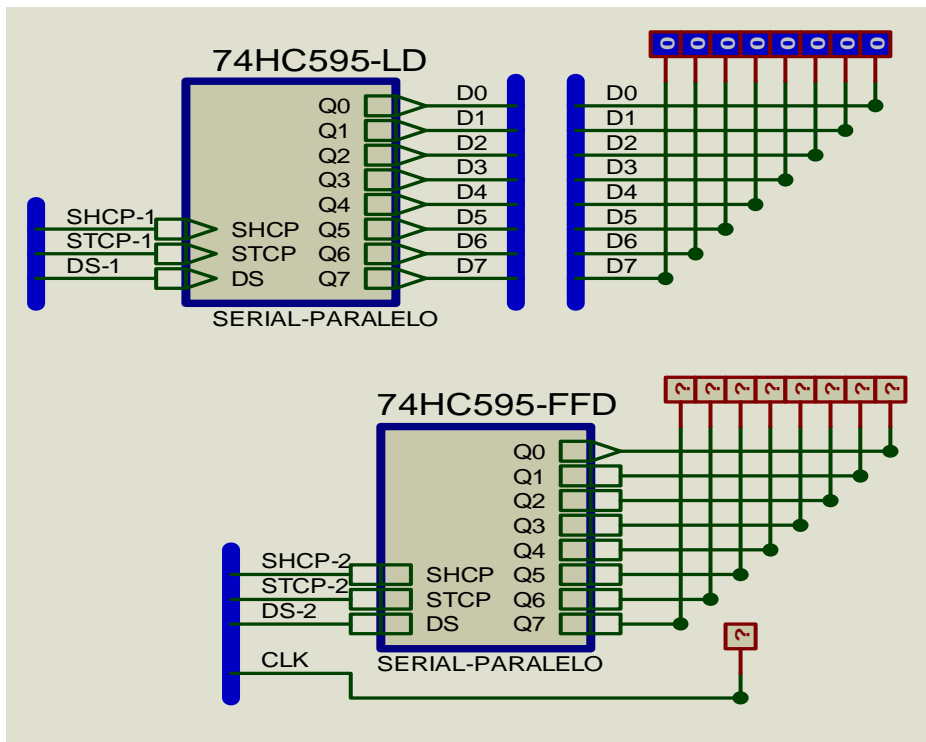


Diagrama 3. Circuito conversor Serial-Paralelo

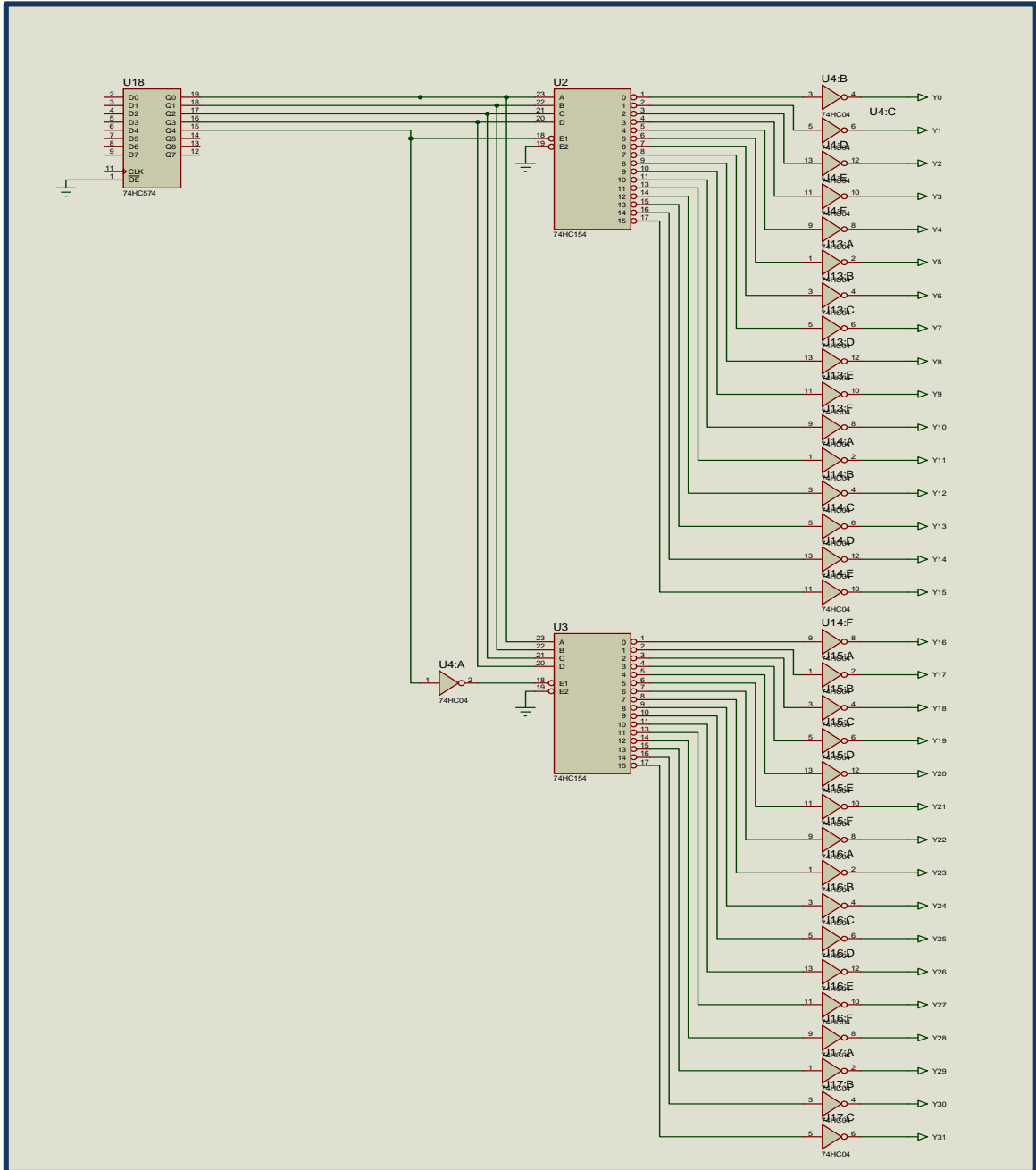


Diagrama 4. Circuito Demultiplexor

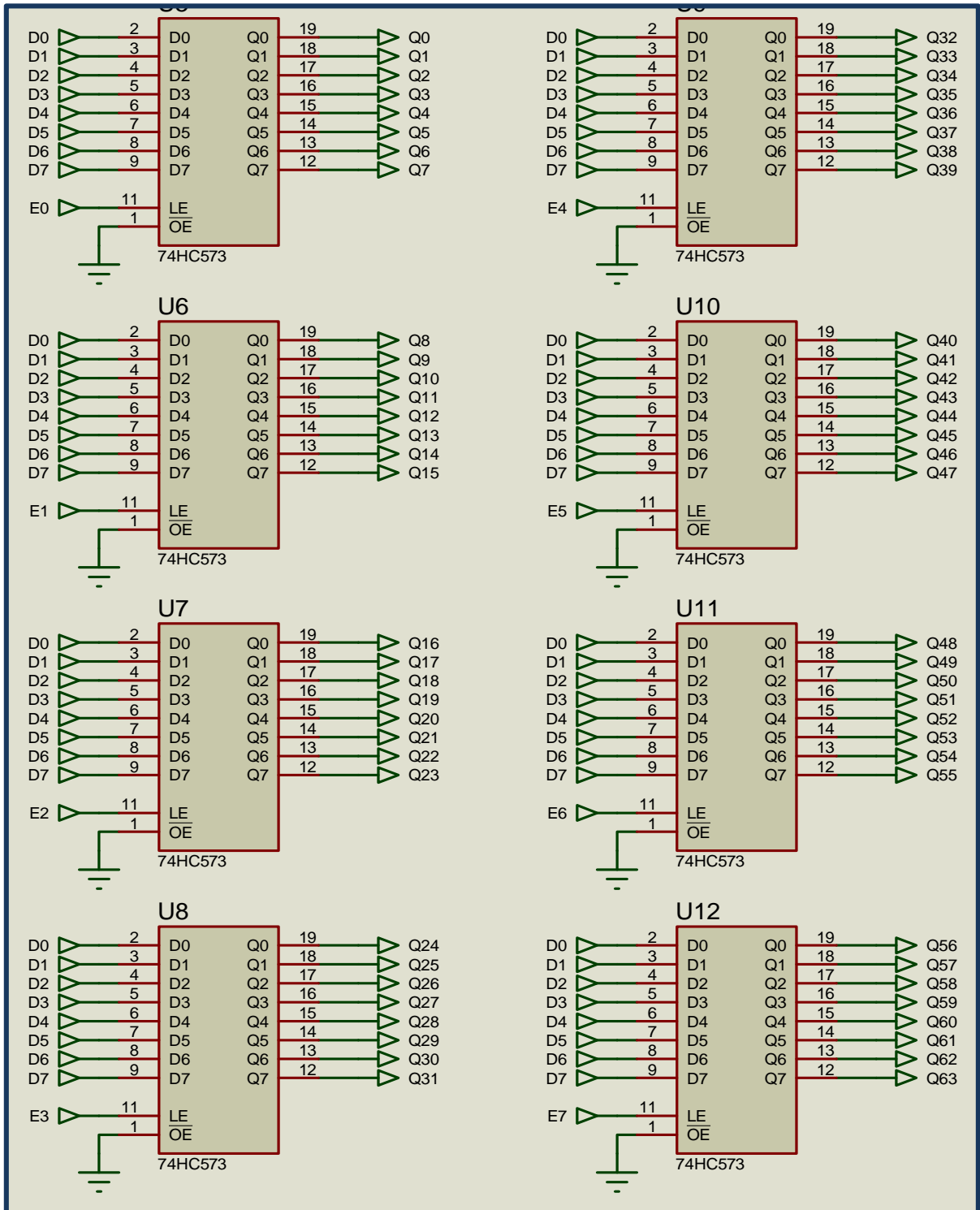


Diagrama 5. Circuito de memoria y registro

Prueba de funcionamiento sobre Planta de Nivel.

Puntos a conectar

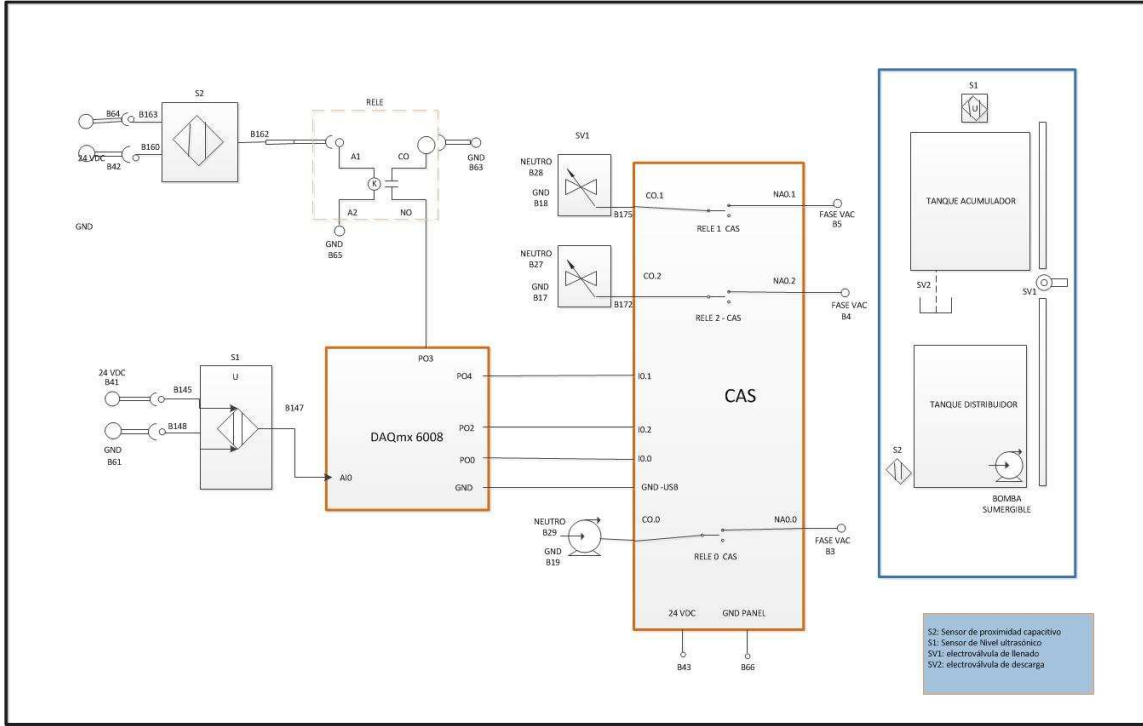


Diagrama 6. Diagrama circuital del control de nivel

Tabla 1. Conexiones para la práctica 1. Planta piloto Nivel.

Botón de Control	Bornera	Servicio	Bornera	Servicio
Q000	B64	+ 24 VDC	B163	+ 24 VDC Sensor capacitivo S2
Q001	B42	GND	B160	GND Sensor capacitivo S2
Q002	B61	+ 24 VDC	B145	+ 24 VDC Sensor Nivel (TL) S1
Q003	B41	GND	B148	GND Sensor Nivel (TL) S1
Q004	B162	Salida Sensor capacitivo S2	A1 RELE1	Bobina del relé
Q005	B65	GND	A2 RELE 1	Bobina del relé
Q006	B63	GND	CO RELE 1	Contacto común
Q007	P0.3	DAQ _{mx}	NO RELE 1	Contacto NO
Q008	B147	Salida Voltaje sensor Nivel (TL) S1	AI.0	DAQ _{mx} Entrada
Q009	IO.0-CAS	A1 RELE 0- CAS	P0.0	DAQ _{mx} Puerto
Q010	IO.1-CAS	A1 RELE 1- CAS	P0.1	DAQ _{mx} Puerto
Q011	IO.2-CAS	A1 RELE 2- CAS	P0.2	DAQ _{mx} Puerto
Q012	B3	Fase 110 VAC/4A	NA0.0	Bobina RELE 0 - CAS
Q013	B178	Actuador (Bomba)	CO.0	RELE 0 - CAS
Q014	B4	Fase 110 VAC/4A	NA0.1	Bobina RELE 1 - CAS
Q010	B175	Electroválvula SV1	CO.1	RELE 1 - CAS
Q011	B5	Fase 110 VAC/4A	NA0.2	RELE 2 - CAS
Q015	B172	Electroválvula SV2	CO.2	RELE 2 - CAS
Q016	B63	+ 24 VDC	CAS	+24 VDC - CAS

Q017	B43	GND	CAS	GND - CAS
Q018	GND	DAQ _{mx}	CAS	USB-GND - CAS
Q019	B62	GND	AI.4	DAQ _{mx}
Q020	B18	Tierra	B176	Electroválvula SV1 - Tierra
Q021	B28	Neutro	B177	Electroválvula SV1 - Neutro
Q022	B17	Tierra	B173	Electroválvula SV2 - Tierra
Q023	B27	Neutro	B174	Electroválvula SV2 - Neutro
Q024	B43	+24 VDC	24 VDC	Alimentación CAS
Q025	B66	GND	GND	Alimentación CAS

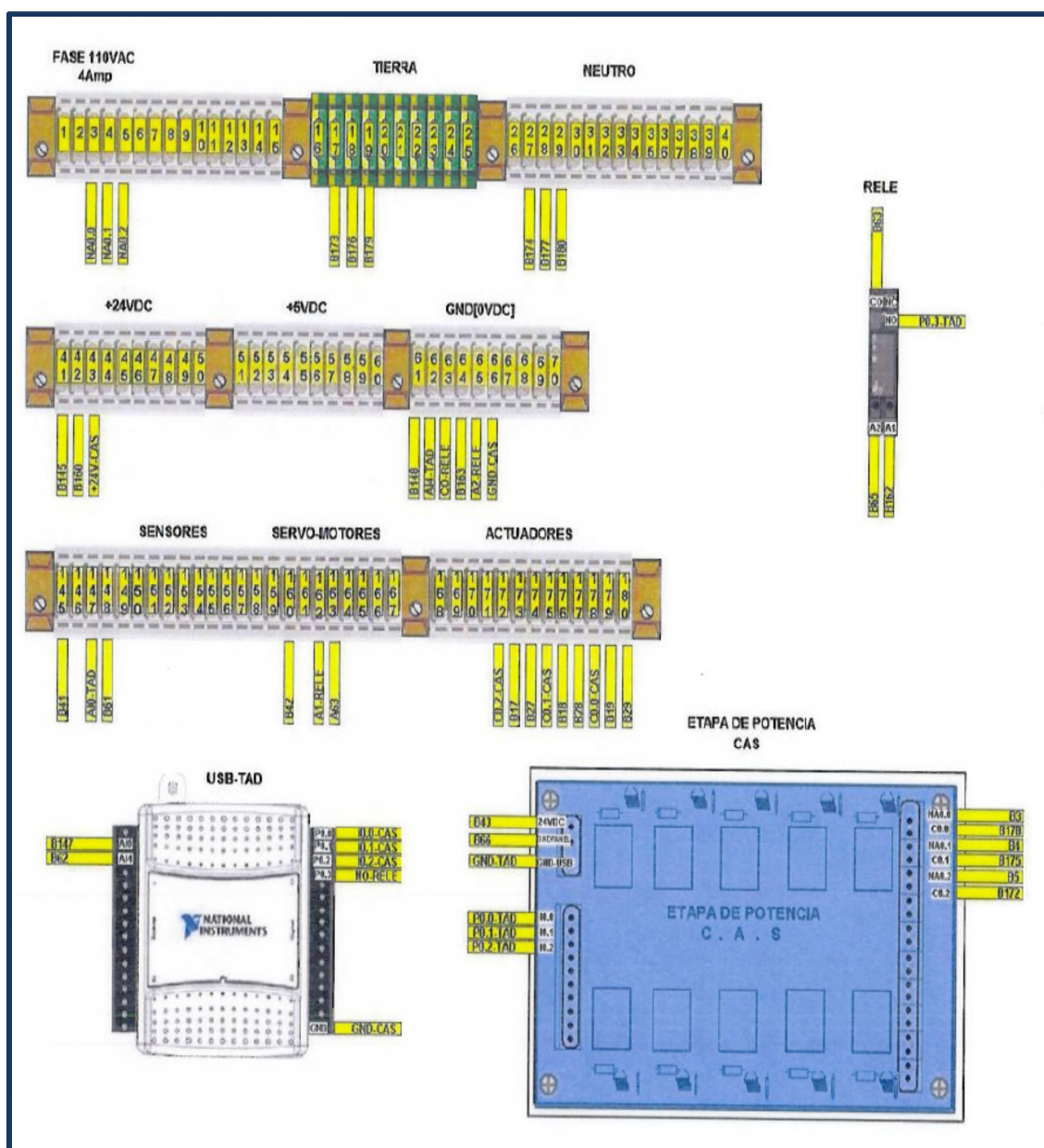


Diagrama 7. Conexión propuesta para el control de la planta de Nivel







6. DESARROLLO DEL PROYECTO

6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO

Después haber realizado cada una de las etapas para llevar acabo el desarrollo del proyecto nos profundizaremos en comprender

6.2 CRONOGRAMA

Tabla 2. Listado de actividades

Actividad	Meses					
	1	2	3	4	5	6
Reconocimiento de la estructura y funcionamiento del módulo M2CI						
Realización del sistema de conmutación para el encendido de las plantas del M2CI						
Diseño de la interfaz de acceso del usuario mediante el software Labview						
Implementación del sistema de conexión de la interfaz gráfica y el sistema de conmutación						
Pruebas y entrega del proyecto						

CONCLUSIONES

Después de haber realizado cada uno de los puntos planteados para este proyecto se logró concluir:

Se logró realizar el reconocimiento del módulo de monitoreo y control donde se identificó que está compuesto de varios instrumentos como los instrumentos indicadores que son los que permiten representar las diferentes medidas e indicaciones sonoras y visuales como lo son el voltímetro, amperímetro, indicador de temperatura, indicadores pilotos, alarma visual y sonora. Existen también los instrumentos actuadores que son los que están permanentemente en cambios, permitiendo diferentes estados en el módulo, estos instrumentos son la bomba electro sumergible, electroválvulas, válvulas manuales, motores DC, SSR's, resistencia eléctrica, Cooler y por ultimo están los sensores y transmisores que son los que permiten tomar los diferentes datos generados en los diferentes dispositivos como son RTD PT100, transmisor de temperatura TH100, Sensor fotoeléctrico sensor inductivo, sensor capacitivo, sensor de posición, sensor de nivel ultrasónico

Para plantear el sistema de conmutación se optó por utilizar dos dispositivos electrónicos, el circuito integrado 74HC4066, el cual se compone de 4 interruptores bilaterales analógicos que se activan con una señal de 5V, este se utilizó para activar dispositivos cuyo rango de operación máximo es 5v, y para controlar los dispositivos de voltajes superiores a 5v hasta los 110v se optó por utilizar un módulo de relés de la Shield de Arduino.

En el diseño del sistema de control se eligió por utilizar los integrados de la familia 74HC ya que tienen tecnología CMOS de alta velocidad combinando la velocidad del TTL, ideal para este proyecto ya que nos permite tener la posibilidad de implementar aplicaciones adicionales si deseamos mejorar el sistema el control y además su precio y facilidad es muy comercial.

Mediante la utilización de la Board de Arduino, nos facilitó crear la interfaz de comunicación entre nuestro sistema de control con la interfaz gráfica realizada en Labview, ya que el Arduino nos permite de manera sencilla conectarlo a un PC, y su aplicativo es sencillo de manipularlo, y mediante un conocimientos básicos de programación podemos controlarlo y además nos ofrecen la utilidad de integrarlo con Labview.

RECOMENDACIONES

1. Establecer las diferentes conexiones para las demás plantas pilotos del M2CI.
2. Continuar la investigación en cuanto a la posibilidad de un cableado directo por parte del estudiante.
3. Desarrollo de la arquitectura para la implementación de la plataforma de laboratorios remotos. A la cual se puedan conectar múltiples prácticas.

BIBLIOGRAFÍA

[1] Velásquez Q. G. (2013). Tecnología e innovación: Aplicaciones para el desarrollo de la ciencia y la sociedad. Electrónica, telesalud y biomédica, Pag 45-47. Universidad Nacional Abierta y a Distancia. Obtenida de URL <http://encuentroingenierias.unad.edu.co/memorias/memorias/assets/downloads/Libro%20Memorias%20CEM%202013.pdf>

[2] Flórez, M.; A. et al (2010) Sistema de monitoreo y control de instrumentación Industrial asistido por computador. Manual de experiencias de laboratorio. Bucaramanga. Colombia

[3] M2CI - Entrenador De Procesos. Red de emprendedores. Consultado el 31 de Julio 2015. Obtenida de URL <http://www.redemprendedoresbavaria.net/pg/photos/view/815728/m2ci-entrenador-de-procesos>

[4] Arduino. Obtenido de URL. <http://arduino.cl/que-es-arduino/>

[5] Shield Arduino. Obtenido de URL.
<https://aprendiendoarduino.wordpress.com/2015/03/23/shields-para-arduino/>

[6] El Relé Shield. Principio De Funcionamiento. Obtenida de URL <http://webcache.googleusercontent.com/search?q=cache:FstWGSUZciUJ:tienda.br icogeek.com/53-shields-arduino+&cd=7&hl=es&ct=clnk&gl=co>

[7] El relé (2010). Obtenido de URL.
<http://platea.pntic.mec.es/~pcastela/tecno/documentos/apuntes/rele.pdf>

[8] Sendoya, L., D.; Bolivar, M., F.; Perez, W., H.; Torres-Silva, P.; (2015) Diseño e implementación de un sistema de conmutación automática aplicado al Módulo de Monitoreo y Control de Instrumentación (M2CI). Informe 1 del proyecto de investigación PIE01-15. UNAD. Neiva.

ANEXOS

Anexo A: circuito de simulación en Proteus

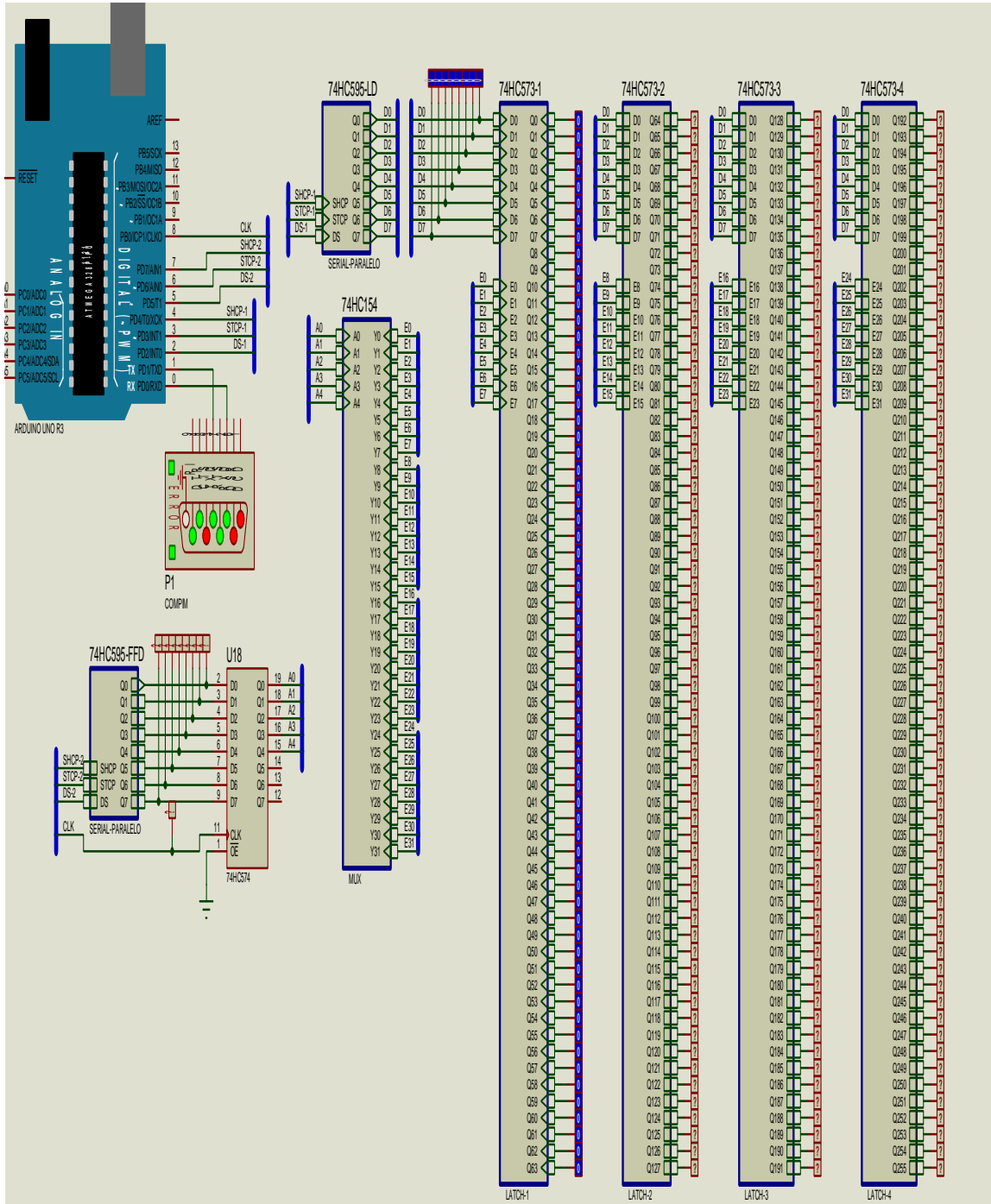
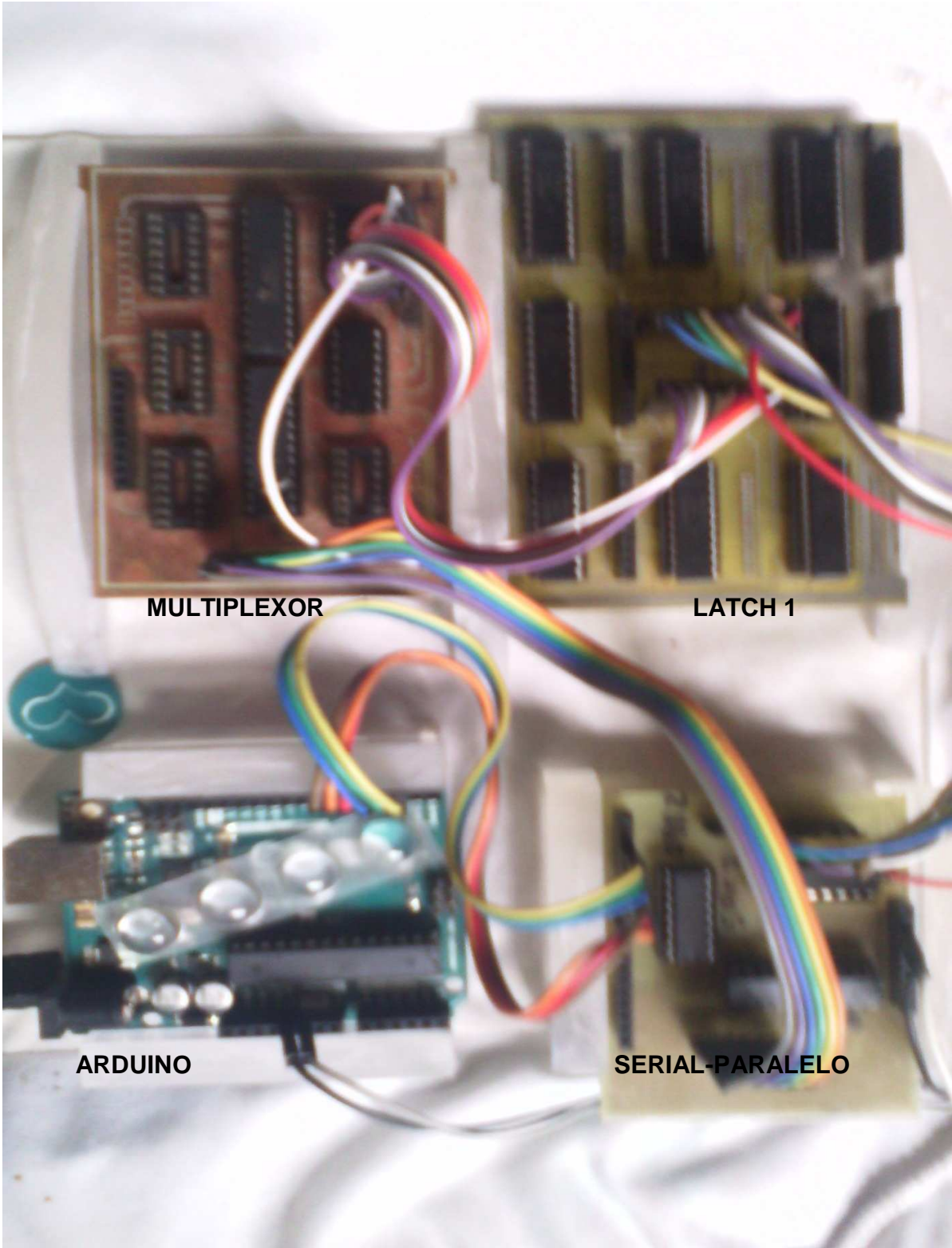


Diagrama 8. Hardware general del proyecto

Anexo B: Montaje físico



Anexo C: Link de descarga

Labview, de la interfaz para Arduino y de los drivers para Arduino Uno.
<https://decibel.ni.com/content/docs/DOC-15971>

Anexo D: Práctica de llenado de tanque y control de nivel

Pasos iniciales:

Prueba y Calibración del sensor de nivel ultrasónico [2] establece las condiciones mínimas para la prueba y calibración del sensor de nivel ultrasónico UB-800 (pag. 82). Finalmente se deberá diligenciar la tabla 3.

Existen dos funciones diferentes de salida en el sensor:

- Valor analógico ascendente, si la distancia del objeto aumenta (rampa ascendente).
- Valor analógico descendente, si la distancias del objeto aumenta (rampa descendente).

El procedimiento:

Teach-in rampa ascendente ($A_2 > A_1$):

- Ubicar el objeto en el límite bajo de evaluación
- Teach in límite bajo: Pulsar A_1 por 1 segundo
- Ubicar el objeto en el límite alto de evaluación
- Teach in límite alto: Pulsar A_2 por 1 segundo

Teach-in rampa descendente ($A_1 > A_2$):

- Ubicar el objeto en el límite bajo de evaluación
- Teach in límite bajo: Pulsar A_2 por 1 segundo
- Ubicar el objeto en el límite alto de evaluación
- Teach in límite alto: Pulsar A_1 por 1 segundo

Tabla 3. *Teach-in* Rampa descendente ($A_1 > A_2$)

Distancia (mm)	V_{out} (Voltaje de salida)
50	
70	
100	
150	
200	

250	
300	
350	
400	
450	
500	
550	
600	
650	
700	
750	
800	

El instrumento virtual (VI)

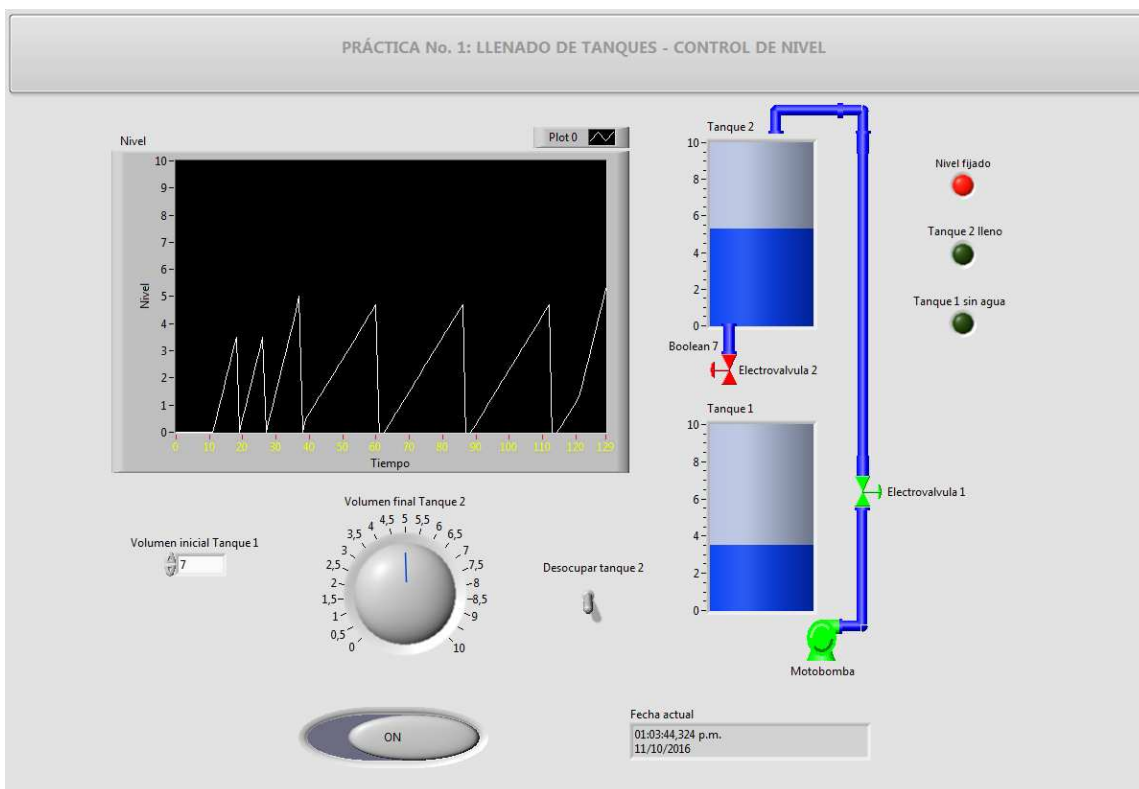


Gráfico 7. Diagrama frontal de VI de la práctica 1. Ejemplo de llenado tanque 2

Procedimiento (Operación simulada):

1. Establecer un nivel para el tanque distribuidor o tanque 1.

2. Encender el VI (Accionar el interruptor a ON)
3. Establecer el nivel de llenado del tanque acumulador o tanque 2, mediante la perilla central del VI
4. Iniciar la prueba, dando clic en ejecutar del Labview.
5. Esperar terminar la prueba
6. Exportar la lectura en Microsoft Excel. Dando clic en la pantalla del Plot, y seleccionando exportar.

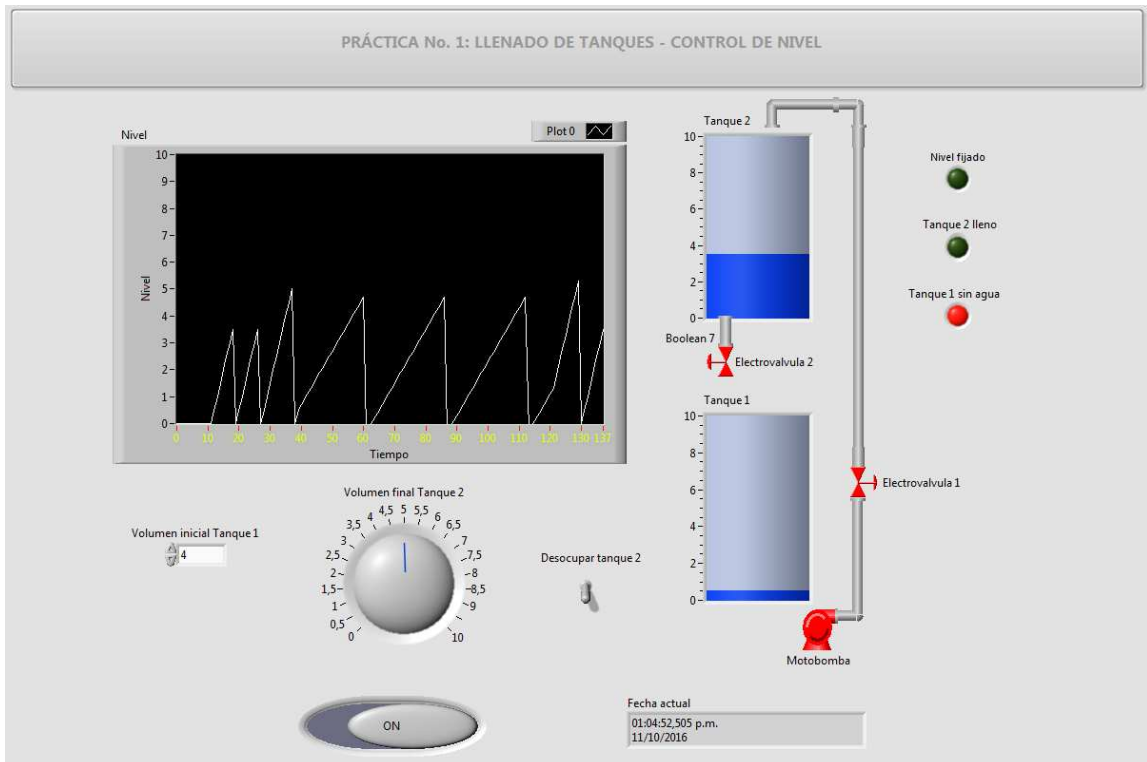


Gráfico 8. Panel frontal práctica 1. Ejemplo tanque 1 vacío.

En la operación de VI simulado, no existen problemas cuando el tanque distribuidor queda desocupado. Se observa el LED de alarma de tanque vacío. Pero en la práctica real con el módulo M2CI, se puede poner en riesgo la bomba sumergible, cuando el tanque queda por debajo del nivel de la misma. Por ello, se hace uso del sensor de proximidad capacitivo. El detecta, el cambio de nivel (Ausencia de agua), activa un relé que a su vez, le informa a la tarjeta de adquisición de datos (TAD - DAQ_{mx} 6008) mediante una señal de GND en el puerto P0.3. Esto activa en el puerto P0.0 una salida en ALTO, que el relé 0 de la tarjeta CAS, apaga la bomba sumergible.

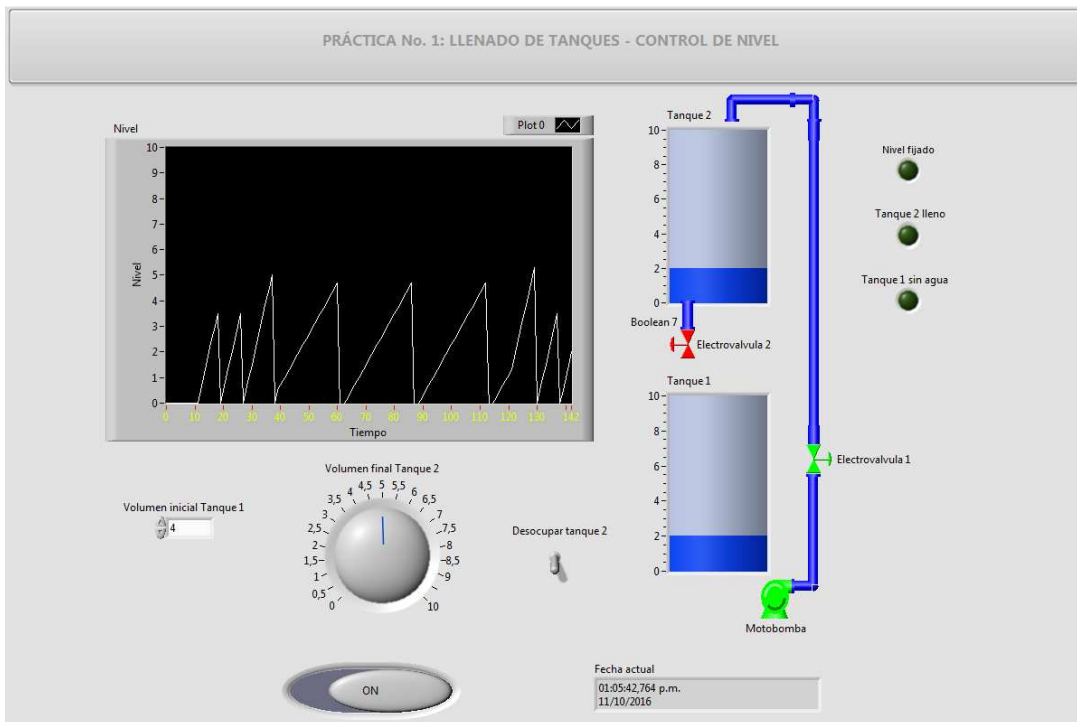


Gráfico 9. Panel frontal práctica 1. En funcionamiento.

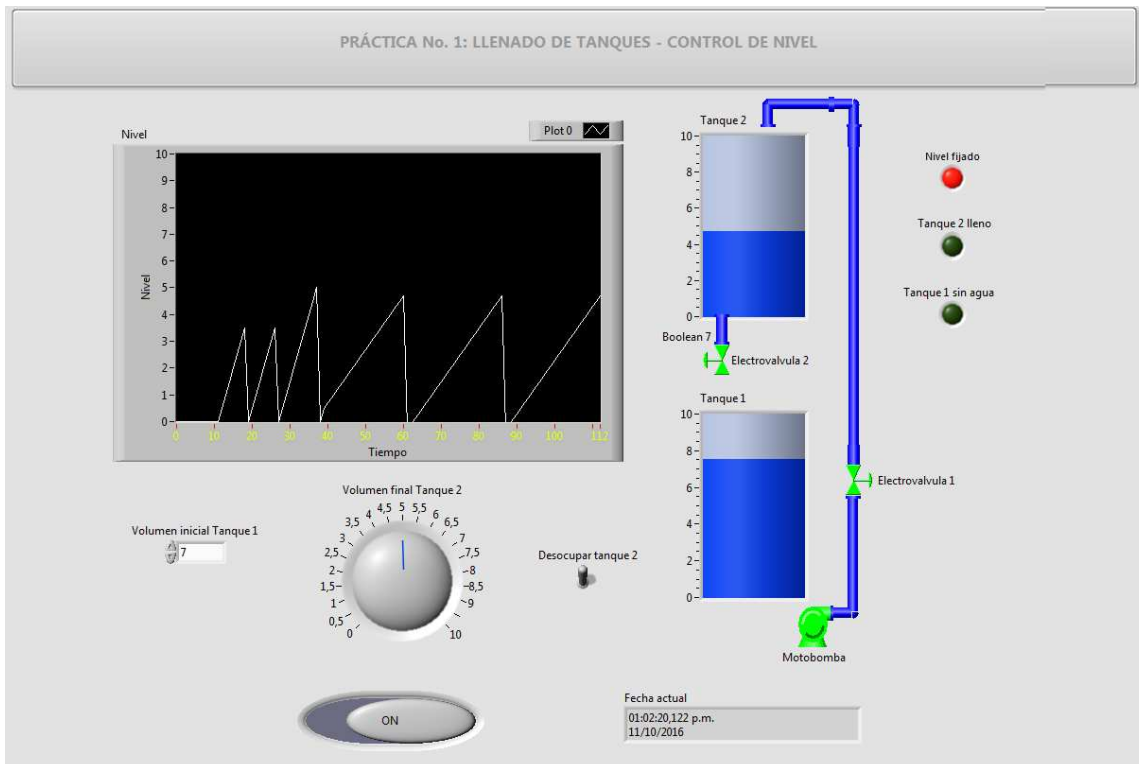


Gráfico 10. Panel frontal práctica 1. Llenado tanque 2 con electroválvula 2 accionada.

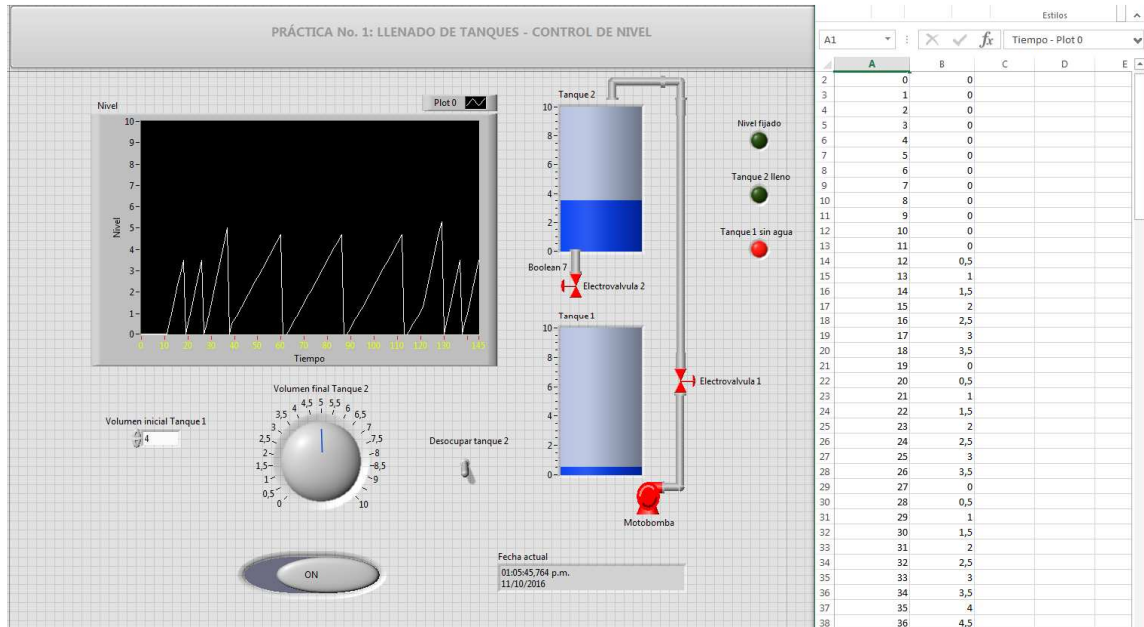


Gráfico 11. Exportación de datos de la práctica 1.

En la realización de la prueba con el VI, se puede accionar la electroválvula 2 (de descarga), con ello, la curva de llenado cambia y se puede realizar los análisis correspondientes.

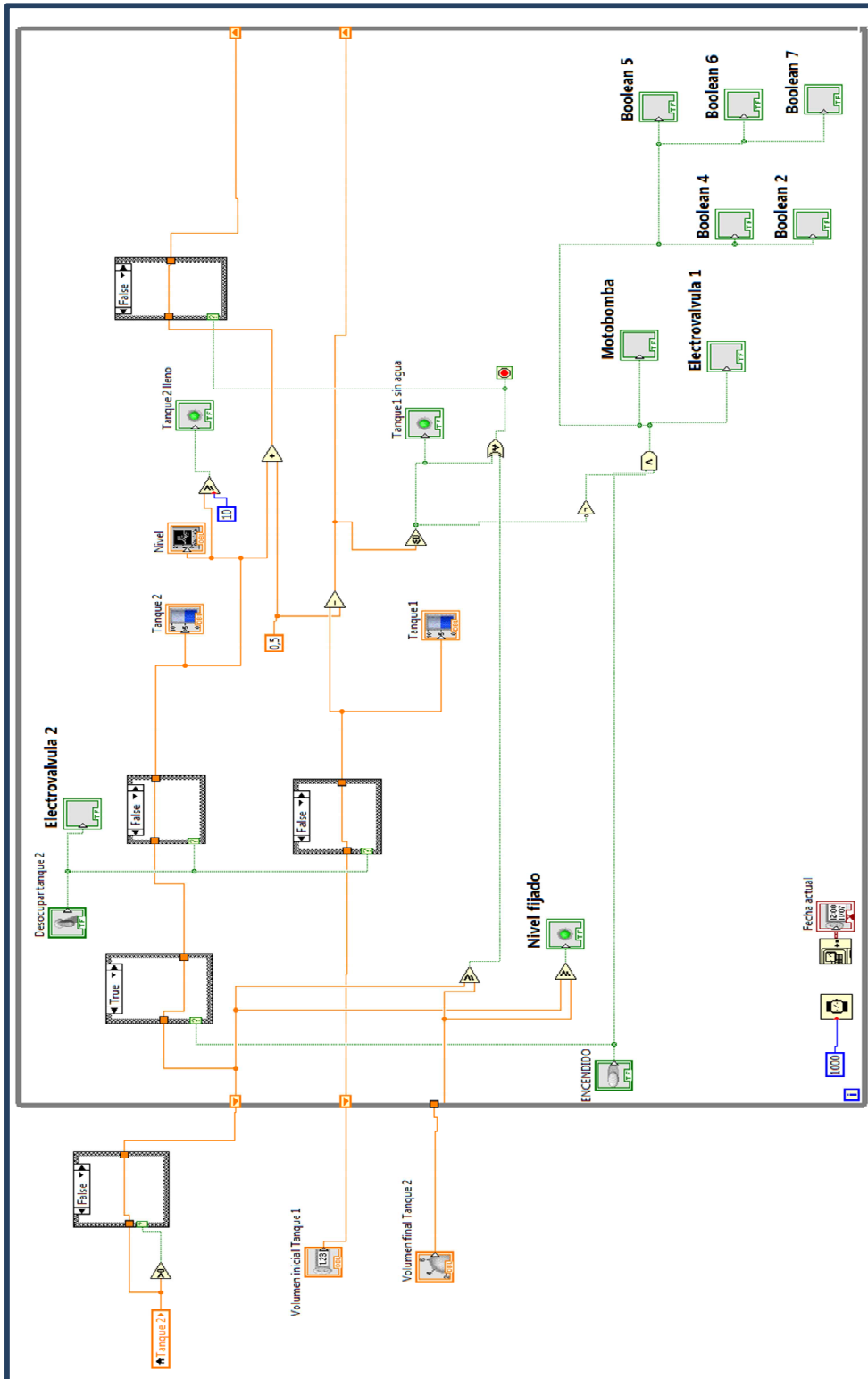


Diagrama 9. Diagrama de bloques del VI de la práctica 1.