ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

"DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL Y MONITOREO DE PROCESOS CON SISTEMAS EMBEBIDOS ARDUINO Y RASPBERRY PI PARA PYMES"

TRABAJO DE TITULACIÓN

TIPO: PROYECTO TECNICO

Trabajo de Titulación presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES

AUTOR: JOHN JAVIER MACIAS OLIVES

TUTOR: ING. EDWIN ALTAMIRANO

Riobamba-Ecuador

2018

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

El Tribunal de Trabajo de Titulación certifica que: El trabajo de investigación: "**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL Y MONITOREO DE PROCESOS CON SISTEMAS EMBEBIDOS ARDUINO Y RASPBERRY PI PARA PYMES**", de responsabilidad del señor Jhon Javier Macias Olives, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	**FIRMA**	**FECHA**
Ing. Washington Luna E. DECANO FIE	----------	----------
Ing. Freddy Chávez V. DIRECTOR EIE CONTROL Y REDES INDUSTRIALES	----------	----------
Ing. Edwin Altamirano DIRECTOR DE TRABAJO DE TITULACIÓN	----------	----------
Ing. Mónica Zabala MIEMBRO DEL TRIBUNAL DE TRABAJO DE TITULACIÓN	----------	----------
DOCUMENTALISTA SISBIB – ESPOCH	----------	----------

DERECHOS DE AUTORÍA

El Trabajo de Titulación que presento, es original y basado en el proceso de investigación y/o adaptación tecnológica establecido en la Facultad de Informática y Electrónica de la Escuela Superior Politécnica de Chimborazo. En tal virtud, los fundamentos teóricos-científicos y los resultados son de exclusiva responsabilidad del autor. El patrimonio intelectual le pertenece a la Escuela Superior Politécnica de Chimborazo.

_____

John Javier Macías Olives

DECLARACIÓN DE AUTENTICIDAD

Yo, John Javier Macías Olives, declaro que el presente trabajo de titulación es de mi autoría y que los resultados de este son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados. Como autor, asumo la responsabilidad legal y académicas de los contenidos de este trabajo de titulación.

_____

John Javier Macías Olives

C.I. 172118702-7

DEDICATORIA

Mi tesis se la dedico con todo amor y cariño a mis amados padres Juanita Olives Quiñones y Júpiter Loor Solorzano, que con su apoyo incondicional nunca dejaron que decayera y siempre creyeron en mis capacidades.

JOHN JAVIER MACIAS OLIVES

AGRADECIMIENTO

A dios, a mis maestros, a mis padres y amigos que estuvieron en todas las etapas de este proceso, de corazón se les agradece por haber estado siempre ahí con esas palabras de aliento que sirvieron para mantenerme firme y no decaer durante este gran esfuerzo que comprendió mi carrera como ingeniero.

.

JOHN JAVIER MACIAS OLIVES

CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

El presente trabajo de titulación describe el diseño e implementación de un sistema de control y monitoreo basado en los sistemas embebidos de la gama “open source”, Arduino y Raspberry para un proceso prototipo de dosificación de sólidos, tapado y sellado en tarros de 125 cc metálicos tomado como modelo piloto para presentar una opción de tecnificación para los procesos de las pequeñas y medianas empresas (PYMES). Se utilizó SolidWorks para el diseño estructural del proceso, determinación de esfuerzos máximos y zonas propensas a ruptura. Dosificación y tapado & sellado complementadas con una acción de transporte basada en una banda transportadora componen toda la línea de proceso. La etapa de dosificación consta de una tolva para almacenamiento de la materia prima en cuya base dispone de una compuerta gobernada por un actuador neumático, que en su estado de abierta o cerrada determinará la dosificación del sólido. En el tapado y sellado dispone de dos actuadores neumáticos que en combinación el uno permite o bloquea el desplazamiento de la tapa al tarro en turno y el otro se encarga por medio de presión y golpe sellar. En el sistema de control se fijó como motor de gestión de recursos del proceso (sensores y actuadores) el sistema embebido Arduino MEGA. Para el sistema de monitoreo se desarrolló una aplicación sobre la plataforma QT Creator con soporte para Python montada sobre una Raspberry Pi3. La integración de los sistemas embebidos Arduino y Raspberry Pi3 por medio de una comunicación serial permitieron obtener un sistema robusto de control con una interfaz de monitoreo en tiempo real y a bajo coste representando una opción válida para la tecnificación de los procesos de las PYMES.

Palabras clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <SISTEMAS DE CONTROL>, <SISTEMA DE MONITOREO>, <ARDUINO>, <RASPBERRY>, <AUTOMATIZACIÓN>.

ABSTRACT

This research focuses on the design and implementation of a control and monitoring system based on the embedded systems of the "open source", Arduino and Raspberry for a prototype process of dosage for solid, capping and sealing in metal jars of 125 cc. This prototype has been taken as pilot model to present a technification option for the processes of small and medium enterprises (SMEs). Solid Works was used for the structural design of the process, determination of maximum efforts and areas prone to rupture. Dosing and Capping & sealing, complemented with a transport action based on a conveyor belt make up the entire process line. The dosage stage consists of a hopper for storing the raw material in whose base it has a gate governed by a pneumatic actuator, which in its open or closed state, will determine the dosage of the solid. Capping and sealing has two pneumatic actuators that in combination, one of them enables or blocks the displacement of the lid to the jar in turn, and the another one is charged by means of pressure and blow seal. In the control system, the Arduino MEGA embedded system was set as the engine of the resources management of the process (sensors and actuators), the embedded system Arduino MEGA. For the monitoring system, an application on the QT Creator platform with support for Python mounted on a Raspberry Pi3 was developed. The integration of the Arduino and Raspberry Pi3 embedded systems by means of a serial communication made possible to obtain a robust control system with a real-time, and low-cost monitoring interface representing a valid option for the technification of the processes of SMEs.

Keywords:

<Technology and engineering skills>, <Control systems>, < Monitoring system>, <Arduino>, <Raspberry>, <Automation>

INTRODUCCIÓN

Actualmente la automatización industrial se ha convertido en uno de los pilares fundamentales para el sector productivo en cualquier país, lo que ha despertado gran interés dentro del área de la ingeniería tanto para académicos como para industriales. La automatización en la industria permite la diversificación de tecnologías con el objetivo de asegurar el control y funcionamiento de un sin número de procesos industriales que puedan ser capaces de reaccionar a situaciones previstas e imprevistas, logrando una reducción de costos, mejora de calidad en los productos y liberando al ser humano de tareas tediosas, peligrosas e insalubres. (Sánchez & Pizarro, 2010)

Según estudios efectuados por la Comisión Europea establecen que las pequeñas y medianas empresas que asimilan la inserción de nuevas tecnologías en sus líneas de producción tienen una tasa de crecimiento de un 15% más alta, y 22% más de ingresos que las pymes que no lo hacen. Considerando que al adoptar nuevas tecnologías en los procesos se debe también capacitar al personal operativo de la empresa para el manejo correcto de las herramientas de las nuevas implementaciones. (Ortiz, 2017)

Las pymes deben ver como una oportunidad el hecho de centrar parte de sus esfuerzos en definir los procesos internos que se pueden automatizar y apostar por soluciones multiplataforma, son claves para que las pymes vivan un cambio exitoso. Teniendo en cuenta que no todo proceso se puede automatizar completamente, que la mayoría suelen ser del back office y que se requiere de una inversión, todo apunta al objetivo de generar espacios que logren integrar productos y servicios en los que se pueda interactuar con el cliente final de manera más natural e intuitiva, acortando brechas de respuestas y demoras innecesarias. En otras palabras: aportando más valor. (Ortiz, 2017)

El presente trabajo está fundamentado en el uso de tecnologías open source disponibles en el mercado ecuatoriano para desarrollar una opción de bajo costo y alta eficiencia para tecnificar los procesos de pequeñas y medianas empresas (PYMES) por medio de la automatización de sus procesos. Se plantea integrar los sistemas embebidos de la gama open source, Arduino y Raspberry para crear un sistema de control robusto y monitoreo en tiempo real puestos a prueba en un modelo prototipo de proceso de dosificación de sólidos y sellado en envases metálicos de 1/8 de litro, se comprobará de esta manera la utilidad de los mencionados sistemas embebidos al momento de aplicarlos en la automatización de procesos.

ANTECEDENTES

Muchos de los procesos en pequeñas industrias se los realiza manual y empíricamente lo que disminuye la calidad del producto y reduce la posibilidad de alcanzar altos estándares y poder competir con grandes empresas cuyas líneas de producción son totalmente tecnificadas.

La tecnificación por medio de la automatización de procesos requiere fundamentalmente un controlador lógico programable (PLC) que centralice el control del proceso o maquinaria en base a la captación y procesamiento de señales de elementos de mando y sensores para gestionar el funcionamiento de actuadores.

El problema para los pequeños productores radica en la alta inversión que representa inicialmente la tecnificación de sus procesos al ser necesaria la adquisición de equipos hardware e incluso en algunos casos software.

El Ministerio de Productividad del Ecuador en Agosto del 2014 indica que el 75% del empleo está generado por las Pequeñas y Medianas Industrias lo que constituye más del 98% de las 500.000 unidades productivas con las que cuenta Ecuador. (Tc Industrial, 2014)

En un estudio realizado por la FLACSO en año 2012 se establece que el nivel de automatización de los procesos productivos en las PYMES es bajo (1,96 en una escala de 5), sin embargo al carecer de sistemas modernos de control las empresas obtienen bajos niveles de calidad en sus productos así como tiempos de producción no aptos para un mercado competitivo y aún más cuando las empresas de este segmento quieren exportar sus productos a otros países. (Tc Industrial, 2014)

El gobierno ecuatoriano en su afán de cambiar la matriz productiva del país se encuentra desempeñando algunos programas para fortalecer la competitividad de las PYMES con el objetivo de que lleguen a economías de escala logrando que los productos ecuatorianos puedan ser vendidos en el extranjero. (Tc Industrial, 2014)

En la ciudad de Cuenca teniendo como grupo de estudio las empresas registradas en la Cámara de Pequeñas Industrias del Azuay y la Cámara de Industrias de Cuenca se desarrolló una investigación empleando encuestas y entrevistas a los responsables de las empresas, de donde se obtuvo información idónea a cerca del nivel de automatización en cada sector siendo que el

nivel de automatización más adecuado a su realidad es el manual con un 36,7%, seguido del automático con 26,5%, semiautomático 25,5%, y 11,2% computarizada. (Sánchez & Pizarro, 2010)

El objetivo número 10 del Plan Nacional del Buen Vivir señala "la transformación de la matriz productiva supone una interacción con la frontera científico – técnica, en la que se producen cambios estructurales que direccionan las formas tradicionales del proceso y la estructura productiva actual, hacia nuevas formas de producir que promueven la diversificación productiva en nuevos sectores, con mayor intensidad en conocimientos y consideraciones de asimetrías tecnológicas". La integración de sistemas embebidos "open source" Arduino y Raspberry para la automatización de un proceso prototipo se la realiza con la finalidad de ofrecer una alternativa más barata y con el mismo grado de funcionalidad de un programador lógico programable para la automatización de procesos. (SENPLADES, 2017).

La integración de los sistemas embebidos, Raspberry Pi y Arduino se la ha empleado en trabajos similares como para el control de un brazo robótico mediante el uso de los lenguajes de programación Java y Phyton, teniendo en cuenta comunicación serial de dos formas diferentes: puerto GPIO y cable USB. El sistema de integración permite la transmisión y la recepción de datos se realiza por medio del software Python, y la interfaz de conexión es realizada con Java sobre una plataforma Android. Se diseñó e implementó un brazo robótico, el cual es controlado por un Arduino, y la interfaz de acceso al sistema se realiza usando la placa Raspberry Pi. Los resultados de las pruebas que se hicieron determinaron la eficiencia del sistema de comunicación y la confiabilidad de la información recibida por la placa Raspberry Pi y transmitida a la placa Arduino para la conexión directa con cada uno de los servomotores. (Suárez Mora, 2015)

JUSTIFICACIÓN

Se pretende realizar el control de un proceso basados en el análisis de señales digitales y analógicas y a su vez el monitoreo de las mismas, facilitando así el seguimiento del proceso. Se pretende usar hardware y software del tipo "open source" con el fin de desarrollar una aplicación que integre dos de los sistemas embebidos más frecuentes en el mercado como lo son Arduino y Raspberry, cada uno de ellos en sus diferentes modelos y versiones. Se utiliza Arduino por sus facilidades de interactuar con el entorno por medio del análisis de señales de sensores que evalúen variables generadas dentro de un proceso para en función de éstas realizar el control de elementos actuadores dentro del proceso. Por otra parte, la Raspberry se emplea por su velocidad para procesamiento de información, operaciones lógicas y aritméticas, creando con la integración de estos sistemas embebidos un sistema robusto.

La integración de los sistemas embebidos Arduino y Raspberry permitirá realizar el control y monitoreo de procesos dotando una herramienta muy similar a lo que sería la implementación de un sistema SCADA (Supervisory Control and Data Adquisition), presentando este prototipo como alternativa para la tecnificación de control y monitoreo de los procesos en las PYMES con una inversión no muy significativa capaz de generar a futuro mayor rentabilidad, orientándose así al cumplimiento del mencionado objetivo 3 del Plan Nacional del Buen Vivir que habla de mejorar la calidad de vida de la población.

OBJETIVOS

General

- Diseñar e implementar un prototipo de control y monitoreo de procesos con los sistemas embebidos “Open Source” Arduino y Raspberry Pi

Especifico

- Realizar el estudio del arte para identificar la gama de hardware eléctrico, electrónico y software bajo el criterio “open source”.

- Analizar las características de Arduino y Raspberry Pi para el control y monitoreo de procesos.

- Establecer las variables dentro del proceso prototipo para su monitoreo y el control.

- Controlar el funcionamiento de los actuadores en el proceso del prototipo en base al análisis de variables analógicas y digitales para las PYMES.

- Evaluar la funcionalidad del prototipo propuesto.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1 OPEN SOURCE

Es el conjunto de dispositivos y componentes electrónicos de los que consta el ordenador, es decir, es la parte "física" o "mecánica". Proporciona un marco para el desarrollo de soluciones a problemas concretos (Villar, 2010)

Se llama hardware libre, hardware de código abierto, electrónica libre o máquinas libres a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago, o de forma gratuita. La filosofía del software libre es aplicable a la del hardware libre, y por eso forma parte de la cultura libre. Un ejemplo de hardware libre es la arquitectura Ultra Sparc cuyas especificaciones están disponibles bajo una licencia libre. (Díaz, 2015)

1.1.1 "Open Source" – Hardware

La replicación de hardware médico con código gratuito y abierto proporciona ahorros superiores al 90% del costo, lo que hace que el material médico y científico resulte mucho más accesible. (Pearce, 2015)

A la sombra del crecimiento del software libre, ha aparecido en los últimos años el llamado hardware libre. Su objetivo es crear diseños de aparatos informáticos de forma abierta, de manera que todas las personas puedan acceder, como mínimo, a los planos de construcción de los dispositivos. Lejos de ser una novedad, esta corriente enlaza directamente con década de los años 70, cuando los primeros aficionados a los ordenadores construían sus propios equipos en los garajes con piezas compradas a diferentes fabricantes y creaban sus propias implementaciones. (Delgado, 2007)

Es un conjunto de diseño y especificaciones de dispositivos electrónicos que es de público conocimiento, que permite la libertad de uso, estudio, modificación distribución y redistribución de las versiones modificadas.

Se puede clasificar este dispositivo de acuerdo con su naturaleza en el cual tendremos:

- Hardware reconfigurable
- Hardware estático

En la figura 1.1 se muestran gráficamente las aplicaciones que se pueden realizar con open hardware.

Figura 1-1: Open Hardware

Fuente: https://opensource.com/sites/default/files/styles/image-full-size /public/lead-images/openhardwaretools_0.png?itok=NUIvc-R1

1.1.2 Tipos de "Open Source" – Hardware

Hardware reconfigurable

Su estructura puede configurarse mediante software (lógica programable), esto establece que como el hardware reconfigurable depende del software, se puede aplicar fácilmente la licencia de Software libre. (Staff USERS, 2014)

Hardware estático

El Hardware estático libre es más complicado de ser desarrollado por su propia naturaleza. Aunque se puede configurar Software para que actúe de manera distinta o tenga otra finalidad, su funcionalidad sigue siendo la misma, pese a que su diseño sea liberado. (Staff USERS, 2014)

Free Hardware Design

El diseño puede ser copiado, distribuido, modificado y fabricado libremente, sin implicar que el diseño no pueda ser vendido o que la puesta en práctica del diseño esté libre de coste.

Libre Hardware Design

Idéntico al Free Hardware Design, pero matizando con la palabra libre la libertad, no al precio. Se refiere a un diseño que pueda ser copiado, distribuido, modificado, y fabricado libremente. No implica que el diseño no puede también ser vendido, o que cualquier puesta en práctica de hardware del diseño estará libre de coste. Todas las mismas discusiones sobre el significado de la "libertad" entre los partidarios de la Free Software Foundation, y los partidarios de la Licencia BSD que afecta al software, desafortunadamente las trasladan a los diseños del hardware.

Open source hardware

Aquel en el cual la información del diseño se pone a la disposición del público en general, se puede basar en una Free Hardware Design o puede estar restringido (Universidad Politécnica de Cartagena, 2013)

1.1.3 Ejemplos de proyectos de hardware libre:

Si bien Raspberry Pi y Arduino son, quizás, los proyectos de hardware libre más conocidos por el gran público; se puede encontrar un extenso abanico de iniciativas y proyectos que siguen la senda del hardware libre que vale la pena conocer. La realidad es que existen muchísimos proyectos más de hardware libre que están apoyados por potentes comunidades de usuarios e incluso por empresas. (Eisner, 2005)

Open Compute Project

El Open Compute Project, es uno de las iniciativas con un enfoque en el mundo del data center y los servidores. Open Compute Project es una iniciativa que surgió hace 2 años de la mano de Facebook, un proyecto de hardware abierto en el que la compañía impulsa el diseño y fabricación de servidores propios cuyos esquemas comparte y así abrir este sector para que se puedan implementar servidores a medida o extremadamente optimizados. (Hypertextual, 2013)

Uzebox

Este proyecto de hardware libre tenía como objetivo desarrollar una consola de videojuegos totalmente libre y abierta, un dispositivo que se ha distribuido en forma de kit y que permita rescatar los mandos de la "clásica" SuperNES y jugar con ellos. (Hypertextual, 2013)

La consola Uzebox está basada en el microcontrolador AVR de Atmel en una placa de hardware extremadamente simple con 4 KB de memoria RAM, 64 KB de memoria de programa, una velocidad de reloj de 28.61818 Mhz (con overclocking del microcontrolador), sonido en 8-bits mono y puerto MIDI en un sistema que usa un kernel basado en interrupciones con el que se sincronizan a tiempo real la generación del vídeo o la mezcla del audio. (Hypertextual, 2013)

Cubieboard

Cubieboard se está haciendo muy popular y se le conoce como "la rival de Raspberry Pi". Esta computadora de 49 dólares (un precio similar al de Raspberry Pi) ofrece una placa de hardware muy potente en la que podremos instalar un disco duro SATA y donde se encuentra 1 GB de memoria RAM, un procesador ARM A10 de 1 GHz y un almacenamiento ya cargado de 4 GB en el que se ha instalado un Android 4.0.4 para que, directamente, se pueda poner a trabajar.

Al ser ofertado como un hardware algo más potente que Raspberry Pi se puede instalar distribuciones Linux como Ubuntu y plantear aplicaciones que requieran un mayor rendimiento o, incluso, introducirse en el mundo de Linux con un computador de gran potencia y muy bajo coste. (Hypertextual, 2013)

RepRap

En el campo de la impresión 3D, el nombre del proyecto RepRap es bastante familiar puesto que se trata de una impresora 3D de diseño abierto cuyos inicios se remontan al año 2005. Hoy

en día, se puede encontrar en la red un amplio abanico de ejemplos de impresoras 3D y muchas de ellas son dispositivos libres cuyos esquemas podemos conseguir para construirnos nuestra propia impresora, sin embargo, RepRap sigue siendo un proyecto de hardware libre interesante porque tenía entre sus objetivos ser capaz de autoreplicarse. (Hypertextual, 2013).

1.1.4 Software "Open Source"

Es aquél que es descrito mediante un lenguaje descriptivo del hardware (HDL). Se desarrolla de manera similar al software. Los diseños son archivos de texto que contienen el código fuente y son cargados en el módulo electrónico. (Díaz, 2015)

El software libre y de código abierto conocido también como FOSS o FLOSS, siglas de Free/Libre And Open Source Software, en inglés, es el software en el cual los usuarios tienen acceso total al código fuente y disponen de la libertad para estudiarlo, modificarlo y mejorarlo en su diseño. El desarrollo de software de código abierto ha sido una gran parte de la creación de la World Wide Web tal como se la conoce, con Tim Berners – Lee contribuyendo con su desarrollo de código HTML como la plataforma original sobre la cual ahora se construye Internet. (Andrearrs, 2014)

El término "software libre y de código abierto" abarca los conceptos de software libre y software de código abierto, que, si bien comparten modelos de desarrollo similares, tienen diferencias en sus aspectos filosóficos que destaca la Free Software Foundation (FSF). El software libre se enfoca en las libertades filosóficas que les otorga a los usuarios mientras que el software de código abierto se enfoca en las ventajas de su modelo de desarrollo. (Andrearrs, 2014)

El open software, respeta las cuatro libertades que la FSF establece:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
- La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
- La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

El objetivo que persigue al compartir el código es que el software resultante sea de mejor calidad y supere al software propietario, es una visión técnica. Es por eso que para lograr calidad técnica se pretende idealmente se comparta el código, claro que nadie está obligado a hacerlo.

El software gratis no necesariamente tiene que ser libre o de código abierto ni viceversa, el término "open source" se refiere a algo que las personas pueden modificar y compartir porque su diseño es de acceso público. (UNESCO, 2001)

El término se originó en el contexto del desarrollo de software para designar un enfoque específico para la creación de programas informáticos. Hoy, sin embargo, "código abierto" designa un conjunto más amplio de valores, lo que llamamos " la vía de código abierto". Los proyectos, productos o iniciativas de código abierto adoptan y celebran los principios del intercambio abierto, la participación colaborativa, la creación rápida de prototipos, la transparencia, la meritocracia y el desarrollo orientado a la comunidad. (Eisner, 2005)

El "código fuente" es la parte del software que la mayoría de los usuarios de computadoras nunca ven; es el código que los programadores informáticos pueden manipular para cambiar la forma en que funciona un programa, un "programa" o una "aplicación". Los programadores que tienen acceso al código fuente de un programa de computadora pueden mejorar ese programa al agregarle características o arreglar partes que no siempre funcionan correctamente. Algunos ejemplos de algunos productos de software de código abierto populares son Mozilla Firefox, Google Chromium, Android, LibreOffice y el reproductor multimedia VLC. (Dourish, 2005)

John Terpstra, Samba.org; cofundador, Samba-Team Linux y el software de código abierto han alterado para siempre el panorama informático. Las conversaciones importantes ya no giran en torno a la tecnología, sino más bien a los asuntos comerciales y legales. El libro de Rosen es una lectura obligada para cualquiera que use o proporcione soluciones de código abierto. (UNESCO, 2001)

A medida que los sistemas de software se vuelven cada vez más grandes y complejos, aumenta la necesidad de predecir y controlar los efectos de los cambios de software. El Análisis de impacto del cambio de software captura la última información sobre la ciencia y el arte de determinar qué partes del software se afectan entre sí. Proporciona una batería de ideas para hacer mejor el análisis de impacto, presenta un marco para el campo y centra la atención en resultados importantes. (UNESCO, 2001)

Obtendrá un respeto saludable por las fortalezas y limitaciones de la tecnología de análisis de impacto y una base sólida que será valiosa en los años venideros. El libro identifica definiciones y temas clave de análisis de impacto e ilustra los temas importantes para darle una comprensión sólida para abordar los problemas de análisis de impacto. Incluye informes sobre el análisis de dependencia del código fuente del software y el análisis de rastreabilidad del software y muestra cómo los resultados de ambas áreas pueden respaldar de manera más efectiva el análisis de impacto en repositorios de ingeniería de software. También describe por qué las técnicas de representación y determinación de impactos están en el núcleo del análisis de dependencia de fuentes y el análisis de trazabilidad. (UNESCO, 2001)

En el desarrollo de software distribuido, pueden surgir dos tipos de dependencias. La estructura del sistema de software en sí puede crear dependencias entre los elementos del software, mientras que la estructura del proceso de desarrollo puede crear dependencias entre los desarrolladores de software. Cada uno de estos forma y refleja el proceso de desarrollo. Nuestra investigación se refiere a la medida en que, al observar de manera uniforme los artefactos y las actividades, podemos descubrir las estructuras de los proyectos de software y las formas en que los procesos de desarrollo se inscriben en artefactos de software. Mostramos cómo se puede descubrir una variedad de procesos y arreglos organizacionales en repositorios de software, con implicaciones para el trabajo colaborativo en grandes grupos distribuidos, como las comunidades de código abierto. (Rosen, 2004). En la figura 2.1 se muestra las diferentes aplicaciones desarrollada en base a open source.

Figura 2-1: Open Source

Fuente:https://opensource.org/files/OSI-Affiliates-Slide-v1.3_2.png

1.2 SISTEMAS EMBEBIDOS

1.2.1 Definición

En el estudio del arte mediante revisión bibliográfica se determina varios conceptos de un sistema embebido entre ellos se citan los más relevantes como son:
"Las personas usan el término sistema embebido para referirse a cualquier sistema de cómputo escondido en algún producto o dispositivo" (Wolf, 2008)

"Un sistema embebido es cualquier dispositivo que incluye un computador programable, pero en sí mismo no es un computador de propósito general" (Morton, 2000).

"Un sistema embebido es un sistema electrónico que contiene un microprocesador o micro controlador; sin embargo, no pensamos en ellos como un computador" (Simon, 1999).

1.2.2 Estructura y componentes de un Sistema Embebido

Un sistema embebido se debe caracterizar esencialmente por el bajo costo y consumo de potencia. El costo por unidad radica como aspecto importante pues son producidos en miles o millones de unidades. Además, que deben cumplir:

- Poseer su programa específico, usualmente ejecuta un programa específico de forma repetitiva.

- Permitir operar números reales.

- Varios de los sistemas embebidos deben ser reactivos o presentar una respuesta inmediata a cambios en el ambiente, además de ejecutar cálculos en tiempo real.

- Acoplable a interfaces de hardware.

- Implementación eficiente y entorno de ejecución a tener en cuenta en la etapa de diseño.

(Ramos, 2015)

1.2.2.2 Componentes

Un sistema embebido en su estructura general dispone de hardware de computador en conjunto con software embebido como uno de sus componentes más importantes. Se considera un sistema computacional dedicado para aplicaciones o productos. Puede presentarse dentro de un sistema independiente o parte de un sistema mayor, y dado que usualmente su software está embebido en ROM (Read Only Memory) no necesita memoria secundaria como un computador. (Ramos, 2015)

Un sistema embebido tiene tres componentes principales:

- Hardware.

- Un software primario o aplicación principal. Este software o aplicación lleva a cabo una tarea en particular, o en algunas ocasiones una serie de tareas.

- Un sistema operativo que permite supervisar la(s) aplicación(es), además de proveer los mecanismos para la ejecución de procesos. En muchos sistemas embebidos es requerido que el sistema operativo posea características de tiempo real. (Ramos, 2015)

Un sistema embebido básicamente estará formado por un microprocesador y un software que se ejecute sobre este. El software necesitará como requerimiento almacenarse en algún sitio, donde ésta necesidad la pueden cubrir la memoria RAM o ROM que puede incluso estar albergada en la misma estructura del chip del procesador. Todo sistema embebido necesitará en alguna medida una cierta cantidad de memoria. Además de esto normalmente un sistema embebido contará con una serie de puertos que permitan el manejo de entradas y salidas para comunicarse o relacionarse con el medio exterior. (Ramos, 2015) (Czwienczek, 2016)

Los procesadores comúnmente usados cuentan con registros de 8 o 16 bits pues las operaciones realizadas por sistemas embebidos son de relativa sencillez. El programa destinado a gobernar una aplicación determinada reside en la memoria. Sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles se conectan a sus pines de entradas y salidas con el fin de cubrir requerimientos planteados. Estas son las únicas características que tienen en común los sistemas embebidos, todo lo demás será totalmente diferente para cada sistema embebido en particular debido a la inmensa diversidad de aplicaciones disponibles. (Czwienczek, 2016). La figura 3-1 muestra la arquitectura básica mas empleada-PC embebidos.

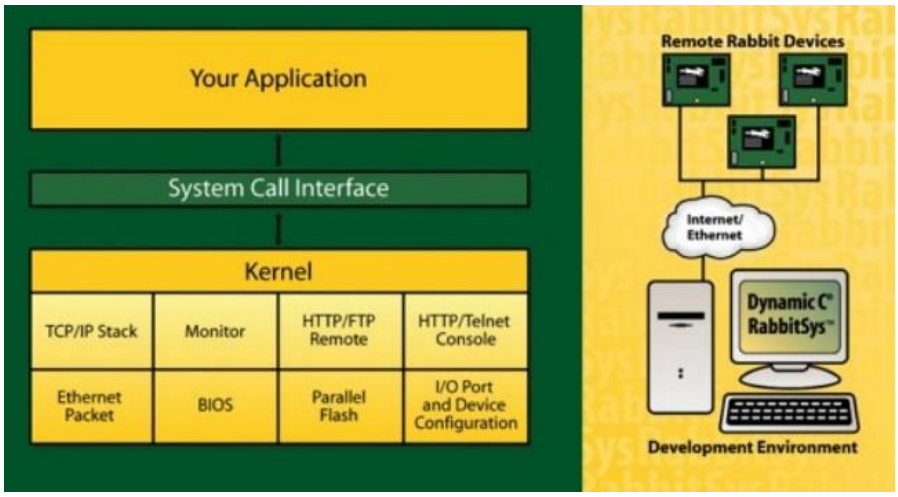


Figura 3-1: Arquitectura básica más empleada - PC embebido

Fuente: (Czwienczek, 2016)

Un PC embebido posee una arquitectura semejante a la de un PC normal con sus elementos básicos como:

Microprocesador

Desarrolla las operaciones de cálculo principales del sistema. Centraliza las acciones de todos los elementos que lo rodean y por medio de la ejecución de código gestiona la realización de una determinada tarea. (Czwienczek, 2016)

Memoria

Alberga tanto código de los programas que el sistema puede ejecutar, así como información adquirida o precargada. Para que el microprocesador no pierda tiempo en tareas que no son meramente de cálculo se plantea esta memoria sea de acceso de lectura y escritura muy rápido. Al ser volátil el sistema requiere de un soporte donde se almacenen los datos incluso sin disponer de alimentación o energía. (Czwienczek, 2016)

Caché Memoria

Almacena datos y código accedido últimamente, de características superior a la memoria principal en cuanto a velocidad. Dado que el sistema realiza micro tareas, muchas veces repetitivas, la caché hace ahorrar tiempo ya que no hará falta ir a memoria principal si el dato o la instrucción ya se encuentra en la caché. Dado su alto precio tiene un tamaño muy inferior (8 – 512 KB) con respecto a la principal (8 – 256 MB). (Czwienczek, 2016)

Disco duro

Puede conseguir capacidades muy elevadas, en él la información está grabada en forma fija. A diferencia de la memoria que es de estado sólido éste suele encontrarse en magnético. Pero su excesivo tamaño a veces lo hace inviable para PCs embebidos, con lo que se requieren soluciones como discos de estado sólido. Existen en el mercado varias soluciones de esta clase con capacidades suficientes para la mayoría de sistemas embebidos desde 2 hasta más de 1 GB. El controlador del disco duro de PCs estándar cumple con el estándar IDE y es un chip más de la placa madre. (Czwienczek, 2016)

Disco flexible

Peculiar encontrarlo en una PC simple más no se lo encontrará en un PC embebido.

BIOS-ROM BIOS (Basic Input & Output System, sistema básico de entrada y salida)

Es código necesario para inicializar el ordenador y para poner en comunicación los distintos elementos de la placa madre. La ROM es un chip donde se encuentra el código BIOS. (Czwienczek, 2016)

CMOS-RAM

Es un chip de memoria de lectura y escritura alimentado con una pila donde se almacena el tipo y ubicación de los dispositivos conectados a la placa madre (disco duro, puertos de entrada y salida, etc.). Además contiene un reloj en permanente funcionamiento que ofrece al sistema la fecha y la hora. (Czwienczek, 2016)

Chip Set Chip

Se encarga de controlar las interrupciones dirigidas al microprocesador, el acceso directo a memoria (DMA) y al bus ISA, además de ofrecer temporizadores, etc. Es frecuente encontrar la CMOS-RAM y el reloj de tiempo real en el interior del Chip Set. (Czwienczek, 2016)

Entradas y salidas al sistema

1.3 Arduino

Arduino es una plataforma de electrónica considerada de fuente libre "open source" que puede ser empleada por diseñadores, aficionados o cualquier interesado en innovar incursionando en la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Arduino se relaciona con variables de su entorno a través de su amplia gama de sensores conectados a sus pines de entrada y puede afectar a aquello que le rodea controlando luces, motores y otros actuadores. (Paredes, 2014).

El microcontrolador en el sistema embebido Arduino se lo programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software, los proyectos elaborados con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, (Arduino, 2013).

El acceso al software es libre se lo descarga de manera gratuita y está disponible para sistemas operativos como Windows, Mac OS X, y Linux (Arduino, 2013). Como ocurre con las distribuciones Linux, Arduino también cuenta con multitud de ediciones, cada una pensada para un público en particular o para una serie de tareas específicas. Existen gran variedad de modelos oficiales, no oficiales y compatibles que es normal que la gente tenga problemas al momento de elegir la correcta dependiendo para el tipo de aplicación que se la requiera (Paredes, 2014).

Figura 4-1: Arduino – Open Source - logo

Fuente: https://www.arduino.cc/en/uploads/Trademark/ArduinoCommunityLogo.png

1.3.1 Modelos de Placas

Entre los modelos de placas comerciales y utilizadas tenemos: placas oficiales y placas no oficiales o compatibles.

Placas Oficiales

Las placas oficiales se consideran a aquellas elaboradas por compañías específicas tales como la compañía italiana Smart Projects y algunas han sido diseñadas por la empresa estadounidense SparkFun Electronics (SFE) o por la también estadounidense Gravitech. Arduino Pro, Pro Mini y LilyPad son las manufacturadas por SFE y Arduino Nano por Gravitech, el resto se fabrican en Italia. Estas placas son las reconocidas oficialmente, incluyen el logo y son las únicas que pueden llevar la marca registrada de Arduino (Paredes, 2014).

Placas no oficiales o compatibles

Frecuentemente estas placas usan un nombre que integra el sufijo "duino" para identificarlas, como por ejemplo Freeduino y Funduino. Estas placas son compatibles con Arduino pero no pueden estar registradas bajo el nombre de Arduino pues son diseñadas y fabricadas por otras compañías ajenas. El desarrollo de estas placas no brinda aporte al desarrollo propio de Arduino, sino que son derivados que han salido para cubrir otras necesidades. (Paredes, 2014).

Existen placas compatibles a nivel del entorno de desarrollo, es decir, solo nivel de software (pudiendo emplear Arduino IDE para programarlas). Otras placas son compatibles a nivel de hardware y eléctricamente para poder emplear los shields y módulos existentes para Arduino sin problema (Paredes, 2014).

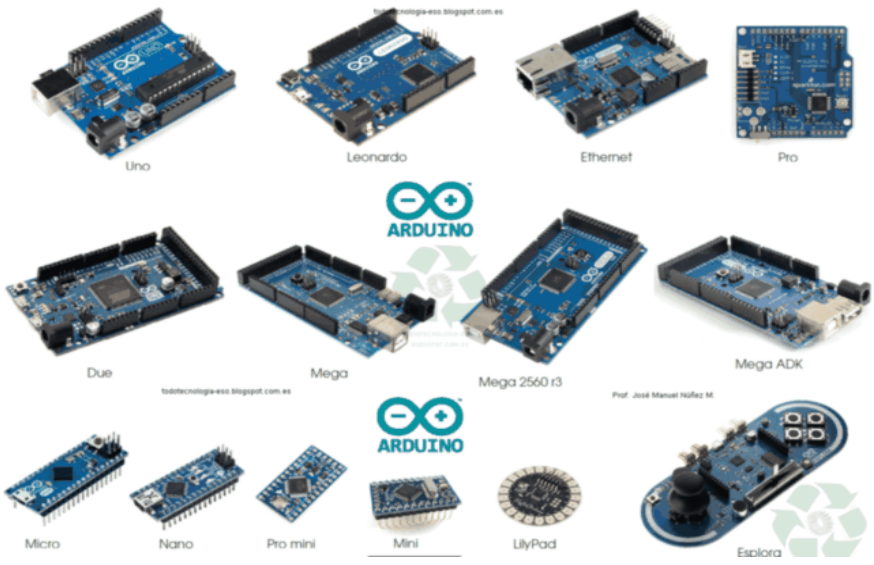


Figura 5-1: **Modelos de placas Arduino**

Fuente:https://1.bp.blogspot.com/-5b5WBm45iFM/VI741_YkpTI/AAAAAAAABcw/eONpHusj6bw/s1600/modelos_Arduino_2014.png

1.3.2 Características Generales

Se destacan entre las principales características del Arduino las siguientes:

- Dependiendo del modelo manejan cierto número de pines digitales y analógicos (las digitales son de tipo normal y de PWM o modulados por ancho de pulso que permiten simular una salida analógica) (Paredes, 2014).

- Para la selección de la placa a utilizar se debe también considerar la extensión del código a generar, pues un programa muy largo, con muchas constantes y variables demandará una cantidad mayor de memoria flash para su almacenamiento, por lo que se debe elegir una placa adecuada (Paredes, 2014).
- Memoria RAM encargada de cargar los datos para su inmediato procesamiento, y afectaría a la velocidad de procesamiento. La RAM va ligada al microcontrolador, puesto que ambos afectan a la agilidad de procesamiento de Arduino (Paredes, 2014).

- Microcontrolador En los Arduinos oficiales se puede diferenciar entre dos tipos fundamentales de microcontroladores, los de 8 y 32 bits basados en ATmega AVR y los SMART basados en ARM de 32 bits y con un rendimiento superior, ambos creados por la compañía Atmel (Paredes, 2014).

- Voltaje En cuanto al voltaje, no importan demasiado a nivel electrónico, excepto en algunos casos, para tener en cuenta la cantidad de tensión que la placa puede manejar para montar el circuito (Paredes, 2014). (3.3V a 5V)

En la figura 6.1 se muestra la estructura de una placa Arduino.

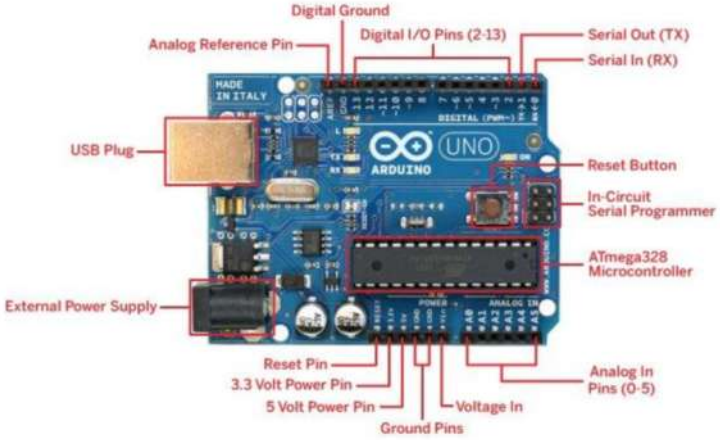


Figura 6-1: Modelos de placas Arduino

Fuente: https://www.robomart.com/image/catalog/RM0058/02.jpg

1.4 Raspberry PI

Raspberry Pi es un ordenador de placa reducida o SBC de bajo coste, cuyo objetivo es el de Estimular la Enseñanza de las ciencias de la computación. El gran éxito de esta minicomputadora, radica en la gran comunidad que se ha creado alrededor de esta; gracias a ello se dispone de mucha documentación y ayuda alrededor de esta placa. Los diferente tipos de placas Raspberry Pi se muestran en la figura7.1

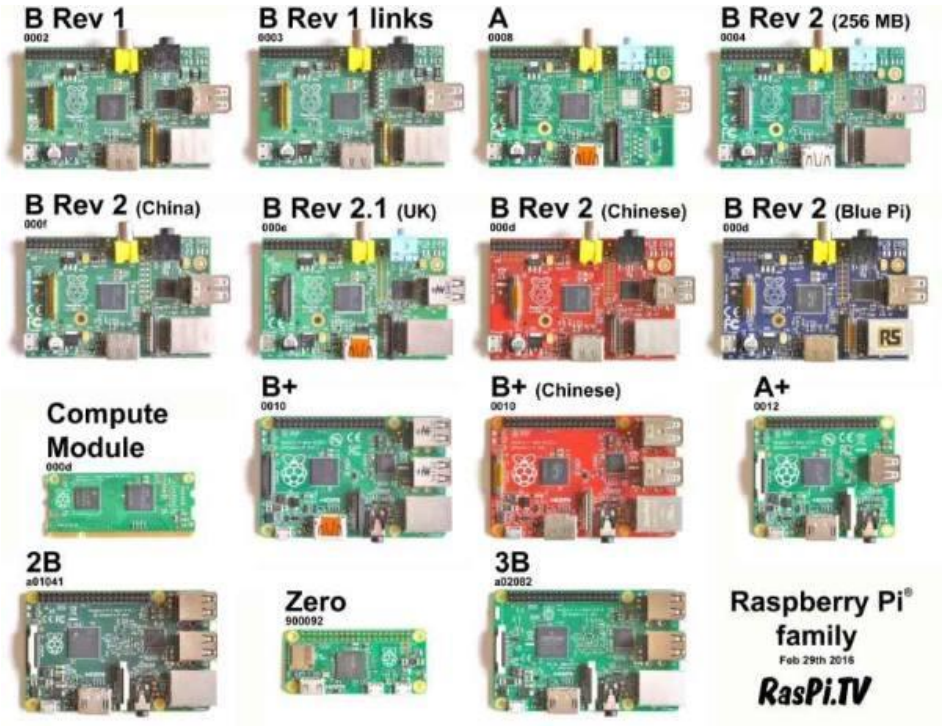


Figura 7-1: Modelos de placas Raspberry Pi.

Fuente: https://www.robomart.com/image/catalog/RM0058/02.jpg

1.4.1 Características Generales.

- Dispone de: Cargador de 1 A para la RaspBerry Pi 1; 1.5 A para la RaspBerry Pi 2 y 2.5 A para la Raspberry Pi 3. Acepta cualquier cargador de Android con Micro-USB.

- Se puede conectar una pantalla HDMI o una pantalla con RCA.

- Se puede conectar Teclado y Ratón a través de sus puertos.

- Incluye un conector para Cable Ethernet o adaptador Wifi USB. En la Raspberry Pi 3 viene incorporado el modulo Wifi.

- Dispone de una ranura para tarjeta SD o MicroSD mayor a 8Gb se recomienda de clase 10 en adelante.

  En la figura 8.1 se muestran los componentes de la placa Raspberry Pi.

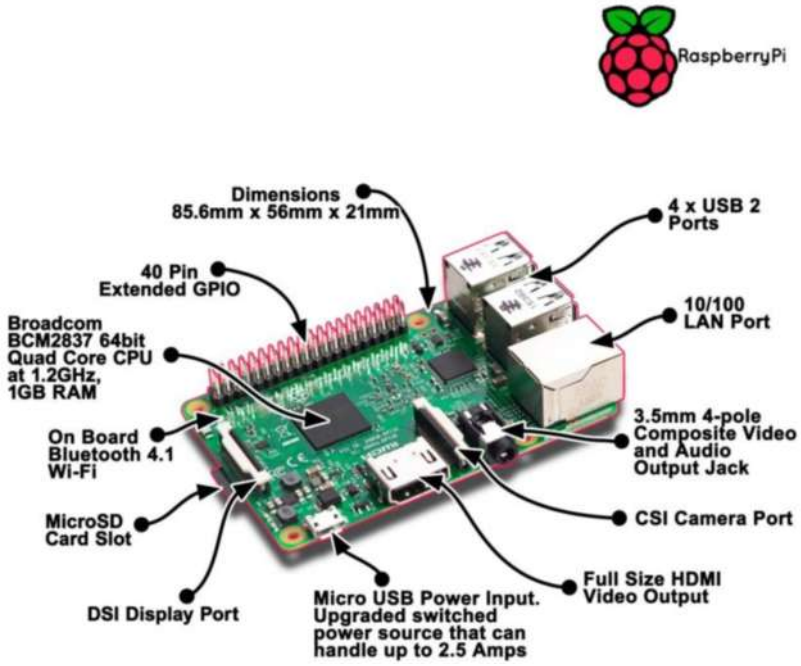


Figura 8-1: Puertos de Conexión Raspberry PI3

Fuente: https://www.planetaelectronico.com/images/productos/raspberry-pi-3-modelo-b-1gb-1-18327.jpeg

1.4.2. Sistema Operativo de la tarjeta Raspberry.

Una parte importante de la Raspberry Pi, es el Sistema Operativo; el mismo que se instalará en la tarjeta SD y con la facilidad de que al cambiar el Sistema Operativo simplemente se cambie de tarjeta. Uno de los sistemas Operativos más utilizados en este tipo de tarjetas es Raspbian que se trata de un Debian (Linux).

Figura 9-1: Sistemas Operativos Raspberry Pi3

Fuente: https://soloelectronicos.files.wordpress.com/2017/08/raspb.png

1.4.3 Instalación de Raspbian

Usando NOOBS (New Out Of the Box Software); es un instalador para los distintos sistemas operativos de Raspberry Pi que se puede usar para realizar la instalación de manera sencilla. Para usar NOOBS, simplemente se descarga desde la web de Raspberry Pi y se copia los archivos en una Tarjeta SD. Después se arranca la Raspberry Pi con dicha tarjeta y seguimos las instrucciones de pantalla.

1.4.4 Aplicaciones de la Raspberry Pi

Ordenador de Oficina

Se puede emplear la Raspberry Pi como un computador de oficina ya que el entorno que presenta Raspbian es completo.

Programación

Raspberry Pi, dispone de herramientas tales como Scratch o SonicPi con el fin de orientar a la iniciación de la programación y relacionar a todo el mundo en el campo de la computación.

Internet de las Cosas

El desarrollo de la tecnología con la inserción del internet de las cosas implica un campo amplio de aplicación de la Raspberry al estar dotada de recursos útiles para comunicarnos con servicios como Bluemix o instalar un Sistema Operativo Windows 10 para realizar aplicaciones con IoT.

Además, la Raspberry puede actuar como un servidor de películas y series, consola de juegos entre muchas de las aplicaciones que se las está dando hoy en día.

1.5 Inserción tecnológica en pequeñas y medianas industrias

Los elementos empleados para el surgimiento, sostenibilidad y desarrollo de cualquier tipo de empresa, se determinan por el eficiente uso de sus recursos y las capacidades competitivas que puede generar con sí mismos en el tiempo; basadas en estrategias que lleven al éxito o al fracaso. (Ortiz, 2013)

Los recursos son esenciales en las empresas; estos forman capacidades tecnológicas que pueden ser conocimientos y aprendizajes organizacionales; por los cuales la empresa acumula tecnología para competir en el entorno y son base para la innovación que se basa en cambiar lo rutinario en el trabajo, con el objeto de mejorar a través de estrategias que lleven a mantenerse en el mercado. (Ortiz, 2013)

La relación que tienen los recursos con la obtención de competitividad, son señaladas por Barney (1997), en su modelo VRIO (Valioso, Raros, Inimitabilidad y Organización), Barney señala cuatro atributos necesarios que deben tenerse para ser competitivo; el primer atributo menciona que los recursos debe ser "valiosos" deben responder a las amenazas del entorno y aprovechar las oportunidades; otro atributo para que el producto sea competitivo es la capacidad del mismo para ser necesario y a su vez "Raros o Escasos" para que sean de difícil obtención para los competidores, se refiere también este atributo a las capacidades y cualidades del talento humano con el que cuenta la organización, ya que con los mejores profesionales con las competencias necesarias para laborar en mi organización puedo lograr mejoras en los procesos; para que estos dos pasados atributitos puedan ser duraderos y sostenibles en el tiempo y a su vez difícil de imitar para los competidores se tiene que contar con el tercer el tercer atributo la "Inimitabilidad"; por último se tiene el atributo de "Organización" el cual hace referencia a la forma como se comporta la empresa; es decir, el clima, la cultura organizacional y los aspectos organizativos que fomente para lograr la maximización de los recursos, capacidades y en general el potencial que posee la empresa.(Ortiz, 2013)

La tecnología ha ido adquiriendo una gran importancia para la teoría de los recursos y las capacidades, dicha importancia radica en que "la posesión de determinadas capacidades tecnológicas se concreta en los conocimientos y habilidades necesarias para diseñar y fabricar productos" (Castillo Saldaña, 2002)

La teoría analiza la complejidad de las organizaciones con la gran cantidad de partes que lo integran, partes que se convierten en factores generadores de competitividad; las organizaciones del sector industrial son integradas por un conjunto de recursos de diversa índole que de ser adecuadamente gestionados pueden aportar beneficios a la organización, ya sea beneficios de disminución de costos, disminución de tiempos, etc., "La teoría de recursos y capacidades se basa en la concepción de la empresa como un conjunto de recursos, preocupándose del estudio de los factores sobre los que se apoyan las ventajas competitivas, para poder justificar la formulación de determinada estrategia de una empresa" (Castillo Saldaña, 2002)

1.6 Automatización

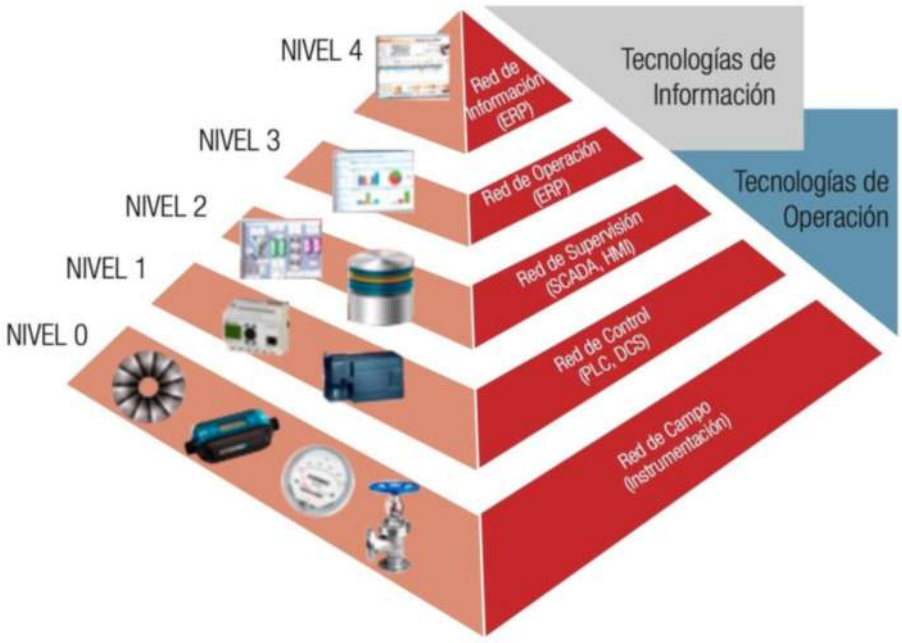


Figura 10-1: Sistemas Operativos Raspberry Pi3

Fuente: https://userscontent2.emaze.com/images/b6b0acdb-44ab-4e1e-99c6-7b5611800435/d1a344a4d9d91de8f58e9a42a15cd7fa.jpg

Se define como automatización a la aplicación de la automática al control de procesos industriales, entendiéndose como automática al conjunto de métodos y procedimientos para la

substitución del operario en tareas físicas y mentales previamente programadas. (Pere Ponsa & Vilanova Arbos, 2005)

Por proceso, se entiende aquella parte del sistema en que, a partir de la entrada de material, energía e información, se genera una transformación sujeta a perturbaciones del entorno, que da (lugar a la salida de material en forma de producto. (Pere Ponsa & Vilanova Arbos, 2005)

Los procesos industriales se conocen como procesos continuos, procesos discretos y procesos batch. Los procesos continuos se caracterizan por la salida del proceso en forma de flujo continuo de material. Los procesos discretos contemplan la salida del proceso en forma de unidades o número finito de piezas, finalmente los procesos batch son aquellos en los que la salida del proceso se lleva a cabo en forma de cantidades o lotes de material. (Pere Ponsa & Vilanova Arbos, 2005)

El control de procesos industriales abarca desde un punto de vista académico, la teoría de control básica de realimentación y acción PID, la instrumentación de control, la aplicación a procesos industriales, las diversas arquitecturas de control, las estructuras de control y la teoría de control avanzada. (Pere Ponsa & Vilanova Arbos, 2005).

1.7 SolidWorks

Es un software o herramienta para dibujo asistido por computador (CAD) empleado para el modelamiento de cuerpos o estructuras en 3D, de fácil aprendizaje que hace posible a los diseñadores mecánicos modelar con rapidez sus ideas, experimentar con las operaciones, cotas y producir modelos y dibujos detallados de fácil fabricación.

Figura 11-1. Entorno SOLIDWORKS
Fuente: http: //sybprogramas.blogspot.com/2016/02/solidworks-premium-2015-64-bits.html

Las principales características que hace de SolidWorks es una herramienta versátil y precisa es su capacidad de ser asociativo, variacional y paramétrico de forma bidireccional con todas sus aplicaciones. Además utiliza el Gestor de diseño que facilita enormemente la modificación rápida de operaciones tridimensionales y de croquis de operación sin tener que rehacer los diseños ya plasmados en el resto de sus documentos asociados. (Gómez, 2014)

CAPÍTULO II

2. METODOLOGÍA

El presente trabajo plantea el demostrar la utilidad de los sistemas embebidos de la gama open source o de fuente libre disponibles en el mercado para su inserción como recursos óptimos para la automatización de procesos, con el fin de proporcionar una alternativa de bajo costo a las pequeñas y medianas empresas para la tecnificación de sus procesos.

Se propone realizar el diseño e implementación de un prototipo de línea de dosificación de sólidos y tapado de recipientes para sobre éste aplicar los sistemas embebidos para su respectiva automatización.

Para el desarrollo del trabajo se plantea seguir una metodología propia desplegada a partir de la experiencia del autor basada en un conjunto de criterios descritos en la siguiente figura:

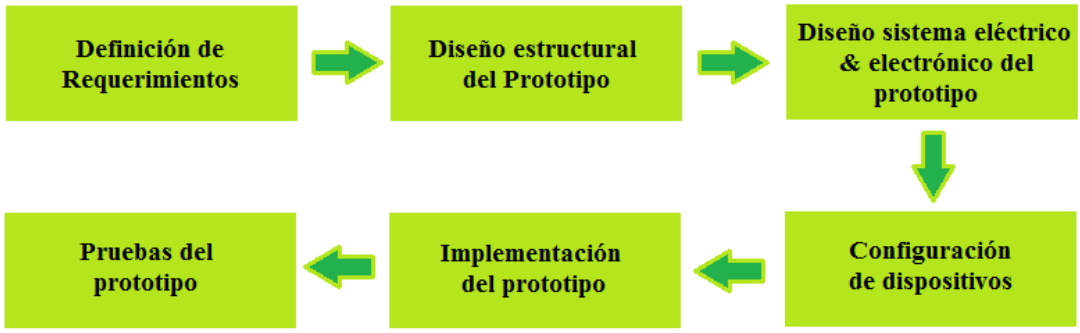


Figura 1-2: Metodología empleada para el desarrollo del trabajo
Fuente: (Macías, 2018)

2.1 Definición de Requerimientos

El prototipo a implementarse debe cumplir ciertos aspectos que determinen la representación de un proceso real en escala atenuada, que mantenga los mismos detalles de un gran proceso, por lo que inicialmente se plantean ciertos requerimientos a considerar al momento de su construcción:

- Se plantea un proceso de dosificación de sólidos en tarros de 125cc de litro del tipo metálico y el tapado de los mismo.

- El prototipo debe tener una estructura sólida capaz de brindar soporte y albergar a los elementos que han de intervenir en el desarrollo del proceso.

- Para el desplazamiento de los tarros a lo largo de las etapas del proceso se requiere un sistema gobernado por una banda transportadora.

- Se requiere de un sistema autónomo capaz de decidir basado en la lectura de sensores los procesos a ejecutarse dentro de la linea prototipo planteada.

- La escencia del presente trabajo está fundamentada en el uso de los sistemas embebidos Arduino y Raspberry Pi por lo que se tiene como requerimiento fundamental emplear estos dos dispositivos como parte principal del sistema de control y monitoreo del proceso.

- El sistema de control contemplará el manejo de actuadores que intervengan en el proceso.

- El sistema de monitoreo se basará en la trazabilidad del producto a través del proceso, adicional a esto se deberá llevar un control de producción, es decir, contabilizar el número de productos terminados.

2.2 Diseño estructural del prototipo

En esta etapa para el diseño del prototipo se empleo una herramienta de dibujo asistido por computador, especificamente el software SolidWorks, por su prestaciones a la hora de crear cuerpos en 3D, seleccionar el tipo de material, realizar simulaciones, efectuar un análisis de cargas, esfuerzos, entre otras de las utilidades que presenta.

Para el diseño del proceso prototipo se efectúa el modelado individual de las piezas que la conformarán empleando SolidWorks.
Se establece una pequeña clasificación de los elementos estructurales de la máquina en la que de acuerdo a la posición y función que desempeñan se dice que pueden ser:

- Elementos de máquina fijos.

- Elementos de máquina móviles.

- Elementos de máquina electromecánicos.

En la figura 2-2 se muestran las piezas que conforman la máquina estáticos o fijos, a continuación se presenta una descripción de cada uno de ellos según lo establecido en el diseño.

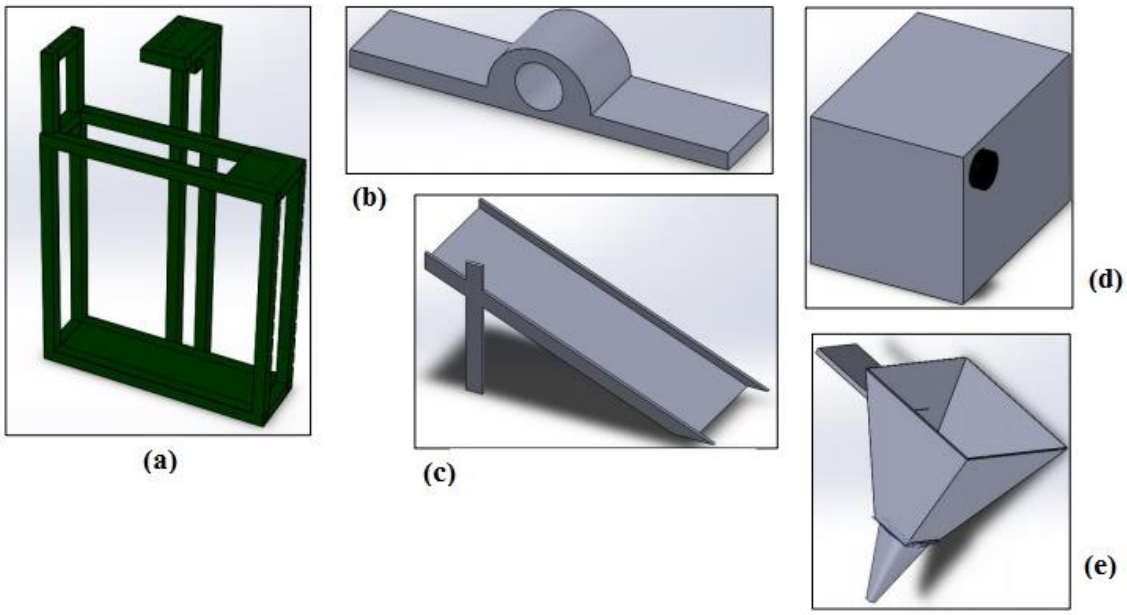


Figura 2-2: Elementos Fijos
Fuente: (Macías, 2018)

a) Estructura metálica: Compuesta de tubo cuadrado de aleación de aluminio 6063-O, se selecciona este material por la disponibilidad en el mercado, bajo costo y utilidad en la construcción de estructuras para prototipos. Representa la base principal donde albergará los componentes y accesorios para el desarrollo del prototipo.

b) Caparazón del rodamiento metálico: Denominada chumacera, pieza de metal con una muesca que incluye un rodamiento en que descansa y gira un eje de una maquinaria.

c) Rampa metálica: Su diseño se enfoca a la generación de una pendiente para la dosificación de tapas y en conjunto con un actuador dosifican la circulación de las mismas.

d) Fuente de poder: Empleada para alimentar al motor eléctrico. Situada en la base de la estructura.

e) Tolva: Actúa como acopio de la materia prima a envasar, en este caso se proyecta a la dosificación de granos, de forma de pirámide o cono invertido, con una abertura en su parte inferior para el paso de su contenido poco a poco a otro lugar o recipiente de boca más estrecha.

En la figura 3-2 se muestran las piezas de la máquina que tendrán movimiento por el impulso de un actuador, a continuación se presenta una descripción de cada una de ellas según lo establecido en el diseño.

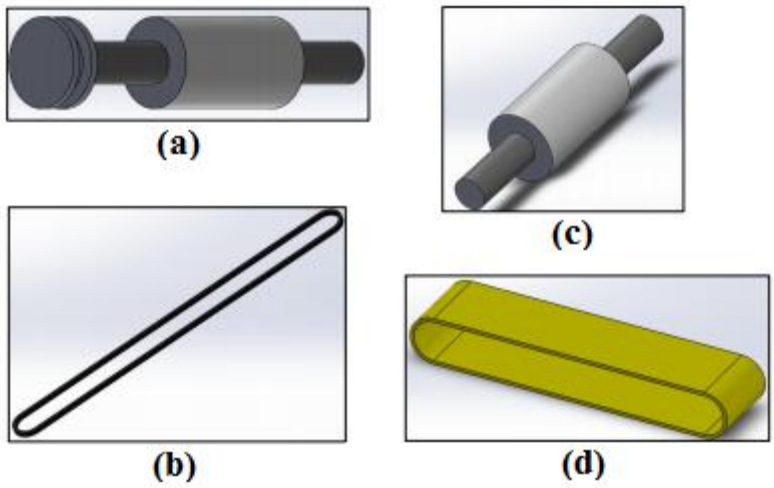


Figura 3-2: Elementos Móviles

Fuente: (Macías, 2018)

a) Rodillo y eje con polea: Es un cilindro o tubo hueco largo, en su orificio ingresa un eje que actúa como centro para el giro. En este elemento se conecta al eje una polea para su movimiento.

b) Banda trapezoidal: Correas trapezoidales o de sección en "V", permiten transmitir pares de fuerzas más elevados mas alta y una velocidad lineal de la correa más alta, que puede alcanzar hasta los 30 m/s.

c) Rodillo y eje: Cilíndro hueco que alberga un eje utilizada para el rodamiento de la banda.

d) Banda transportadora. Elaborada de caucho, actúa como un sistema de transporte basado en una cinta que se mueve continuamente entre dos rodillos. Esta banda es arrastrada por fricción por uno de los dos rodillos, que es accionado por un motor. El otro rodillo gira libremente y tiene como función el de servir de retorno a la banda. (Mott, 2006)

En la figura 4-2 se muestran las piezas de la máquina que por el impulso neumático o eléctrico generarán movimiento, a continuación se presenta una descripción de cada una de ellas según lo establecido en el diseño.

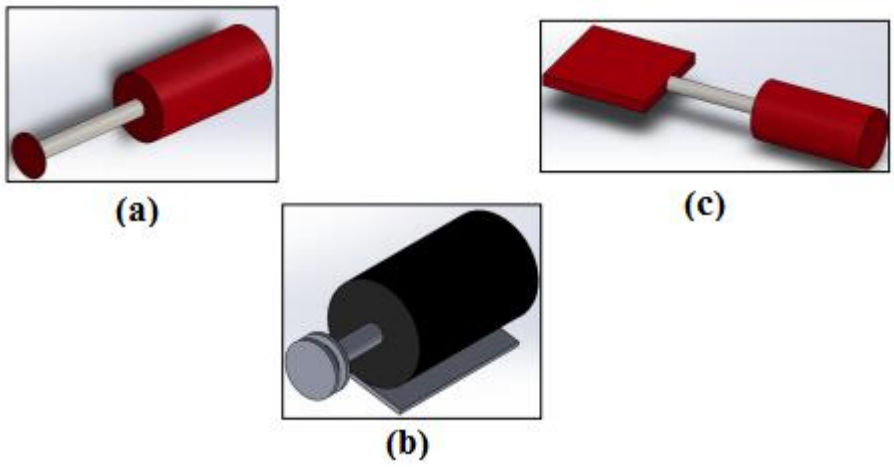


Figura 4-2: Elementos Electro – mecánicos - neumáticos
Fuente: (Macías, 2018)

a) Cilíndro: Es un dispositivo mecánico encargado de producir una fuerza, para dar continuidad a una acción en la etapa de dosificación de tapas y de sellado, para su accionamiento emplea aire comprimido.

b) Motor: Elemento eléctrico que proporciona energía mecánica, encaminado a generar el movimiento para el desplazamiento de la banda transportadora.

c) Compuerta: En el orificio inferior de la tolva se secciona la materia prima a dosificar, se propone un sistema de compuerta para regular el flujo mediante su apertura y cierre.

Una vez trabajados en forma individual los elementos que conforma el proceso del prototipo se procede al ensamble, consiguiendo de esta forma una vista de la maquina completa a implementarse como se lo muestra en la figura 5-2. (ver ANEXO A especificaciones tecnicas de la maquina)

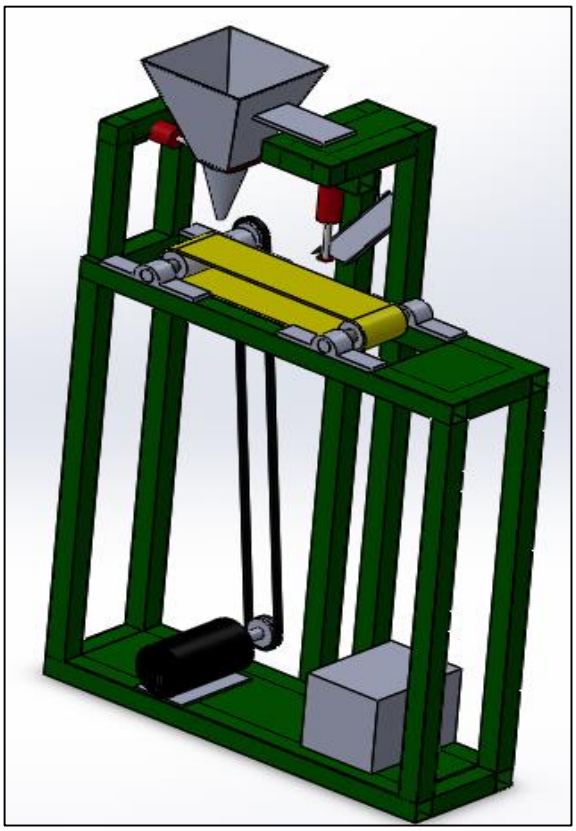


Figura 5-2: Modelado proceso prototipo

Fuente: (Macías, 2018)

2.2.1 Análisis Estructural

Para realizar el análisis estructural de la máquina, empelando SolidWorks donde se simula la estructura para determinar su esfuerzo máximo y las zonas propensas a ruptura por la fuerza generada, como se muestra en la figura 6-2.

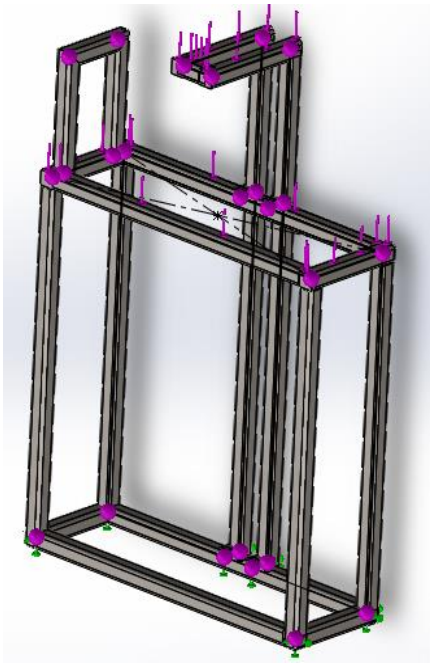


Figura 6-2: Estructura – Definición zonas propensas a ruptura

Fuente: (Macías, 2018)

La figura 7-2 muestra las zonas propensas a ruptura por sobrecarga ejercida en la banda.

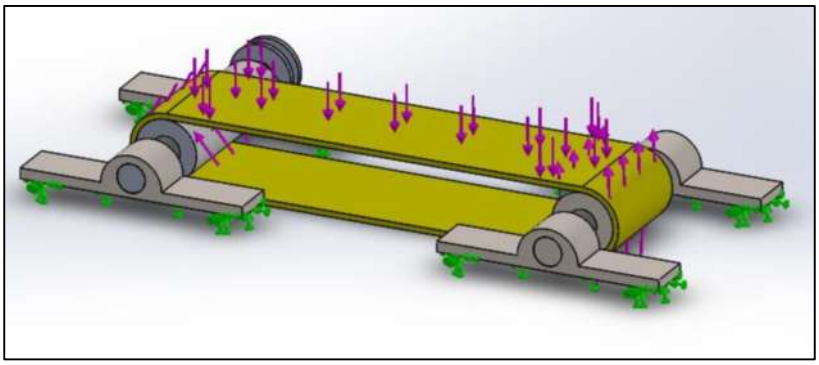


Figura 7-2: Banda – Definición zonas propensas a ruptura

Fuente: (Macías, 2018)

Para determinar las cargas a soportar por la banda se determinan los siguientes parámetros en software SolidWorks.

- El tipo de material es un tubo cuadrado de Aleación de Aluminio 6063-O, una Banda transportadora de Caucho, n Rodillos de nylon 101, 2 Ejes de Acero Aleado y rodamientos de Acero Aleado.

- Definición de los puntos donde se va a someter la carga hasta que llegue a su punto de fatiga.

- Determinación de puntos fijos o de apoyo de la máquina.

A partir del análisis estático en la simulación de la estructura de la maquina en SolidWorks, se muestran los valores aproximados que pueden ser empleados para determinar el peso máximo que soporta la estructura metálica.

2.2.2.1 Fatiga en la estructura

La fatiga ocurre cuando un metal se somete a ciclos de esfuerzo o de deformación repetidos, ello ocasiona que su estructura colapse, y, finalmente se fracture. Se ajusta el diseño y se define en el programa, para reducir los costos de garantía y a provechar a lo máximo la vida útil del producto.

SolidWorks, presenta valores como resultado de las reacciones de fuerzas aplicadas sobre la estructura. La grafica muestra el desplazamiento de puntos propensos a la fatiga y la deformación máxima del material, como se muestra en la figura 8-2.

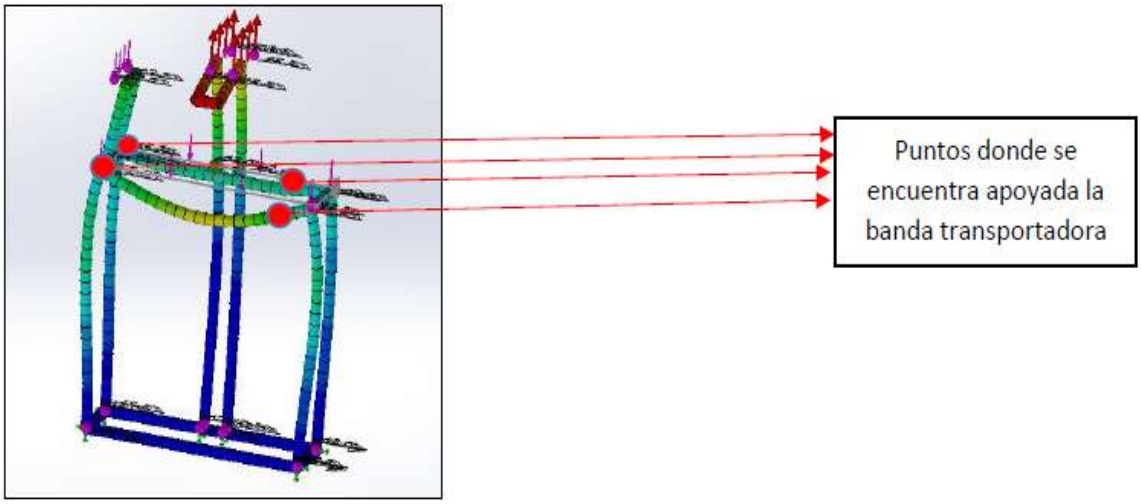


Figura 8-2: Estructura – Puntos de Fatiga (Banda)

Fuente: (Macías, 2018)

La estructura metálica, muestra dos puntos donde puede existir fatiga del material, el primero se encuentra en el soporte de la banda transportadora. El segundo punto se genera en la parte superior, donde soporta el peso de la tolva con el material a dosificar y la presión de la selladora, como se muestra en la figura 9-2.

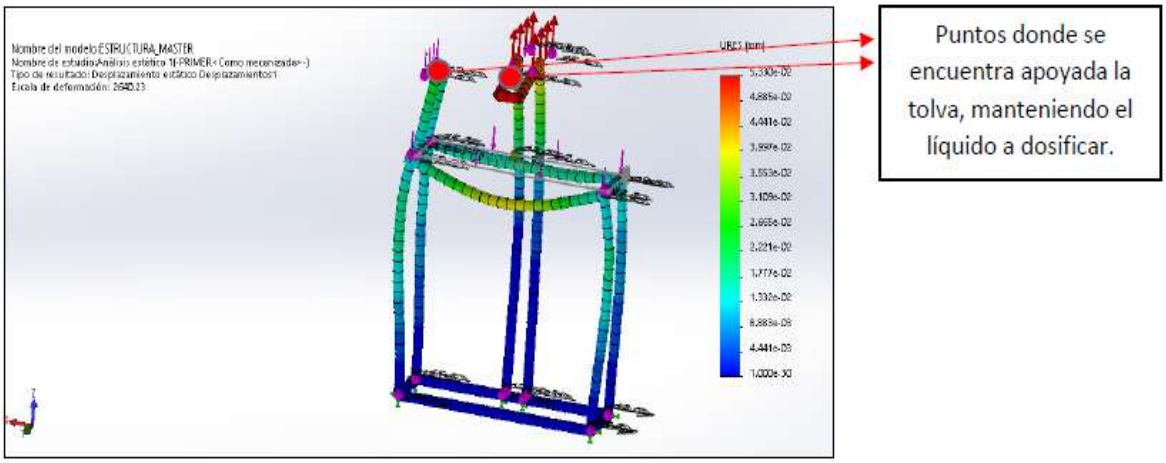


Figura 9-2: Estructura – Puntos de Fatiga (Tolva)

Fuente: (Macías, 2018)

Se observa en la escala de colores de la figura 9-2, que denota de manera ascendente desde el color azul hasta el color rojo, cual es el punto de falla que tendría la estructura metálica de aluminio y se observa cómo se deforma la misma.

El color azul es punto muerto donde no hay fatiga, el color verde indica que va aumentando la falla del material y el color rojo es donde se existe más falla en la estructura. Todo esto depende del peso que se le aplique a la estructura y el tipo de material con el que está fabricada, en este caso el aluminio 6063-O.

2.2.2.2 Análisis de desplazamiento y Fatiga en la banda transportadora.

En la figura 10-2, se indican cuatro puntos donde puede existir fatiga del material. El eje principal del rodillo donde se genera el torque por medio de la polea al motor, es donde más fuerza de torsión ejerce, su color característico es rojo y amarillo, y los otros tres puntos se encuentran en la banda transportadora donde soporta fuerza de tracción y presión, estando en un rango admisible de soportar la fuerza generada por el motor.

La baliza de colores, indica cual es el punto de falla que tendría la banda de distribución en conjunto con los rodillos y ejes. Se observa tambien que se deforma el sistema, el color azul se menciona que es punto muerto, y el color anaranjado es donde existe más falla. El material utilizado en este sistema es una banda de caucho, rodillos de Nylon, y rulimanes de acero.

Hay que tener en cuenta que la banda de la máquina transportadora sufre tres tipos de fuerzas: una de tensión ocasionada por los rodillos; una fuerza aplicada sobre el producto (la selladora) y otro de peso (el producto).

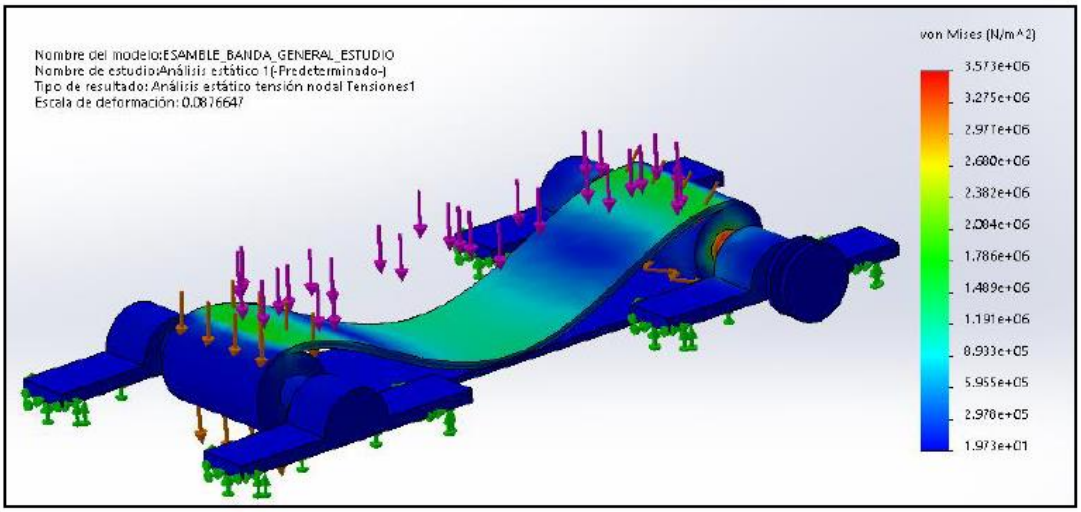


Figura 10-2: Puntos de fatiga de la banda en conjunto de los rodamientos, eje, y rodillos.

Fuente: (Macías, 2018)

De la figura 10-2 se obtuvo resultados de la deformación máxima que soportaría la banda metálica, se obtuvo un esfuerzo máximo de 16460000 $^{N}/_{m^2}$y un esfuerzo mínimo de 19260$^{N}/_{m^2}$.

2.2.2.3 Análisis de Esfuerzos y desplazamiento de la Estructura

La resistencia es la causa que se opone a la acción de una fuerza. También podemos definirlo como la capacidad de un sólido a soportar pesos sin romperse. Este concepto está relacionado con el esfuerzo máximo que un sólido es capaz de soportar. (López, 2013)

En la figura 11-2, se obtiene que la deformación máxima que soportaría la estructura metálica, es de un esfuerzo máximo de 3573000 $^{N}/_{m^2}$ y un esfuerzo mínimo de 19.73 $^{N}/_{m^2}$, con un límite elástico de 50000000.

Cuando hay movimiento, el desplazamiento de una partícula tiene una dirección en el espacio y un módulo. La magnitud que expresa la dirección y la distancia en la línea recta comprendida entre dos puntos del espacio es un segmento lineal llamado vector desplazamiento. (Allen & Mosca, 2005)

En la figura 11-2 se refleja como el cuerpo en cargas máximas se desplaza. En la estructura su desplazamiento máximo con respecto a una fuerza superior a lo establecida, permite observar que se deforma el cuerpo.

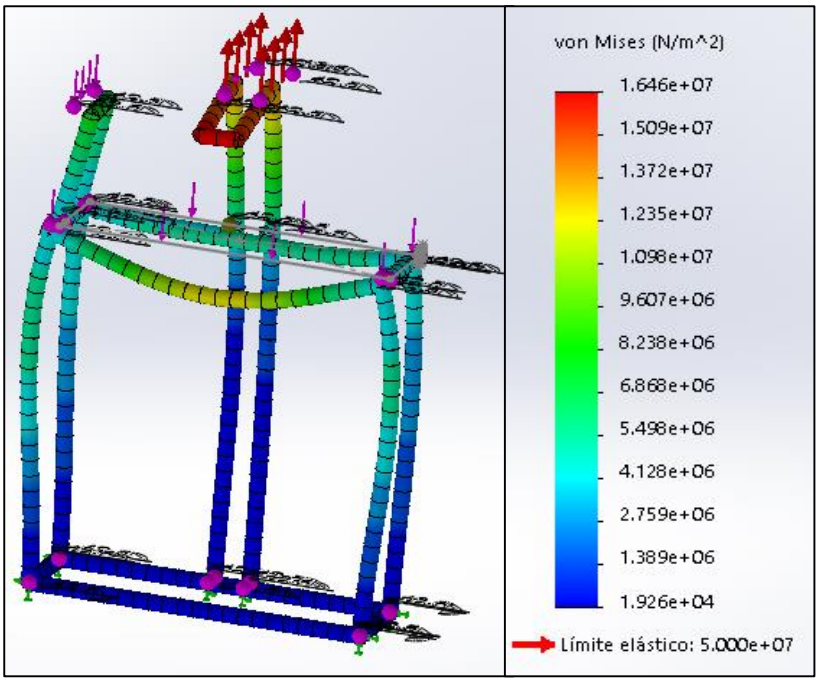


Figura 11-2: Esfuerzo máximo de cargas en la estructura de aleación de aluminio 6063- O.
Fuente: (Macías, 2018)

2.2.2.4 Factor de Seguridad

El factor de seguridad es la razón de la carga de falla, dividida entre la carga permisible. La Carga de Falla se determina por medio de ensayos experimentales del material y el Factor de Seguridad se selecciona con base en la experiencia, de manera que las incertidumbres mencionadas antes sean tomadas en cuenta cuando el miembro se use en condiciones similares de cargar y simetría. (Hibbeler, 2006)

Expresado matemáticamente:

$$FS = \frac{F\ falla}{F\ permisible}$$

Asistido de SolidWorks en la estructura ya definida, se aplica en todo el modelo el factor de seguridad. Se considera un factor de seguridad inferior al valor especificado para localizar las áreas débiles del cuerpo.

Los factores de seguridad altos en una región indican que puede guardar material de dicha región. Muchos códigos requieren un factor de seguridad mínimo entre 1.5 y 3.0. (SolidWorks, 1995)

- Un factor de seguridad inferior a 1.0 en una ubicación significa que el material que se encuentra en esa ubicación ha fallado. (SolidWorks, 1995)

- Un factor de seguridad de 1.0 en una ubicación significa que el material que se encuentra en esa ubicación ha empezado a fallar. (SolidWorks, 1995)

- Un factor de seguridad superior a 1.0 en una ubicación significa que el material que se encuentra en esa ubicación es seguro. (SolidWorks, 1995)

- El material que se encuentra en una ubicación empezará a fallar si aplica nuevas cargas iguales a las actuales multiplicadas por el factor de seguridad resultante, teniendo en cuenta que las tensiones/deformaciones unitarias permanecen en el intervalo lineal. (SolidWorks, 1995)

La figura. 12-2 y 13-2, indican el factor de seguridad 3 durante la simulación en el programa que asegura que se encuentran en un rango permisible.

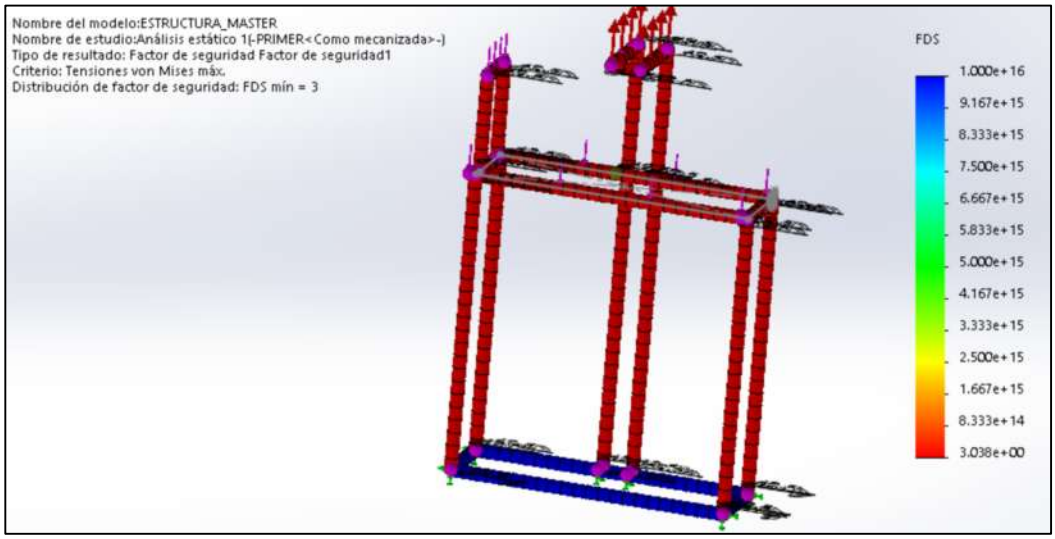


Figura 12-2: Factor de seguridad estructura de aleación de aluminio 6063-O

Fuente: (Macías, 2018)

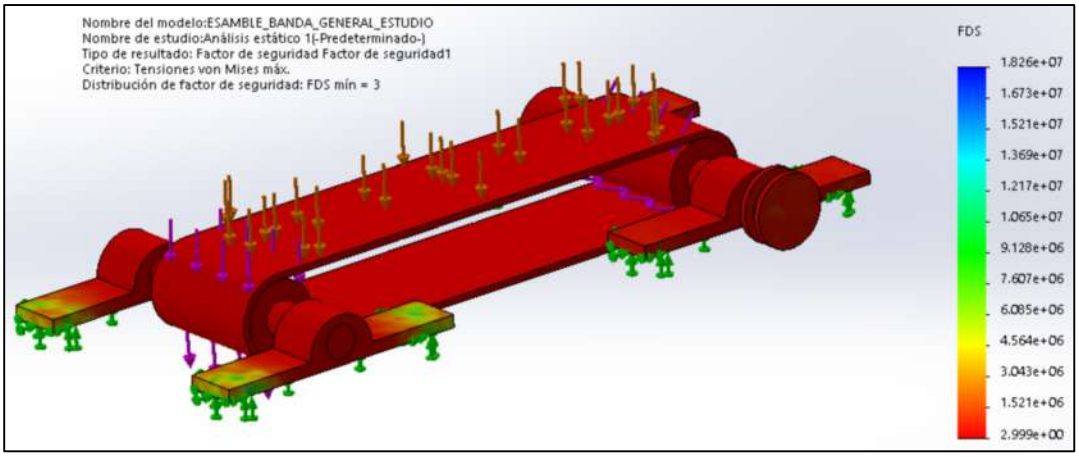


Figura 13-2: Factor de seguridad estructura de aleación de aluminio 6063-O

Fuente: (Macías, 2018)

2.2.2.7 Distribución de pesos

De acuerdo a los datos obtenidos del peso de cada uno de los componentes de la maquina, se presentan en la tabla 1-2.

Tabla 1-2: Pesos de los elementos de máquina

OBJETO	PESO (KG)
ESQUELETO METÁLICO DE ALEACIÓN DE ALUMINIO 6063-O	6.55112
BANDA TRANSPORTADORA	1.52967
MOTOR	2.5
FUENTE	0.5
TOLVA	4
SELLADORA	0.5
DOSIFICADORA	0.5
RESBALADORA	0.01
TOTAL	16.09079

Fuente: (Macías, 2018)

La máquina tendrá un peso de 16,1 kilogramos, en conjunto de todos los materiales propuestos.

2.3 Descripción del funcionamiento del sistema esperado

En base al diseño realizado se genera las siguiente necesidades a cubrir para el desarrollo esperado en el proceso prototipo:

- El sistema deberá iniciar al dotar de alimentación eléctrica a los sistemas de control y monitoreo.

- El proceso detectará la presencia de un tarro (Sensor) al inicio de la banda, al llegar a las etapa de dosificación y sellado.

- Para el control de la dosificación de sólidos en la parte inferior de la tolva se usa una compuerta que esta gobernada en su acción de apertura y cierre por un actuador, por lo que representa una señal a controlarse.

- Para la dosificación de tapas se presenta el diseñado una rampa direccionada e inclinada a la posición del tarro que contendrá en fila el conjunto de tapas, que de acuerdo a la necesidad de surtir interviene un actuador que libera la caída estás.

- En el sellado se hará uso de un actuador que mediante acción de presión y golpe selle los tarros dispuestos.

- El sistema de monitoreo presenta en tiempo real la señalización del proceso que se está ejecutando ese momento.

2.3.1 Definición entradas y salidas del proceso

Definido el modo de funcionamiento del sistema se determinan las variables de control y a controlarse expresadas como entradas y salidas en la tabla 2-2.

Tabla 2-2: Señales del sistema

SEÑAL	**TIPO**
Sensor de tarro al inicio de la banda	ENTRADA
Sensor de tarro etapa dosificación	ENTRADA
Sensor de tarro etapa sellado	ENTRADA
Actuador control de dosificación	SALIDA
Actuador control flujo de tapas	SALIDA
Actuador para el sellado	SALIDA
Actuador para control del desplazamiento	SALIDA

Fuente: (Macías, 2018)

2.4 Componentes hardware del prototipo

2.4.1 Selección Placa Arduino

Se plantea realizar el sistema de control del prototipo mediante la aplicación del sistema embebido Arduino que parte por la revisión de la hoja de especificaciones técnicas de cada modelo y mediante un análisis comparativo de características de la placa versus requerimientos (entradas y salidas) del sistema se determina el que se va a usar.

En este caso, se tomó a consideración como opciones el datasheet del Arduino Uno (ver ANEXO B) y el Arduino MEGA (ver ANEXO C) que son los más comunes en el mercado, y por evaluación se determinó a la segunda opción como la más óptima para emplearla en la aplicación a desarrollarse.

Caracterización Técnica Arduino Mega.

A pesar de que en la evaluación de modelos de placas el Arduino UNO cubría las necesidades del proceso se eligió el Arduino Mega que probablemente es el microcontrolador más capaz de la familia Arduino con la finalidad o aspiración de poder en trabajos futuros seguir añadiendo recursos al proceso. Posee 54 pines digitales que funcionan como entrada/salida; 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un botón de reseteo y una entrada para la alimentación de la placa. (Monk, 2012)

Maneja comunicación serial para comunicarse con el ordenador, por lo que sólo se requiere una conexión directa a un puerto USB. (Monk, 2012)

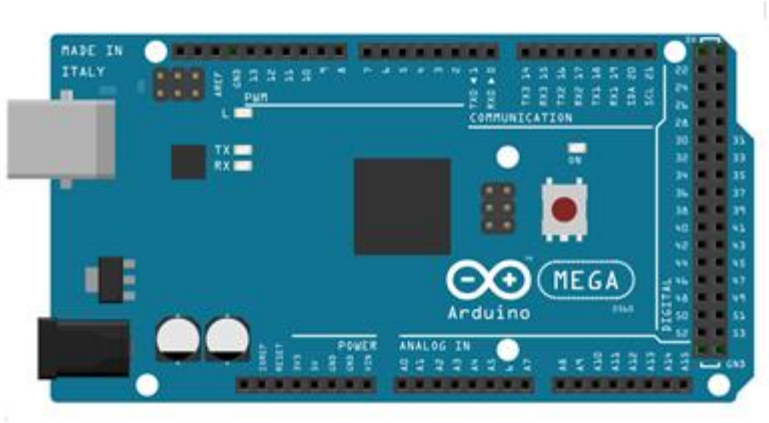


Figura 14-2: Arduino Mega 3D

Fuente: http://blascarr.com/wp-content/uploads/2015/05/hc-05-Mega.png

Características

- Microcontrolador: ATmega2560
- Voltaje Operativo: 5V
- Voltaje de Entrada: 7-12V
- Voltaje de Entrada (límites): 6-20V
- Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)
- Pines análogos de entrada: 16
- Corriente DC por cada Pin Entrada/Salida: 40 mA
- Corriente DC entregada en el Pin 3.3V: 50 mA
- Memoria Flash: 256 KB de los cuales 8KB usados por el bootloader
- Memoria SRAM: 8KB
- Memoria EEPROM: 4KB
- Velocidad de Reloj: 16 MHz

Arduino Mega puede ser alimentado mediante el puerto USB o con una fuente externa de poder. Al trabajar con una fuente externa de poder se requiere una etapa de rectificación y de regulación de dicho voltaje en el rango operativo de la placa. De igual manera se puede alimentar el micro mediante el uso de baterías. Preferiblemente el voltaje debe estar en el rango de los 7V hasta los 12V. (Sabika, 2010)

Arduino Mega posee algunos pines para la alimentación del circuito aparte del adaptador para la alimentación:

- VIN: A través de este pin es posible proporcionar alimentación a la placa.
- 5V: Podemos obtener un voltaje de 5V y una corriente de 40mA desde este pin.
- 3.3V: Podemos obtener un voltaje de 3.3V y una corriente de 50mA desde este pin.
- GND: El ground (0V) de la placa.

2.4.2 Selección Raspberry

Mediante la descripción del sistema embebido Raspberry efectuada en el marco referencial se conoce que existen varios modelos como: Raspberry PI1, Raspberry PI2 y Raspberry PI3 de las cuales las dos primeras están descontinuadas, siendo la Raspberry PI3 un modelo que supera las características de sus antecesoras, de esta manera la selección resulta sencilla.

Caracterización Técnica Raspberry PI3.

La mejor opción es clara, sin embargo cabe mencionar que al evaluar las características de los dos primeros modelos se destaca que para el uso de bluetooth o WiFi requerían módulos externos adicionales, algo que la Raspberry PI3 no requiere pues estos recursos están embebidos también en su placa.

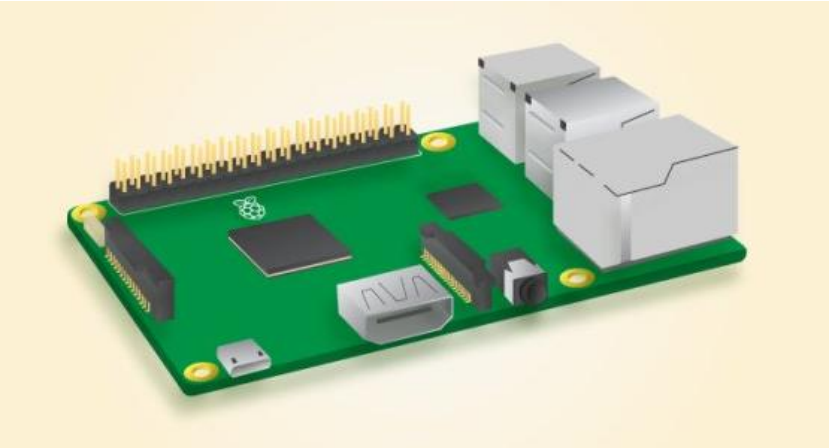


Figura 15-2: Raspberry PI3 3D

Fuente: https://images-na.ssl-images-amazon.com/images/G/30/aplusautomation/vendorimages/32cf0d28-f692- 454d-9429-abe120c0cad8.png._CB281083207__SR970,300_.png

Características

- Marca: Raspberry Pi
- Series Raspberry PI 3 Model B
- Peso del producto: 45,4 g
- Dimensiones del producto: 12,2 x 7,6 x 3,4 cm
- Pilas: 3 9 V (Tipo de pila necesaria)
- Tipo de procesador: Core 2 Quad
- Velocidad del procesador: 1.20 GHz
- Capacidad de la memoria RAM: 1 GB
- Interfaz del disco duro: ATA-4
- Tipo de conectividad: WiFi
- Tipo de conexión inalámbrica: 802.11bgn

- Número de puertos USB 2.0: 4
- Voltaje : 5 voltios DC
- Plataforma de Hardware: Linux
- Sistema operativo: Linux

(Raspberry, 2018)

2.4.3 Selección de actuadores y sensores para el prototipo

Banda transportadora

Para la transmisión de movimiento de la banda transportadora se utiliza un motor eléctrico de bajas revoluciones y de buen torque, para el prototipo se empleo un motor de 12 VCD con caja reductora que otorga bajas revoluciones y buen torque, como se muestra en la figura 16-2.

Figura 16-2: Actuador Eléctrico – Motor DC

Fuente: https://media.rs-online.com/t_large/F3808661-01.jpg

Dosificación de la materia prima

Para la dosificación uso una compuerta, la misma que mediante un acople mecánico se conecta al embolo de un cilindro neumático de doble efecto que según su accionamiento determina el estado abierto o cerrado de la compuerta, como se muestra en la figura 17-2.

Figura 17-2: Actuador Neumático – Cilindro de doble efecto (150mm)

Fuente: http://www.electricosgenerales.com.pe/wp-content/uploads/2015/05/mini-cilindro-doble-efecto-MIC-25X100-S-CA-Electricos-Generales.jpg

Se selecciona un cilindro de doble efecto porque se requiere potencia en el accionamiento tanto en la salida como el retorno debido a que la compuerta estará expuesta a la presión del peso de la materia prima situada en la tolva.

Al ser un actuador neumático requiere un pre actuador que administre el paso del aire comprimido para regular su funcionamiento, en este caso se emplea una electroválvula de 5 vías 2 posiciones de 110VCA, como se muestra en la figura 18-2.

Figura 18-2: Electroválvula 5/2

Fuente: //http2.mlstatic.com/electrovalvula-neumatica-52-rosca-14-bobina-D_NQ_NP_817811-MLA20635913483_032016-F.jpg

La electroválvula mediante un accionamiento electromecánico será la encargada de gestionar el paso del aire comprimido a las cámaras del cilindro de doble efecto. Contiene una solenoide que requiere de una polarización eléctrica para convertirse en un electroimán y provocar la conmutación de la parte mecánica de la válvula habilitando y deshabilitando vías para direccionar el flujo del aire comprimido.

Sellado

Para el proceso de colocación de tapas y sellado final se emplea un actuador similar al anterior, que consiste en un cilindro de doble efecto con su respectiva electroválvula, la variante en estas etapas es que la carrera del cilindro es de 100mm, como se muestra en la figura 19-2.

Figura 19-2: Actuador Neumático – Cilindro de doble efecto (100mm)

Fuente: https://http2.mlstatic.com/electrovalvula-neumatica-52-rosca-14-bobina-D_NQ_NP_817811-MLA20635913483_032016-F.jpg

Detección de tarros

Para la detección de los tarros en las diferentes etapas del proceso se emplea sensores del tipo inductivo, pues estos detectan metal. Las señales de estos sensores serán acondicionadas para ser las entradas del microcontrolador para en base a ellas gestionar las salidas.

La figura 20-2 muestra el sensor inductivo utilizado en el prototipo.

Figura 20-2: Sensor inductivo

Fuente: https://http2.mlstatic.com/sensor-de-proximidad-inductivo-autonics-pnp-na-m12-pr12-4dp-D_NQ_NP_19364-MLV20169507952_092014-F.jpg

2.4.4 Diseño del Sistema Eléctrico y Electrónico

Enfocados en generar la propuesta de un sistema de control y monitoreo basado en los sistemas embebidos Arduino Mega y la Raspberry Pi3 seleccionados para el desarrollo del prototipo. En este punto se realiza el diseño de las configuraciones físicas entre los dispositivos seleccionados como entradas y salidas relacionados a los elementos de control y monitoreo.

Se plantea los diagramas eléctricos dentro de la parte del diseño indicando la configuración eléctrica y electrónica de los circuitos de las entradas y salidas.

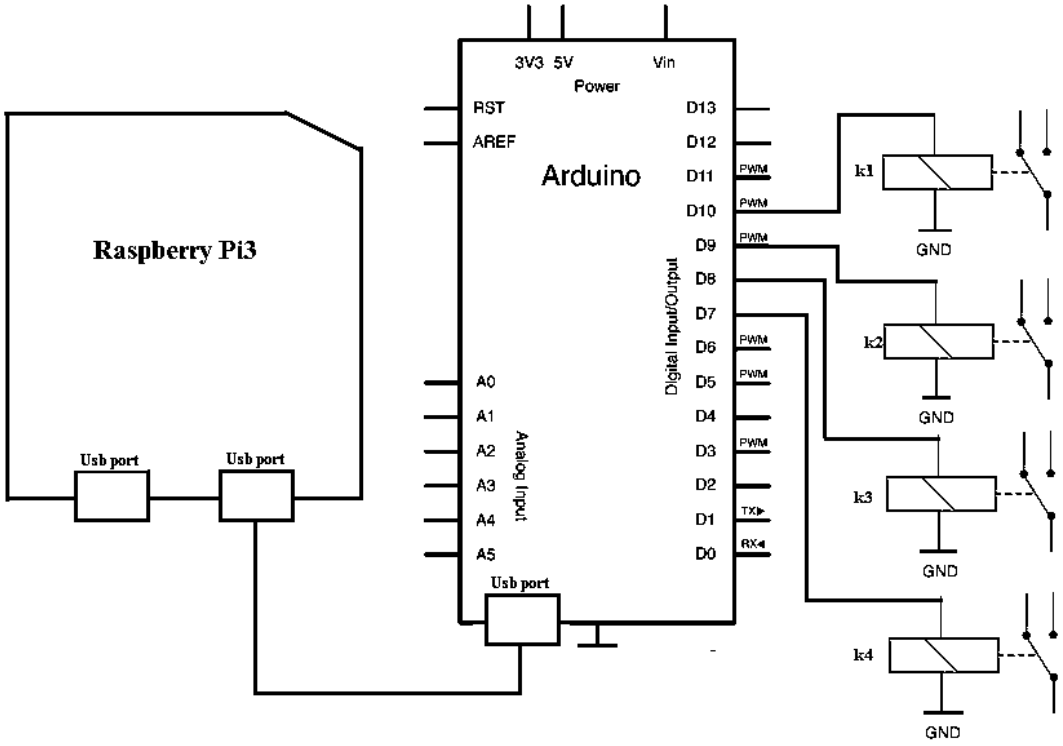


Figura 21-2: Diagrama electrónico de conexiones – salidas

Fuente: (Macías, 2018)

Tabla 3-2- Elementos para el control de las salidas del sistema

ELEMENTO ELÉCTRICO	NOMENCLATURA EN EL CIRCUITO	FUNCIÓN
Relé 1	K1	Interfaz de potencia para control de la electroválvula dosificación
Relé 2	K2	Interfaz de potencia para control de la electroválvula tapado

Relé 3	K3	Interfaz de potencia para control de la electroválvula sellado
Relé 4	K4	Interfaz de potencia para la bomba
Salida digital del Arduino pin 10	D10	Señal de control del relé 1 (HIGH, LOW)
Salida digital del Arduino pin 9	D9	Señal de control del relé 2 (HIGH, LOW)
Salida digital del Arduino pin 8	D8	Señal de control del relé 3 (HIGH, LOW)
Salida digital del Arduino pin 7	D7	Señal de control del relé 4 (HIGH, LOW)
ARDUINO	ARDUINO MEGA	Microcontrolador en cargado de la gestión de señales para el control de las salidas
RASPBERRY	RASPBERRY PI3	Procesamiento y monitoreo de señales

Realizado por: Macías, 2018

Fuente: Autor

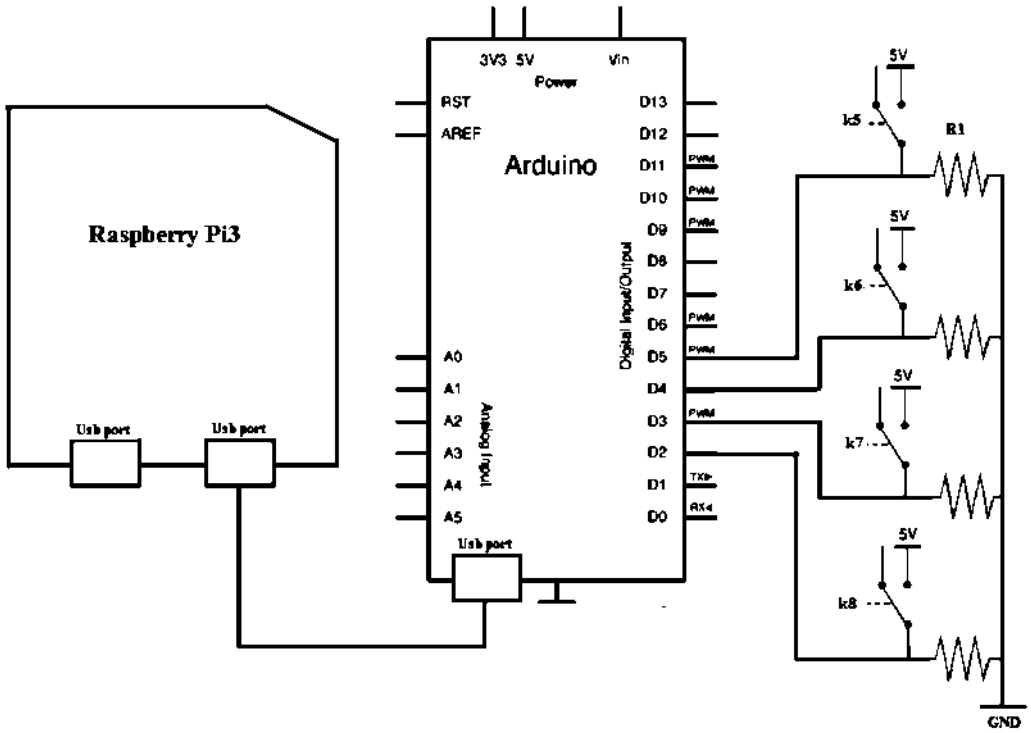


Figura 22-2: Diagrama electrónico de conexiones - entradas

Fuente: (Macías, 2018)

Tabla 4-2- Elementos para adquisición de señales

ELEMENTO ELÉCTRICO	NOMENCLATURA EN EL CIRCUITO	FUNCIÓN
Entrada digital del Arduino pin 4	D4	Señal para el control de la etapa de transporte
Entrada digital del Arduino pin 3	D3	Señal para el control de la etapa de dosificación
Entrada digital del Arduino pin 2	D2	Señal para el control de la etapa de tapado
Contacto del relé 5	K5	Accionamiento de la entrada para asignación de nivel HIGH o LOW relacionados con detección o no del tarro en la etapa inicial
Contacto del relé 6	K6	Accionamiento de la entrada para asignación de nivel HIGH o LOW relacionados con detección o no del tarro en la etapa de dosificación
Contacto del relé 7	K7	Accionamiento de la entrada para asignación de nivel HIGH o LOW relacionados con detección o no del tarro en la etapa de tapado

Realizado por: Macías, 2018

Fuente: Autor

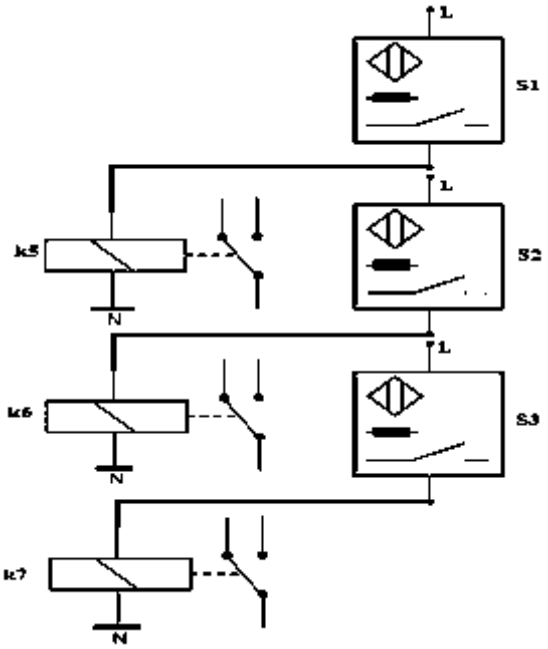


Figura 23-2: Diagrama eléctrico de conexiones - sensores

Tabla 3-2- Elementos de senso en el proceso

ELEMENTO ELÉCTRICO	NOMENCLATURA EN EL CIRCUITO	FUNCIÓN
Sensor Inductivo 1	S1	Detección de tarros metálico entrada al proceso.
Sensor Inductivo 1	S2	Detección de tarros metálico en el proceso de dosificación.
Sensor Inductivo 1	S3	Detección de tarros metálico en el proceso de tapado.
Relé 5	K5	Sistema de adecuación de señal del sensor inductivo 1 para procesamiento de señal en el microcontrolador
Relé 6	K6	Sistema de adecuación de señal del sensor inductivo 2 para procesamiento de señal en el microcontrolador
Relé 7	K7	Sistema de adecuación de señal del sensor inductivo 3 para procesamiento de señal en el microcontrolador

Realizado por: Macías, 2018

Fuente: Autor

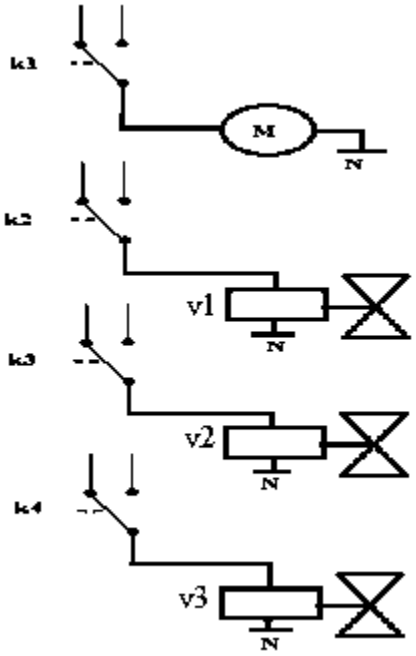


Figura 24-2: Diagrama eléctrico de conexiones - actuadores

Fuente: (Macías, 2018)

Tabla 5-2- Elementos para accionamiento de actuadores

ELEMENTO ELÉCTRICO	NOMENCLATURA EN EL CIRCUITO	FUNCIÓN
Electroválvula 1	V1	Electroválvula de control del cilindro para dosificación
Electroválvula 2	V2	Electroválvula de control del cilindro para dosificación de tapas
Electroválvula 3	V3	Electroválvula de control del cilindro para presión de la tapa y sellar.
Contacto del relé 5	K5	Accionamiento de la entrada para asignación de nivel HIGH o LOW relacionados con detección o no del tarro en la etapa inicial
Contacto del relé 6	K6	Accionamiento de la entrada para asignación de nivel HIGH o LOW relacionados con detección o no del tarro en la etapa de dosificación
Contacto del relé 7	K7	Accionamiento de la entrada para asignación de nivel HIGH o LOW relacionados con detección o no del tarro en la etapa de tapado

Realizado por: Macías, 2018

Fuente: Autor

2.4.5 Distribución del tablero eléctrico & electrónico

Haciendo uso de la herramienta SolidWorks se elabora un plano referencial del posicionamiento de los elementos eléctricos y electrónicos dentro de un tablero metálico.

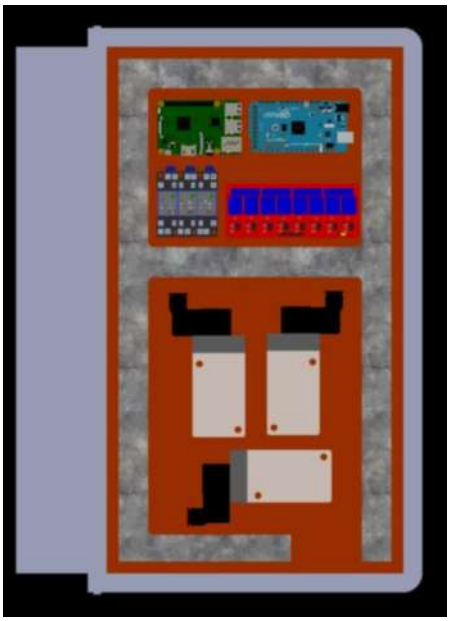


Figura 25-2. Diseño CAD tablero eléctrico & electrónico

Fuente: (Macías, 2018)

El sistema completo se lo puede representar mediante un diagrama de bloques que describe:

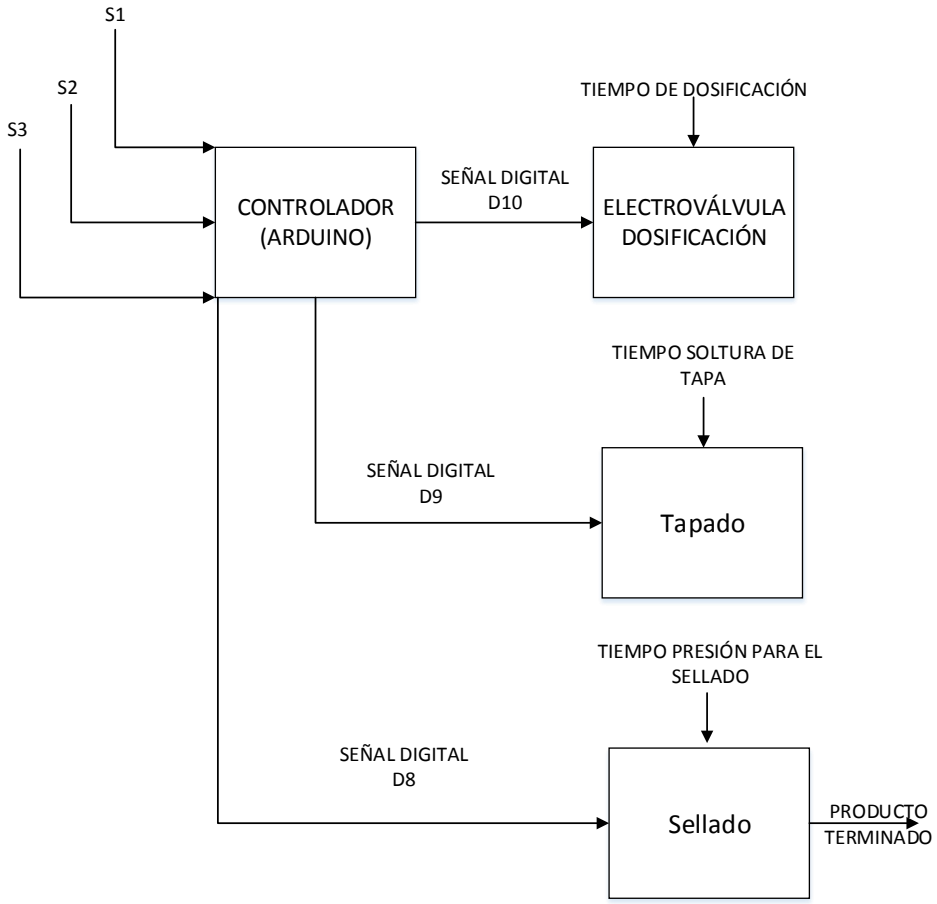


Figura 26-2. Diseño CAD tablero eléctrico & electrónico

Fuente: (Macías, 2018)

2.5 Componentes software del prototipo

2.5.1Configuración inicial - Arduino Mega

Establecido el modelo de placa de Arduino y todos los recursos a utilizarse para el manejo y control del proceso, se instala el software IDE (Arduino) junto a los drivers necesarios para poder programarlo. Una vez ejecutada la aplicación ya en su entorno de desarrollo se selecciona la pestaña Herramientas opción Placa y se selecciona en este caso la placa Arduino/Genuino Mega or Mega 2560, como se muestra en la figura 21-2.

Posterior se establecerá el puerto de comunicación ejecutando nuevamente la pestaña herramientas opción puerto.

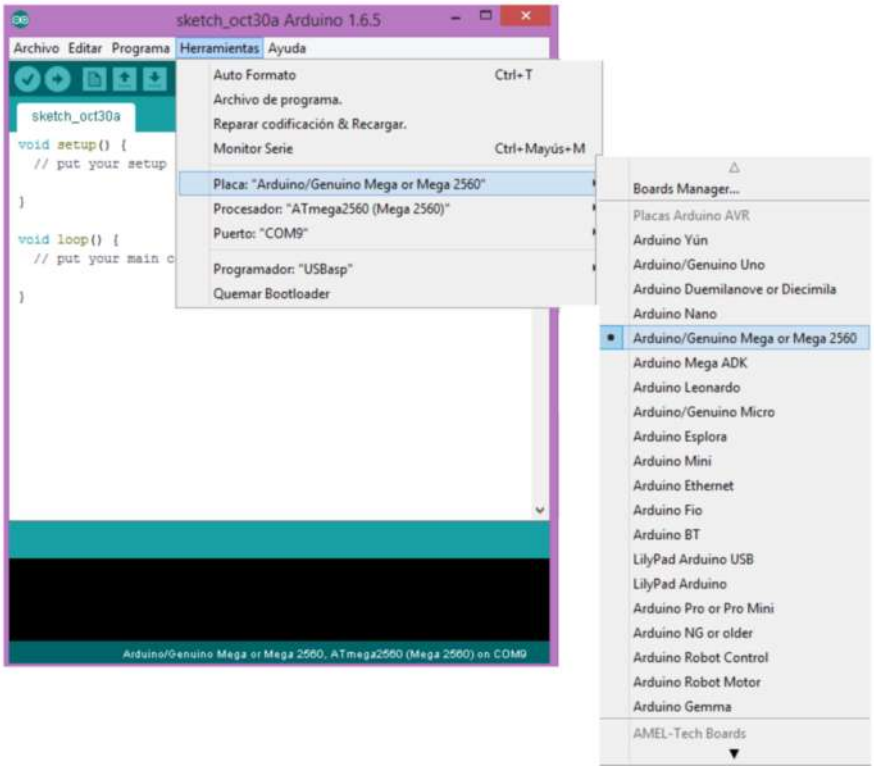


Figura 26-2. Selección de la Placa Arduino en el IDE Arduino

Fuente: (Macías, 2018)

2.5.2 Configuración inicial - Raspberry PI3

En la página oficial de Raspberry PI (raspberrypi.org) se tiene acceso a la descarga libre del software para el sistema embebido en las versiones NOOBS y Raspbian.

Para el desarrollo de esta aplicación se descarga la versión de software Raspbian que presenta dos versiones actualizadas del sistema operativo, se descarga la versión Full en formato .zip, el mismo que al descomprimirlo proporcionará una imagen de disco que será la que se monte en la SD card de la Raspberry.

Realizada la descarga del sistema operativo se procede a ubicar en el computador una micro SD de clase diez, para que pueda correr sin problemas el S.O. Se formatea directamente en un formato básico y se asigna un nombre a la tarjeta. Una vez preparada la SD card se requiere de un programa adicional como el WinDiskImage que se lo descarga de forma gratuita, éste sirve para enrutar la información de la imagen descargada hacia la SD card para de esta manera estar lista para trabajar con la Raspberry Pi3. (Aguilar, 2017)

Figura 27-2. Carga de la imagen de Raspbian en la SD - Win32DiskImage
Fuente: (Macías, 2018)

2.5.3 Programación del sistema de control - Arduino

Instalada y reconocida la placa en el IDE de Arduino se procede a desarrollar el bloque de instrucciones a seguirse para establecer el modo de funcionamiento del proceso.

En la primera etapa del programa se define las variables, constantes, contadores, acumuladores, entre otros recursos sobre los que se va a trabajar durante el desarrollo del programa. En el programa se ha añadido comentarios indicando la relación de las variables del programa más relevantes que guardan relación con el entorno físico del proceso.

Para descripción del algoritmo programado en el microcontrolador se realiza el siguiente diagrama de flujo de instrucciones.

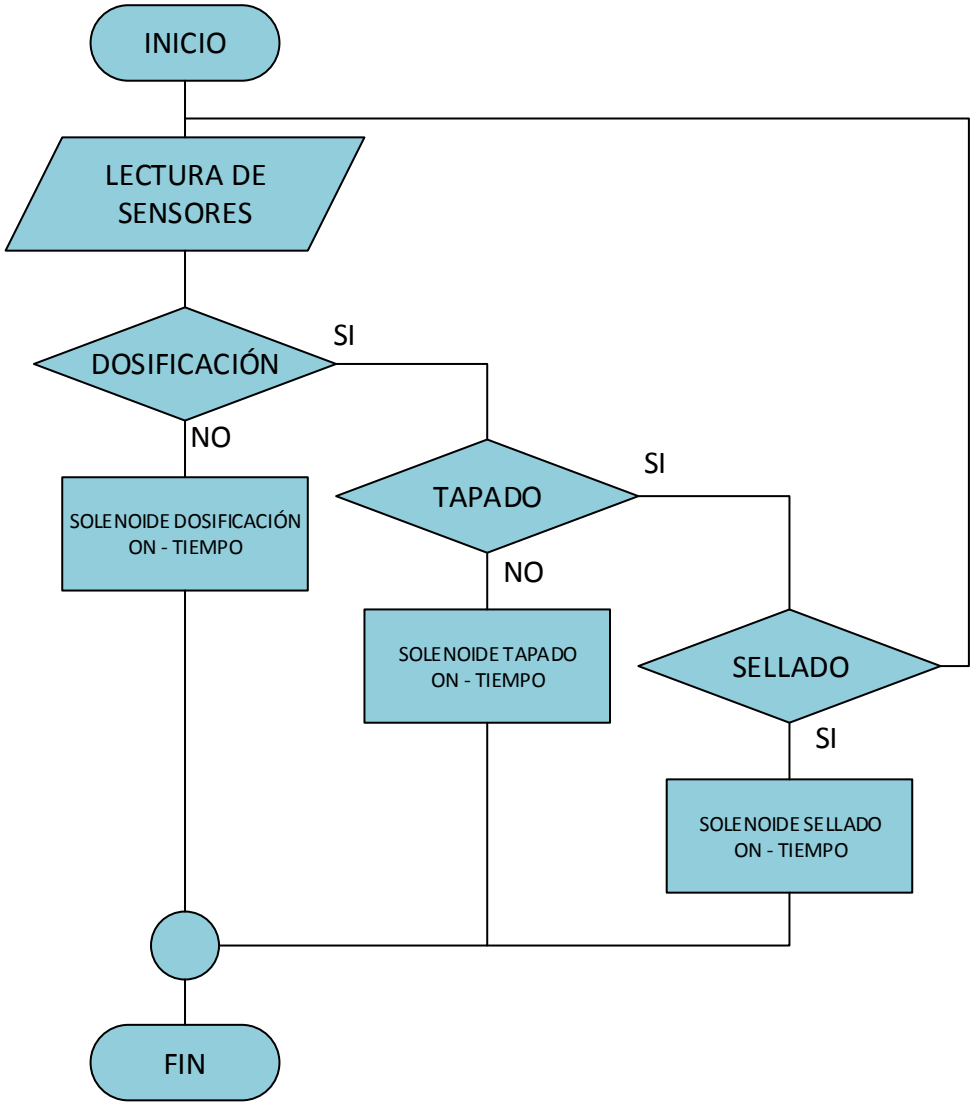


Figura 28-2. Diagrama de flujo – Algoritmo microcontrolador

Fuente: (Macías, 2018)

2.5.4 Programación del sistema de monitoreo – Raspberry Pi3

Para el desarrollo de la función de monitoreo se requiere realizar dos tipos de programaciones. La primera orientada a realizar la comunicación entre el Arduino que gobierna la parte de control del proceso con la Raspberry para el monitoreo y la segunda para el desarrollo de la comunicación entre la Raspberry Pi3 y el HMI.

2.5.5 Comunicación Arduino – Raspberry PI3

La configuración para efectuar la integración de los sistemas embebidos Arduino y Raspberry PI3 se la realiza mediante el ingreso de comandos al terminal de la Raspberry. Se parte de la ejecución del comando ***sudo apt-get install upgrade,*** para cargar las últimas actualizaciones para el dispositivo e instalarlas.

Posteriormente se habilita la comunicación serial por medio del comando ***sudo apt-get install python-serial*.**

Efectuados estos comandos en el terminal se realiza la conexión del Arduino Mega a un puerto USB de la Raspberry PI3 y se ejecuta el código ***lsusb*** para verificar si se estableció la conectividad entre los sistemas embebidos.

Para la verificación del puerto de la Raspberry en el que ha sido reconocido el Arduino se ejecuta el comando **ls/dev/tty*.** Se obtiene algo similar a **/dev/ttyACM0.**

Con la realización de este último paso se determina la comunicación entre los sistemas embebidos. En la programación del microcontrolador se imprime las ordenes de sensores y actuadores al puerto serial por medio de la instrucción *Serial.println(),* al estar conectado Arduino –Raspberry se puede leer y recibir datos en ambas direcciones. Los datos enviados por el microcontrolador serán tomados por la aplicación desarrollada en la Raspberry y sirve para realizar el monitoreo del proceso en tiempo real.

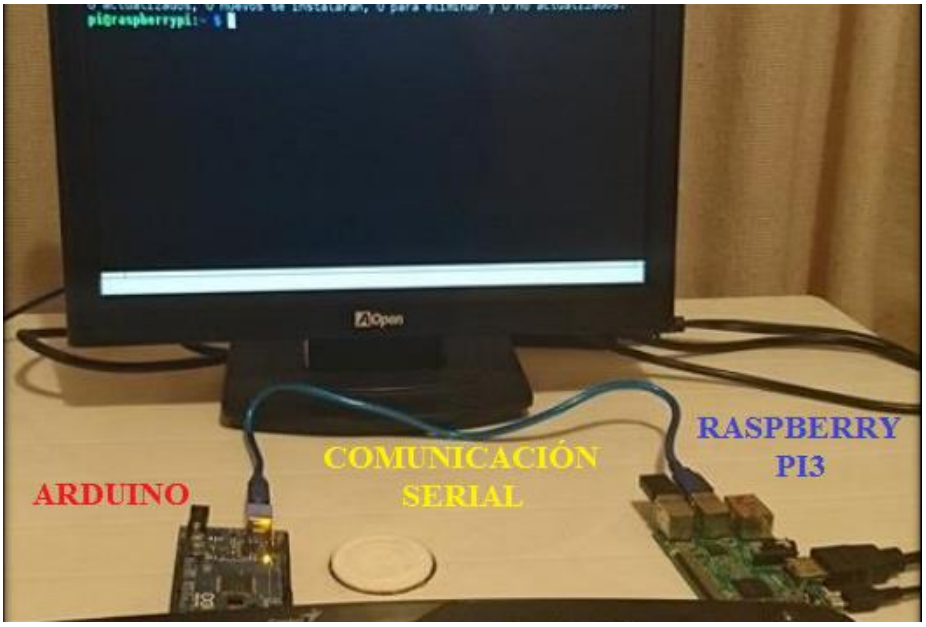


Figura 29-2. Integración Arduino – Raspberry Pi3

Fuente: (Macías, 2018)

2.5.6 Desarrollo de la Aplicación para el monitoreo

El desarrollo de aplicaciones dentro del entorno de Raspberry se efectúa bajo el uso de la plataforma de programación Python que brinda prestaciones potenciales.

Para el desarrollo de la interfaz de monitoreo se utiliza una aplicación multiplataforma como es QT Creator que presenta librerías de recursos gráficos enfocado a una programación orientada a objetos que se relaciona directamente con Python.

QT Creator se lo instala a través del terminal de la Raspberry PI3, se ejecuta los comandos:

sudo apt-get install build-essential

sudo apt-get install qt5-default

sudo apt-get install qtcreator

Figura 30-2 QT para Python

Fuente: https://www.solvetic.com/uploads/monthly_10_2015/tutorials-1415-0-86302100-1445975538.jpg

Ejecutados estos comandos se puede acceder a la aplicación desde el botón de inicio de aplicaciones en la pestaña de programación.

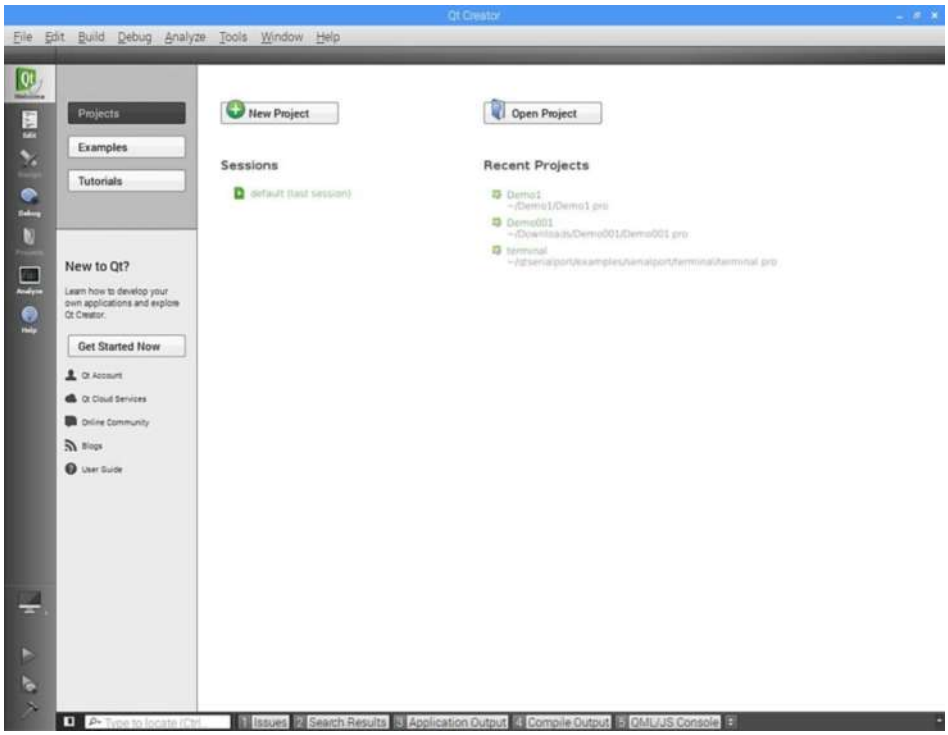


Figura 31-2 QT Creator – Ventana Inicial

Fuente: (Macías, 2018)

Para crear un nuevo proyecto se presiona el botón **New Project** que generará la aparición de una nueva venta en la que se deberá seleccionar el tipo de proyecto a desarrollarse, en el caso de este trabajo se selecciona una **Qt Quick Application.**

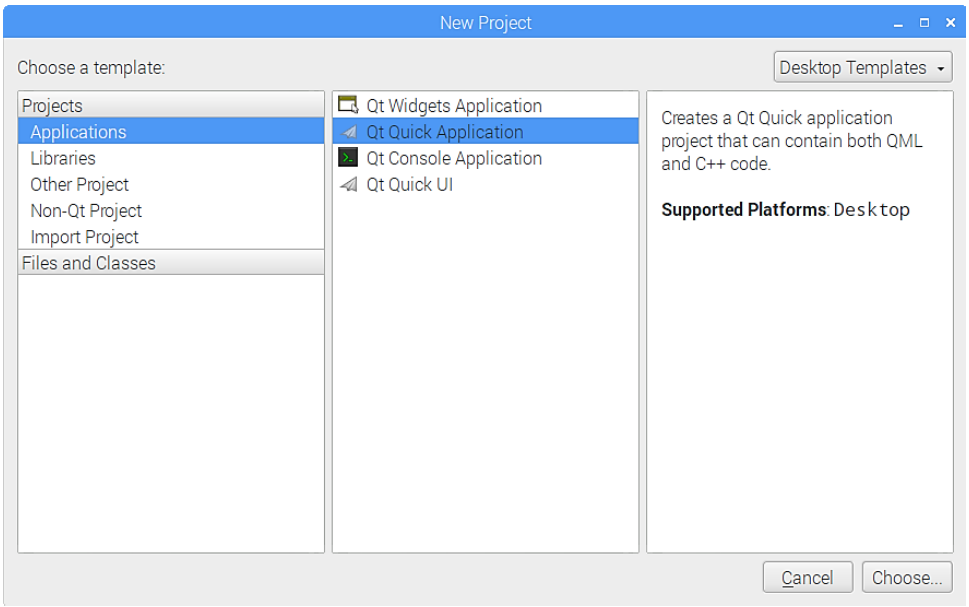


Figura 32-2 Selección tipo de proyecto

Fuente: (Macías, 2018)

Luego de la selección del proyecto aparece la ventana del entorno gráfico de programación para el desarrollo de la aplicación.

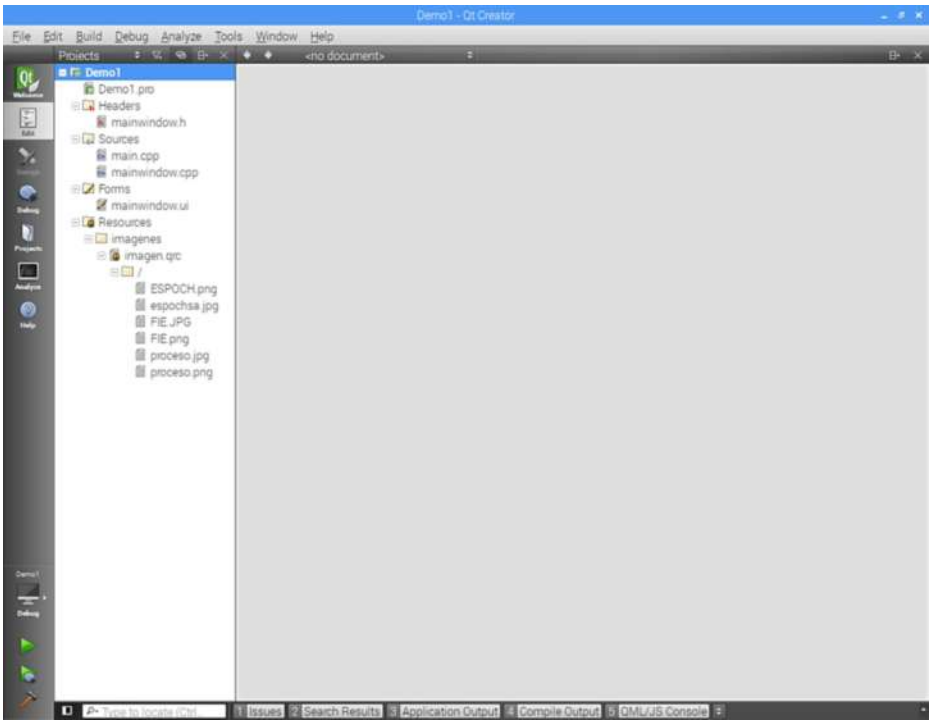


Figura 33-2 Entorno de programación Qt Creator - Python

Fuente: (Macías, 2018)

La figura 29-2 muestra la interfaz gráfica de la plataforma combinada donde en la franja de color blanco se denotan todos los recursos asignados al proyecto. Los de mayor relevancia para la aplicación a desarrollarse son Source, Form y Resource, donde:

Source, contiene los archivos de programación .cpp en los que se desarrolla la programación en Python necesaria para el monitoreo del proceso, el bloque de instrucciones necesarios para interpretar las señales del exterior.

Form, aquí se anidan las ventanas o pantallas que diseñemos para la aplicación.

Resource, contiene los recursos necesarios para ser citados dentro del desarrollo de las forms.

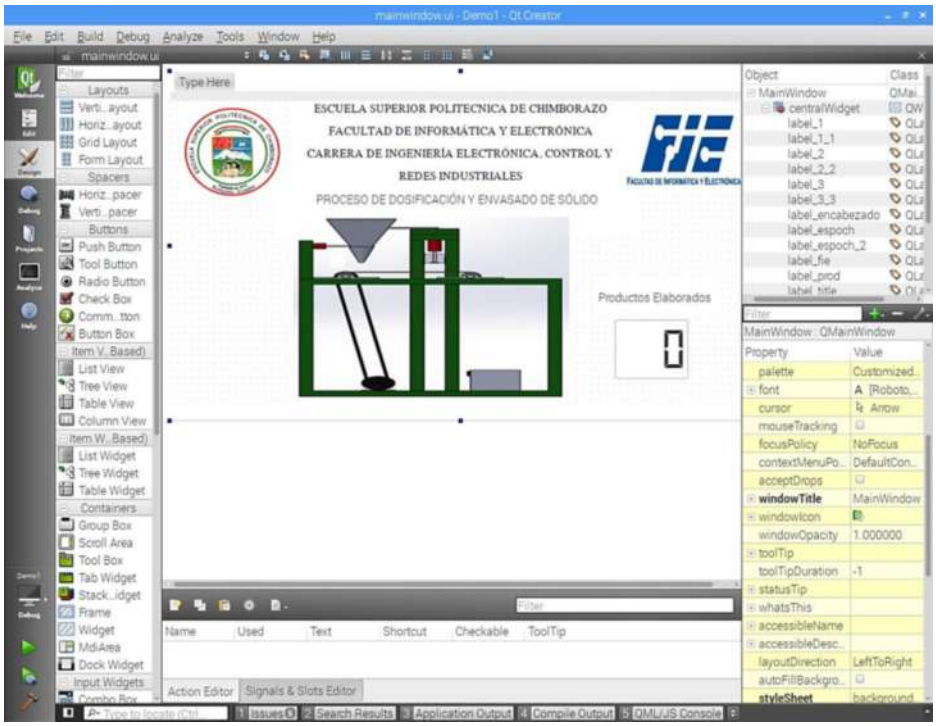


Figura 34-2 Diseño del Form

Fuente: (Macías, 2018)

La figura 30-2 muestra la ventana de desarrollo de la interfaz gráfica para el monitoreo del proceso. En esta ventana se tiene una amplia gama de recursos para inserción de textos, figuras, hipervínculos, entre otros (franja fondo blanco lateral derecha), además que cada recurso también puede ser configurado de acuerdo a parámetros establecidos (franja fondo blanco/naranja costado inferior izquierdo).

CAPÍTULO III

3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS DE RESULTADOS

3.1. Funcionamiento del prototipo

En base a los requerimientos planteados para el diseño del prototipo se ejecutó la implementación del mismo obteniendo una estructura sólida de Aleación de Aluminio 6063-O que tras el análisis estructural realizado garantiza su funcionalidad sin riesgos de ruptura o deformaciones.

Figura 1-3. Proceso prototipo implementado

Fuente: (Macías, 2018)

Mediante la integración de todos los elementos de máquina considerados en el diseño se logra tener una línea de proceso altamente aproximado a un proceso real, como se observa en la figura 1-3.

El proceso denota varias etapas, todas en solución a los requerimientos planteados inicialmente en la descripción del funcionamiento esperado de la línea de producción.

En la figura 2-3 se determina el inicio del proceso, donde el sensor inductivo situado al filo de la banda transportadora en el momento en que detecte el metal del tarro a ingresar al proceso, emitirá una señal al sistema embebido encargado del control y mediante su valoración en la programación se determinará el movimiento inmediato de la banda transportadora para generar el desplazamiento del tarro.

Figura 2-3. Detección inicial de tarros.

Fuente: (Macías, 2018)

Una vez iniciado el desplazamiento del tarro sobre la banda transportadora se mantiene la acción mecánica hasta llegar a la etapa de dosificación donde existe un sensor del tipo inductivo que cumple la función de una señal de paro para el motor de la banda y un arranque para el funcionamiento de un actuador. En este caso la electroválvula 5/2 controla el desplazamiento del cilindro de 150mm de carrera acoplado a la compuerta situada en la parte inferior de la tolva para la dosificación.

La dosificación se la realiza por un intervalo de 3.4 segundos, mediante modificación de código en el microcontrolador puede ser variable. La activación de la electroválvula durante este tiempo tendrá abierta la compuerta para caída del sólido al tarro de 150cc, culminado el tiempo

de dosificación, el cilindro retornará para de esta forma cerrar la compuerta y dar por terminada la etapa.

Figura 3-3. Etapa de dosificación implementada

Fuente: (Macías, 2018)

El fin del tiempo de dosificación marca una señal para nuevamente la activación del motor de la banda transportadora y desplazar el tarro hacia la etapa de colocación de tapa y sellado.

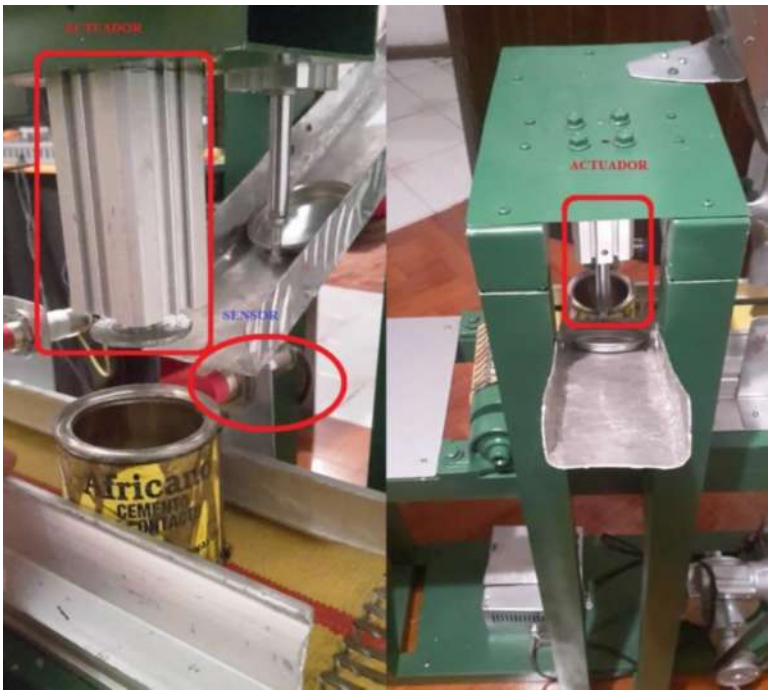


Figura 4-3. Etapa de Sellado.

Fuente: (Macías, 2018)

En la etapa de sellado o tapado se dispone de un sensor inductivo para que al momento en que el tarro llegue se apague el motor de la banda transportadora y se desarrolle el proceso.

La etapa del tapado se desarrolla en dos lapsos:

1. **La colocación de la tapa:** Según el diseño planteado se construyó una rampa en la que las tapas resbalan para colocarse sobre el tarro ya situado en posición. La contención del libre desplazamiento de las tapas la realiza un cilindro por medio de su embolo, donde la electroválvula controla su accionamiento al inicio del proceso. Es así que con la llegada del tarro el microcontrolador desactiva por un lapso de 2.5 segundos la electroválvula para que el cilindro retorne y permita el desplazamiento de una sola tapa a la vez.

2. **Sellado de la tapa:** Se lo realiza a presión por medio de un cilindro que en la punta de su embolo tiene un cabezal que va a ejercer un choque frontal con la tapa y a su vez ejercer una presión que puede ir desde los 0.1 hasta los 0.9 Mpa, para sellar el tarro seleccionado.

Efectuadas las dos acciones descritas el tarro avanza con el accionar del motor de la banda transportadora, como se muestra en la figura 5-3.

Figura 5-3. Trasmisión de movimiento a la banda transportadora.
Fuente: (Macías, 2018)

3.2 Sistema de control

Los sensores inductivos adquiridos son de dos hilos a 110VCA por lo que para la inserción de sus señales al microcontrolador se implementó una etapa de adecuación de señal por medio de relés, en la figura 6-3 se muestra el circuito resultante del sistema embebido Arduino con todas sus entradas y salidas.

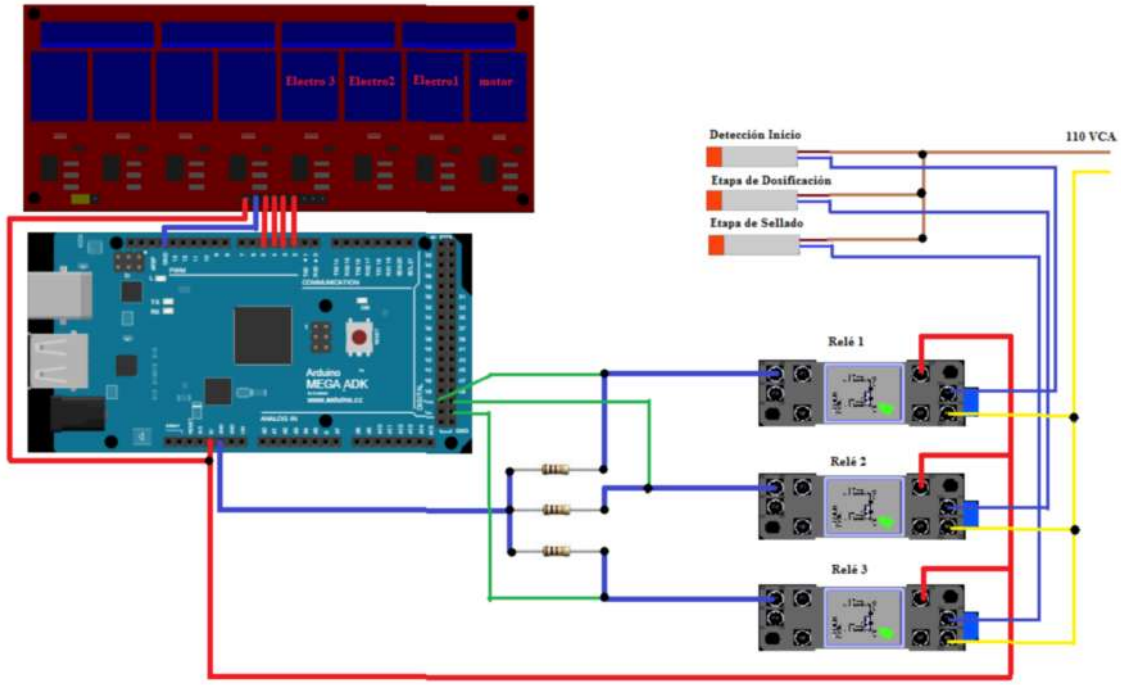


Figura 6-3 Circuito de control (Anexo F)

Fuente: (Macías, 2018)

La figura 7-3 muestra la conexión de la tarjeta Raspberry Pi con la tarjeta Arduino a través de comunicación serial.

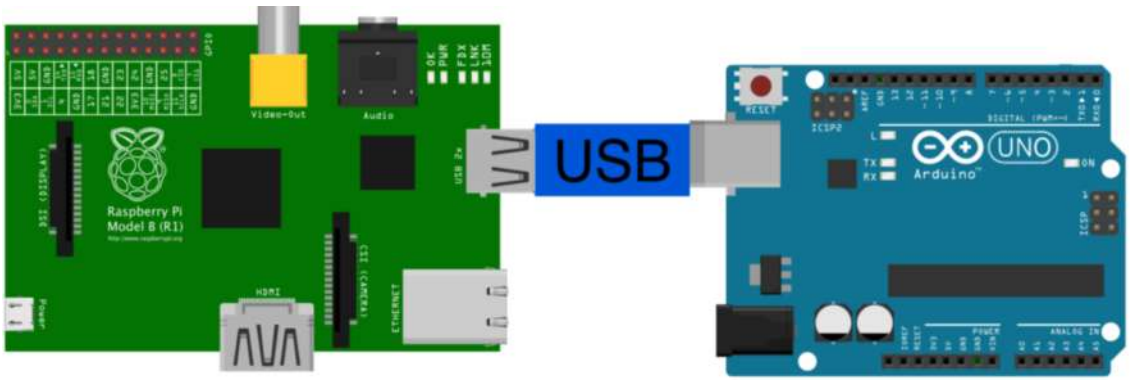


Figura 7-3 Comunicación Serial

Fuente: (Macías, 2018)

La tarjeta Arduino al mismo tiempo que ejecuta las acciones de control emite señales al puerto serial, las mismas que son adquiridas por la Raspberry PI3 por medio de una aplicación diseñada a partir de QT Creator con soporte para Python.

Basados en el diseño planteado en la distribución de elementos para el tablero eléctrico & electrónico se obtiene la implementación del mismo como se puede observar en la figura 8-3.

Figura 8-3 Tablero eléctrico & electrónico implementado

Fuente: (Macías, 2018)

Con el tablero implementado se realizan pruebas de funcionalidad en el laboratorio para posterior instalación en el gabinete. Los sensores inductivos detectan la presencia del envase en las distintas etapas.

3.3 Sistema de monitoreo

La figura 9-3 muestra la pantalla inicial resultante para el monitoreo del proceso, describe una interfaz amigable para el usuario. Donde se resalta como bondades del QT Creator la facilidad de inserción de imágenes para personalizar las pantallas.

Se insertó en la interfaz una de las vistas de los diseños de SolidWork para dar mayor impacto visual, creando un monitoreo con un proceso idéntico al obtenido en la implementación.

Además, en esta interfaz se implementó un control de producción donde cada producto terminado (salida de la última etapa del proceso) se registra en un contador.

Figura 9-3 Pantalla inicial de la interfaz

Fuente: (Macías, 2018)

Con el bloque de control de los componentes del prototipo ya implementado, se procedió a enlazar la parte del monitoreo con pruebas externas al proceso y se comprobó de manera eficiente la comunicación entre los sistemas embebidos.

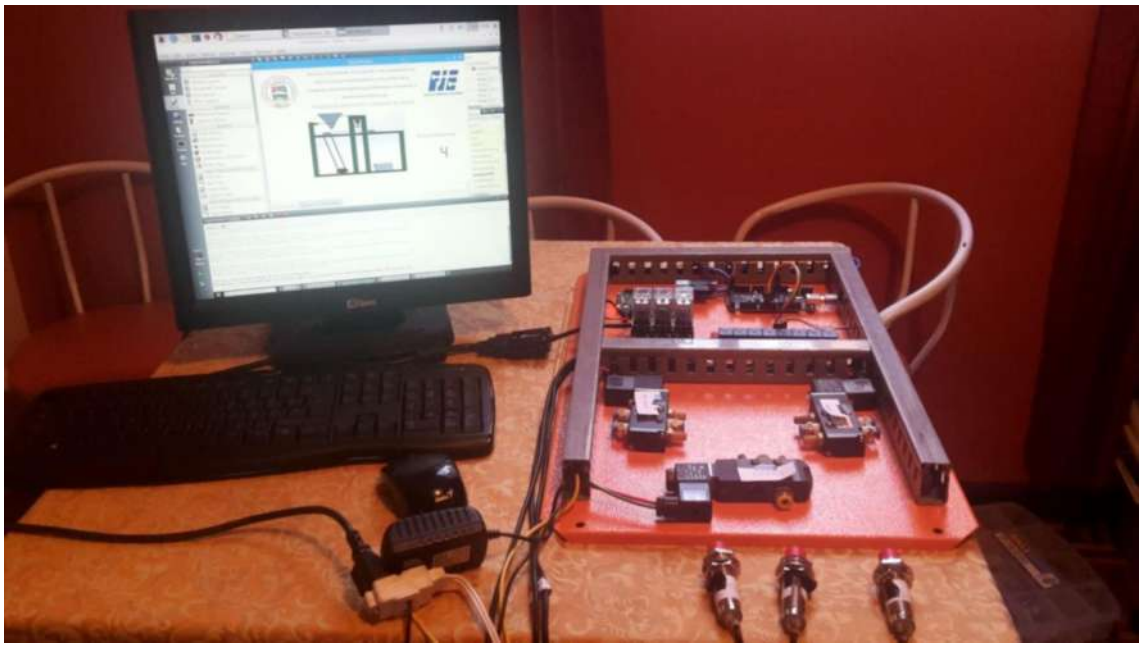


Figura 10-3 Comunicación exitosa Sistema de control y monitoreo

Fuente: (Macías, 2018)

Establecida la comunicación se verifica los puntos de monitoreo establecidos tanto para ingreso, llenado y sellado dentro del proceso, como se muestran en las figuras 11-3; 12-3; 13-3.

Figura 11-3 Monitoreo funcionalidad de la banda transportadora

Fuente: (Macías, 2018)

Figura 12-3 Monitoreo funcionalidad etapa de dosificación

Fuente: (Macías, 2018)

Figura 13-3 Monitoreo funcionalidad etapa de tapado y sellado

Fuente: (Macías, 2018)

3.4 Datos de consumo

Para verificar la eficiencia del prototipo se ejecuta un análisis de consumo energético de los elementos que conforman la parte eléctrica y electrónica de los circuitos de control y monitoreo.

Tabla 1-3 Consumo de dispositivos eléctricos & electrónicos

Cantidad	Dispositivo	Consumo [mA]
1	Raspberry Pi3	2500
1	Arduino MEGA	96
3	Válvulas Solenoides (1,4 w)	12,72
3	Relé Sensores (Medición de 210 Ohm en la bobina, 110VCA)	523,80
4	Relé Módulo (Medición de 90 Ohm en la bobina 5VCD)	5,55
3	Sensor Inductivo	19
1	Motor	1000

Realizado por: Macías, 2018

Fuente: Autor

Tabla 2-3 Consumo de dispositivos eléctricos & electrónicos por etapas del proceso

Inicio
Raspberry (2500[mA])
Arduino (96[mA])
Válvula Tapado (12,72[mA])
Sensor Inductivo (19[mA])
2531,72[mA]
Transporte
Raspberry (2500[mA])
Arduino (96[mA])
Válvula Tapado (12,72[mA])
Motor (1000[mA])
3608,72[mA]
Dosificación
Raspberry (2500[mA])
Arduino (96[mA])
Válvula Tapado (12,72[mA])
Válvula Dosificación (12,72[mA])
Sensor Inductivo (19[mA])
2640,44[mA]
Tapado
Raspberry (2500[mA])
Arduino (96[mA])
Sensor Inductivo (19[mA])

2615[mA]
Sellado
Raspberry (2500[mA])
Arduino (96[mA])
Válvula Tapado (12,72[mA])
Sensor Inductivo (19[mA])
2627,72[mA]

Realizado por: Macías, 2018

Fuente: Autor

La tabla 2-3 describe el consumo de corriente por etapas del sistema medidos en un ciclo completo para procesar un tarro.

Se describe como etapa de inicio cuando el tarro ingresa a la banda y se representa en la tabla los elementos que funcionan en ese instante dando un consumo de corriente de 2531,72[mA]. La etapa de transporte 3608,72[mA]. La etapa de dosificación 2640,44[mA]. La etapa de tapado 2615[mA]. La etapa de sellado 2627,72[mA].

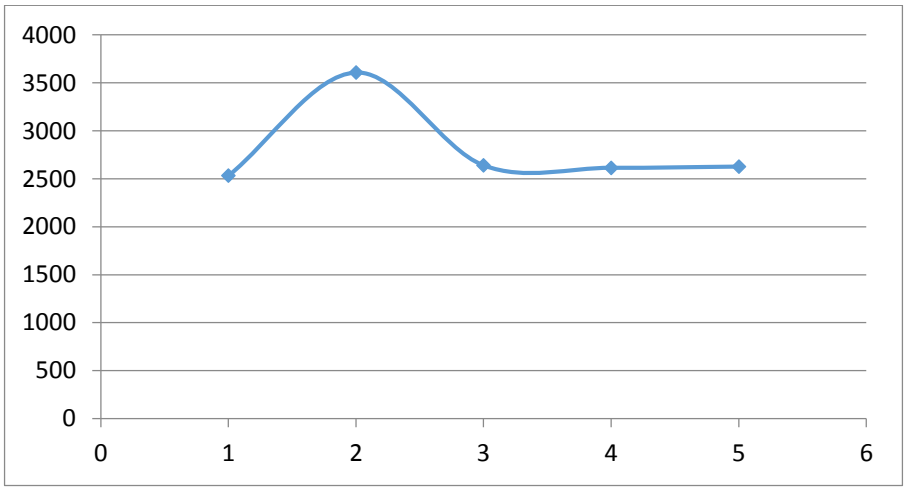


Figura 14-3. Comportamiento consumo de corriente etapas del proceso.

Fuente: (Macías, 2018)

De los resultados expuestos en la tabla se determina que la etapa de mayor consumo es la de transporte y que los sistemas embebidos para el control y monitoreo son constantes en todas las etapas.

3.5 Comparación del desempeño con PLC

En el prototipo de control y monitoreo que se desarrolló, se ha evaluado y comparado en cuanto a desempeño se refiere, en tres parámetros distintos: tiempo de respuesta, robustez en el ambiente de trabajo y el precio.

1. El tiempo de respuesta en el PLC y en el prototipo desarrollado es similar ya que, si se necesitare más o menos velocidad en este parámetro, simplemente modificaríamos la programación.
2. La robustez en el ambiente de trabajo es algo en lo que el PLC tiene ventajas, ya que esta desarrollado y compactado para aguantar perturbaciones de ruido y vibración sin que afecte el desempeño del mismo, sin embrago a nuestro prototipo implementado se le puede construir una caja cerrada herméticamente para aislarlo de este tipo de perturbaciones en el ambiente de trabajo que podrían afectar su desempeño y amenorar su tiempo de vida útil, ya que en base a resultados obtenidos el prototipo funciona correctamente en un ambiente libre de ruido y vibraciones.
3. El precio fue el punto de inicio para comenzar con el desarrollo de este prototipo, ya que se planteó obtener un sistema de monitoreo y control en base a sistemas embebidos de clase OPEN SOURCE, y se logró obtenerlo con excelentes resultados en el escenario propuesto. En este parámetro nuestro prototipo tiene ventaja, ya que el ahorro es significativo y el beneficio desempeño-costo es muy bueno.

El escenario propuesto para el prototipo desarrollado fue algo escogido al azar, sin embargo, haciendo una comparativa en cuanto a eficiencia-beneficio para el dueño de una PYME podemos decir, que al implementar y tecnificar procesos de este tipo obtendrá una producción más alta y a su vez mejores ganancias, ya que se evitaría pagar un sueldo a la persona que hace el llenado y sellado de producto, y a su vez al tener un sistema de este tipo tendría un control exacto de su producto terminado. Por otra parte, un módulo automatizado hará la tarea de una persona tan rápido como se lo programe, para nuestro caso en particular se demora aproximadamente 7 segundos en llenar, poner la tapa y sellar un tarro.

CAPÍTULO IV

4. COSTOS

Se realiza un análisis general de los costos efectuados en la construcción del modelo prototipo del proceso de dosificación de sólidos y sellado de tarros de 125cc de litro.

4.1 Costos directos

Son los que se relacionan directamente para la realización o fabricación del modelo prototipo.

4.1.1. Costos Neumáticos

Tabla: 1-4: Costos neumáticos

Cantidad	Descripción	V. unitario(USD)	V. Total (USD)
3	Válvulas electroneumáticas 5/2 110V	55	165
6	Silenciadores 1/8	1	6
6 mts	Manguera plástica synflex	0,8	4,8
2	Cilindros de doble efecto C.100 mm	40	80
1	Cilindros de doble efecto C.150 mm	50	50
2	Teflón	0,75	1,5
8	Racores	1,75	14
1	Unidad de mantenimiento	45	35
		Total	**356,3**

Realizado por: Macías, 2018

Fuente: Autor

4.1.2 Costos Eléctricos

Tabla: 2-4: Costos eléctricos

Cantidad	Descripción	V. unitario(USD)	V. Total (USD)
1	Motor eléctrico 12V	65	65
1	Fuente alimentación 12 V - 5ª	7	7
1	Fuente alimentación 12 V - 30ª	40	40
1	Raspberry PI3	65	65
1	Monitor	120	120
1	Arduino Mega	27	27
1	Módulo 8 relés	15	15
3	Relé 110VCA	7	21
3	Sensores Inductivos 110VCA	30	90
1	Adaptador HDMI-VGA	16	16
30	Cable flexible Nº 16	0,30	9

4	Cable flexible Nº 12	0,45	1,8
1	Gabinete Eléctrico	45	45
2	Pulsadores	3	6
3	Canaleta Ranurada	5,25	15,75
		Total	**543,55**

Realizado por: Macías, 2018

Fuente: Autor

4.1.3 Costos mecánicos

Tabla: 3-4: Costos mecánicos

Cantidad	Descripción	V. unitario(USD)	V. Total (USD)
2	Chumaceras 3/4"	5	10
2	Rodillos	10	20
2	eje 3/4 "	2	4
1	Cinta transportadora	35	35
2	Poleas de aluminio 13 cm	8	16
6	Tubo cuadrado Aluminio	3	18
8	Accesorios L, T	2,6	20,8
1	Tol (Tolva)	10	10
1	Funda Remaches	8	8
		Total	**141,8**

Realizado por: Macías, 2018

Fuente: Autor

4.1.4 Costo de mano de obra

Tabla: 4-4: Costos mano de obra

Cantidad	Descripción	V. unitario(USD)	V. Total (USD)
1	Mecánico	50	50
1	Tornero	30	30
		Total	80

Realizado por: Macías, 2018

Fuente: Autor

4.1.6 Costos directos totales

Tabla: 6-4: Costo directos totales

Descripción	V. Total
Costos neumáticos	356,3
Costos eléctricos	543,55
Costos mecánicos	141,8
Costo mano de obra	80
Costo directo total	**1121,65**

Realizado por: Macías, 2018

Fuente: Autor

4.2 Costos indirectos

Tabla: 7-4: Costo indirectos totales

Nª	Descripción	V. unitario(USD)	V. Total (USD)
1	Ingenieriles (supervisión y diseño)	200	200
2	Imprevistos	50	50
		Total	**250**

Realizado por: Macías, 2018

Fuente: Autor

4.3 Costo total

Tabla: 8-4: Costo total

Nª	Descripción	V. unitario(USD)	V. Total (USD)
1	Costos directos totales	1121,65	1121,65
2	Costos indirectos totales	250	250
		Costo Total	**1371,65**

Realizado por: Macías, 2018

Fuente: Autor

CONCLUSIONES

Por medio de la revisión bibliográfica se determinó que a nivel mundial se están desarrollando cada día más proyectos enfocados en el uso de tecnología open source, muchos son los desarrolladores que siguen alimentando a esta idea generando más sistemas embebidos de hardware y software que se van incluyendo en el mercado para la experimentación y aplicación de los mismo en diferentes campos.

La aplicación de los sistemas embebidos Arduino y Raspberry PI3 resultó eficiente al momento de considerarlos como alternativas para generar sistemas de control y de monitoreo, donde el Arduino se relaciona con el medio exterior evaluando variables por medio de sensores y en base a ellas controlando actuadores para la generación de acciones y la Raspberry empleada como medio de procesamiento de señales para reflejarlas en una interfaz gráfica.

En el prototipo destacan las variables de tipo digital pues se seleccionó sensores de presencia que señalarán estados discretos de igual manera que los actuadores, sin embargo, el sistema para trabajos futuros presenta la flexibilidad de incluir señales del tipo analógico para mejoras que se pueden plantear dentro del proceso.

El prototipo cumple con la expectativa planteada en el diseño de SolidWorks, basado en una estructura firme y robusta que garantiza la no ruptura ni su deformación, representa uno de los procesos más comunes como es la dosificación de sólidos y tapado de envases que se dan a nivel industrial y que puede adaptarse a las necesidades de las pequeñas y medianas industrias del país para el caso se necesitó 13N de fuerza y una presión de 0,9 Mpa.

Tras pruebas efectuadas en el prototipo se evidencia la funcionalidad consiguiendo demostrar que la automatización de procesos se puede dar con el uso de sistemas embebidos open source lo que representa una alternativa de bajo costo y tiempos mínimos en cada etapa del proceso que van desde los 3 s hasta los 5,5 s; con lo que permite la tecnificación de procesos a nivel de pequeñas y medianas industrias.

RECOMENDACIONES

La difusión de este tipo de aplicaciones en pequeñas y medianas empresas o industrias resultaría óptima para incentivar al empresario a tecnificar sus procesos sin inversiones demasiado grandes.

Promover el uso de hardware y software libre para el desarrollo de aplicaciones que puedan resolver problemáticas de la sociedad.

Al momento de realizar el diseño de una máquina, proceso o línea de producción emplear herramientas CAD que permiten modelar la ideas para luego plasmarlas con la seguridad de que no se van a presentar fallos.

BIBLIOGRAFÍA

ALLEN, P. & MOSCA, G. (2005). Física para la ciencia y la tecnología s.1. Reverte.

CZWIENCZEK, J. (2016). Sistemas Operativos para Sistemas Embebidos.(en línea) [Consultado: 25 de Marzo 2018]. Disponible en:http://catodocomun.blogspot.com/2016/02/componentes-de-un-sistema-embebido.html

DELGADO, A. (2007). ¿Qué es hardware libre? Noticias-Artículos-Actualidad [Consultado: 12 de Abril 2018]. Disponible en: http://www.paginadigital.com.ar/articulos/2007/2007prim/tecnologia41/hardware-mi-211107.asp

EISNER, C. (2005). Verificación formal del código fuente del software mediante modelado semiautomático. [Consultado: 12 de Abril 2018]. Disponible en: https://link.springer.com/article/10.1007/s10270-003-0042-x

HIBBELER, R.C. (2006). Mecánica de Materiales s.l. Pearson Educación.

HIPERTEXTUAL. (2013). 5 proyectos de hardware libre que debes conocer. [Consultado: 12 de Abril 2018]. Disponible en: https://hipertextual.com/2013/05/5-proyectos-de-hardware-libre-para-conocer

LÓPEZ, J. (2013). Módulo 2 Física. Ediciones Paraninfo S.A.

MORTON, T. (2000). Embedded Microcontrollers. Prentice Hall.

MOTT, R. (2006). Diseño de Elementos de Máquinas. Pearson Educación. México.

PEARCE, J. M. (2015). Maximizar la rentabilidad de la inversión para la salud pública con hardware médico de código abierto. Letters to the editor / Gac Sanit.

ORTIZ, D. (2017). Crecimiento de las pymes desde la automatización. América económica Análisis y Opinión. [Consultado: 28 de Marzo 2018]. Disponible en: https://www.americaeconomia.com/analisis-opinion/crecimiento-de-las-pymes-desde-la-automatizacion

RAMOS, M. (2015). Introducción a los sistemas de tiempo real. Universidad Nacional del Nordeste.

SANCHEZ, V., PIZARRO, D. (2010) Diagnóstico del nivel de automatización en las pequeñas y medianas industrias de la ciudad de Cuenca. Ingenius Revista de Ciencia y Tecnología. p. 44 - 56

SENPLADES, S. N. (2017). Buen vivir Plan Nacional 2013 - 2017. p.189. [Consultado: 25 de Febrero 2018]. Disponible en: http://www.buenvivir.gob.ec/objetivo-10.-impulsar-la-transformacion-de-la-matriz-productiva

SIMON, D. (1999). And Embedded Software Primer. Addison-Wesley Professional.

SOLIDWORKS. (1995). DASSAULT SYSTEMES. [Consultado: 25 de Marzo 2018]. Disponible en: http://help.solidworks.com/2013/spanish/SolidWorks/cworks/c_Factor_of_Safety_Check.html.

STAFF USERS. (2014). Plataformas Arduino y Raspberry Pi: Plataformas Arduino y Raspberry Pi. En D. Aranda, ELECTRÓNICA (pág. 320). Buenos Aires: USERS.

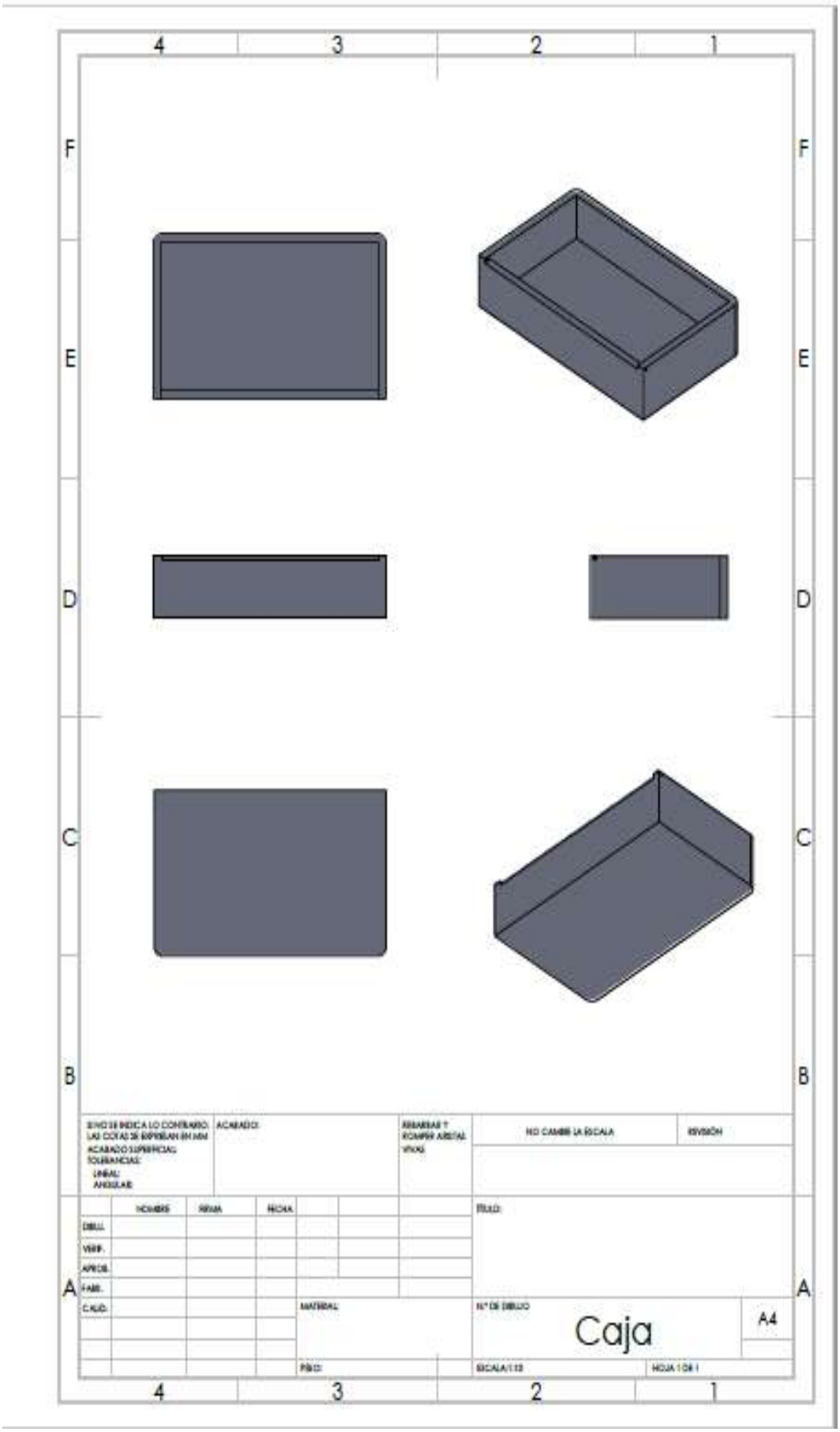
TCS INDUSTRIAL. (2014). Automatización para PYMES. [Consultado: 04 de Abril 2018] Disponible en: http://www.tcsindustrial.com/automatizacion-pymes/

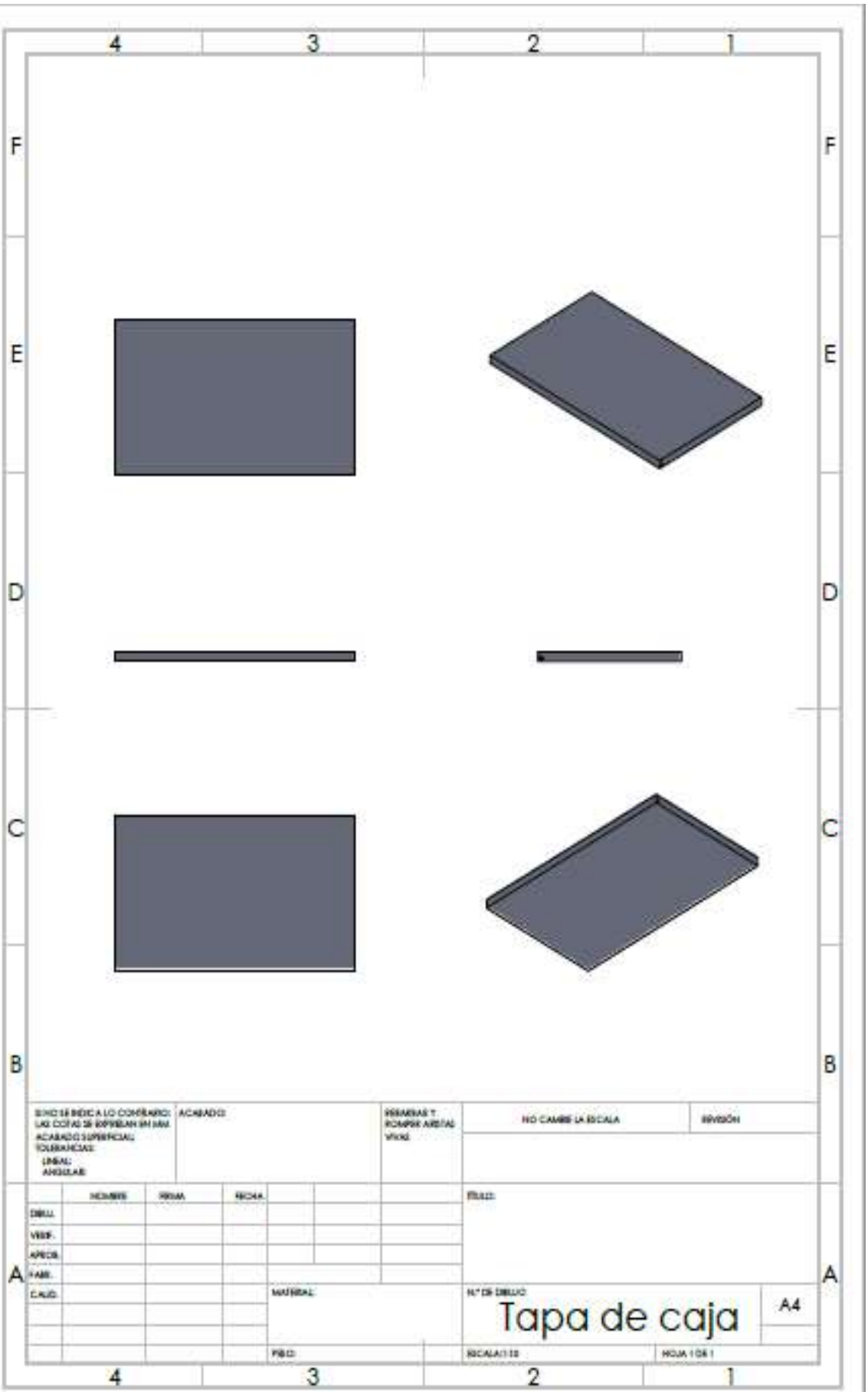
UNIVERSIDAD POLITÉCNICA DE CARTAGENA. (2013). Ingeniería Técnica Industrial, esp. En J. A. Sánchez García, Periférico de acceso al PC (pág. 139). Cartagena: S/N. [Consultado: 04 de Abril 2018] Disponible en: http://repositorio.upct.es/xmlui/bitstream/handle/10317/3486/pfc5273.pdf?sequence=1&isAllowed=y

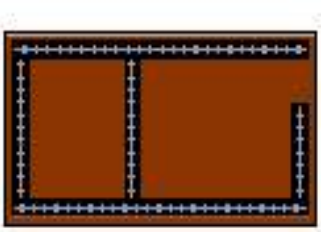
VILLAR, A. (2010). Introducción a la Informática y al uso y manejo de aplicaciones comerciales. Ideas propias. Editorial S.L.

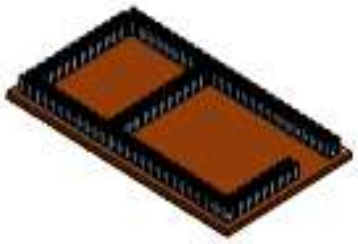
WOLF, W. (2008). Computers as Components: Principles or Embedded Computing System Design. Second Edition. Morgan Kaufmann. 2008.

ANEXOS

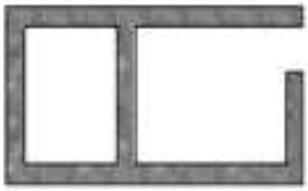


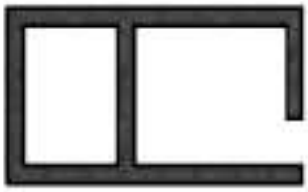

4
3
2
1
F
E
D
C
B
A
ACABADO
NO CAMBIE LA ESCALA
Tapa de caja
A4





SI NO SE INDICA LO CONTRARIO LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:			ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA		REVISIÓN
	NOMBRE	FIRMA	FECHA				TÍTULO:
DIBUJ.							
VERIF.							
APROB.							
FABR.							
CALID.				MATERIAL			Nº DE DIBUJO Base
							A4
				PESO:		ESCALA:1:10	HOJA 1 DE 1





ACABADO:

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.			
VISTO			
APROB.			
FABR.			
CALID.			

MATERIAL:

Nº DE DIBUJO

Tapa canales

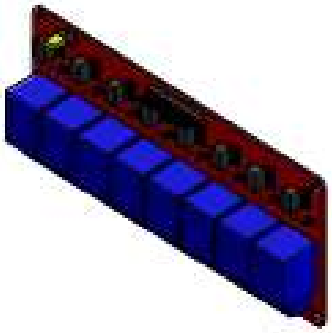
A4

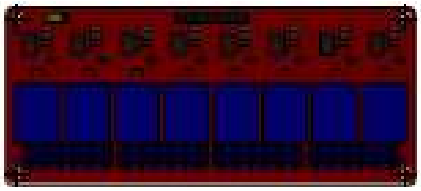
PESO:

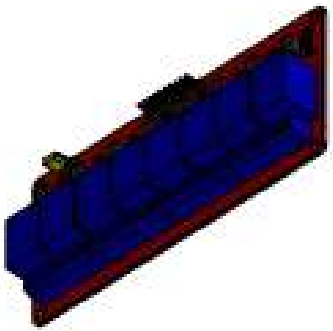
ESCALA 1:10

HOJA 1 DE 1

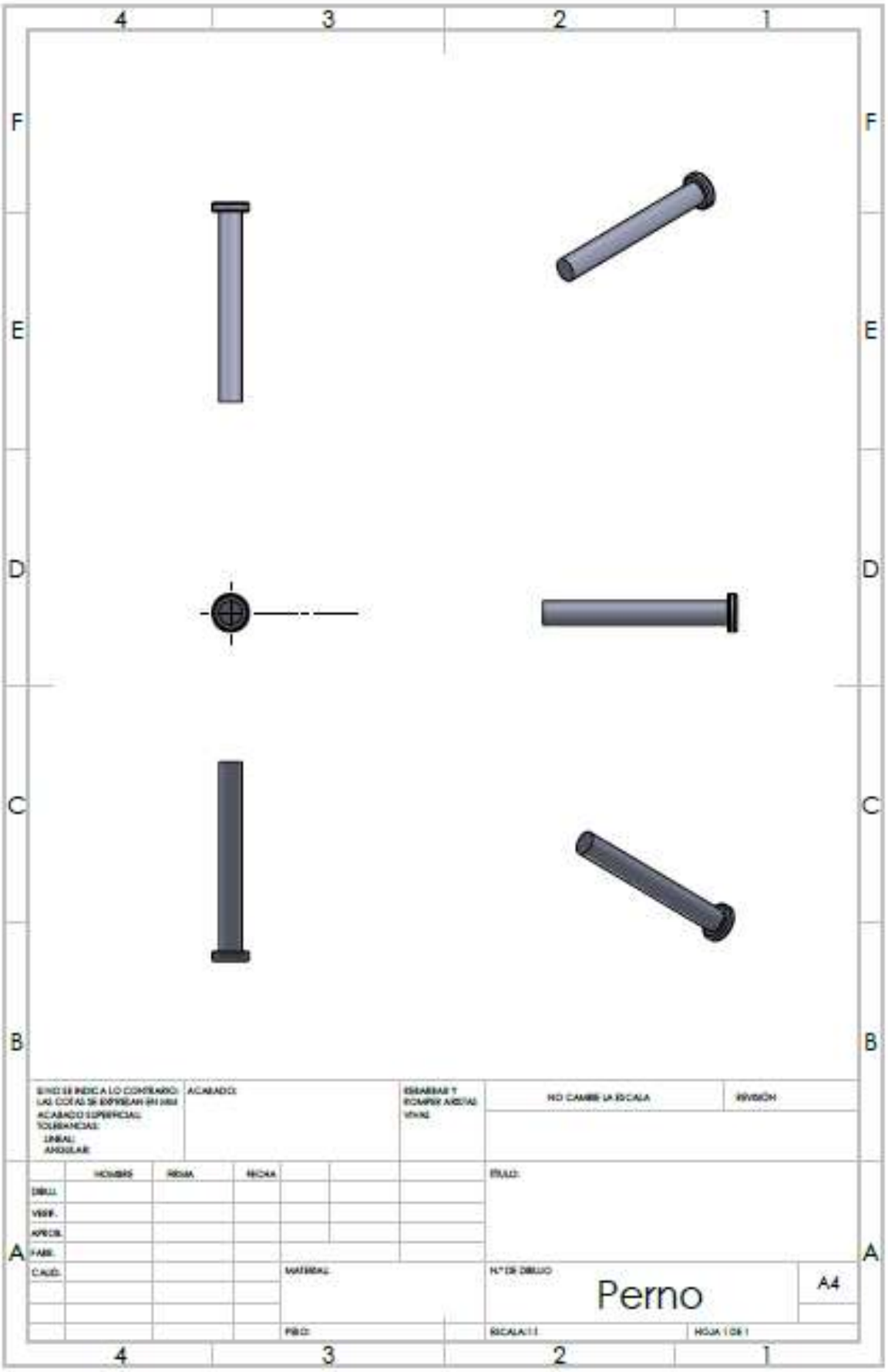
SI NO SE INDICA LO CONTRARIO, LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:		ESMERILAR Y ROMPER ARISTAS Vivos	NO CAMBIE LA ESCALA		REVISIÓN	
	NOMBRE	FIRMA	FECHA			TÍTULO:	
DIBUJ.							
VISTO							
APROB.							
FABR.							
CALID.				MATERIAL:		Nº DE DIBUJO: Bobina relé	A4
				PESO:		ESCALA:1:2	HOJA 1 DE 1

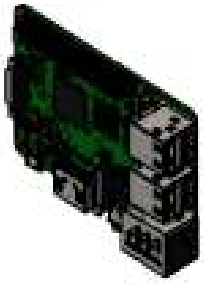


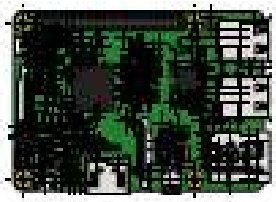


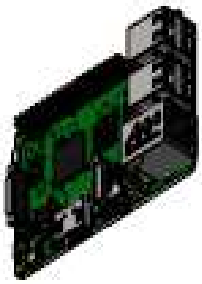


SI NO SE INDICA LO CONTRARIO LAS COTAS SE EXPRESAN EN mm ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:					ELIMINAR Y REMPRE ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
	NOMBRE	FIRMA	FECHA				TÍTULO:	
DIBUJ.								
VERIF.								
APROB.								
FABR.								
CALID.				MATERIAL			Nº DE DIBUJO	A4
							Relé	
				PESO:			ESCALA 1:5	HOJA 1 DE 1

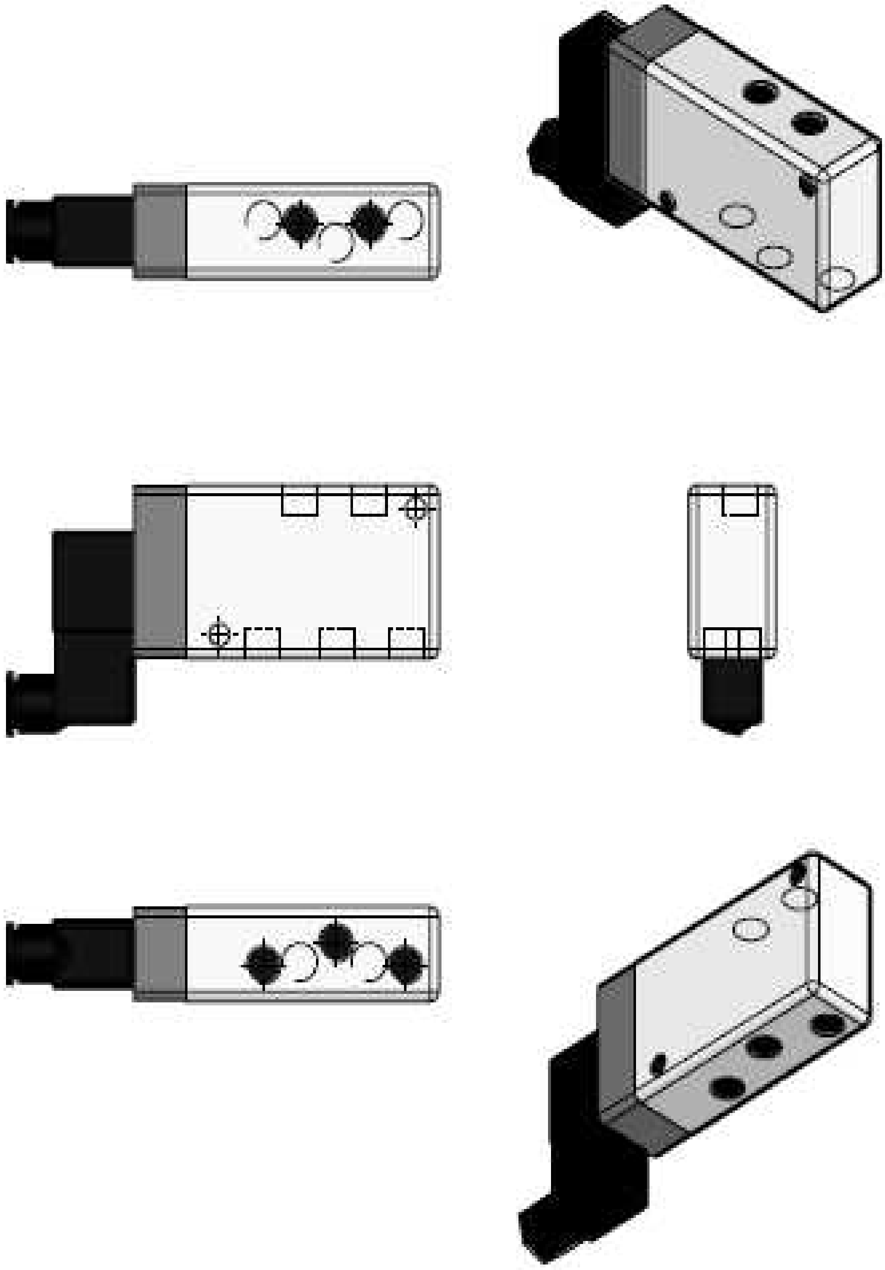

4
3
2
1
F
E
D
C
B
A
ACABADO:
NO CAMBIE LA ESCALA
REVISIÓN
DIBUJ.
VERIF.
APROB.
FABR.
CALID.
MATERIAL:
TÍTULO:
Perno
A4
ESCALA:1:1
HOJA 1 DE 1





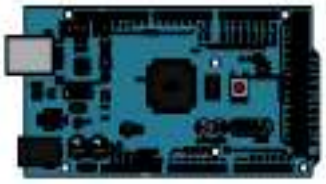


SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:					ESBARIAR Y ROMPER ARISTAS	NO CAMBIE LA ESCALA	REVISIÓN
	NOMBRE	FIRMA	FECHA				TÍTULO:	
DIBUJ.								
VISTO.								
APROB.								
FABR.								
CALID.				MATERIAL			Nº DE DIBUJO Raspberry	A4
				PESO:			ESCALA 1:2	HOJA 1 DE 1



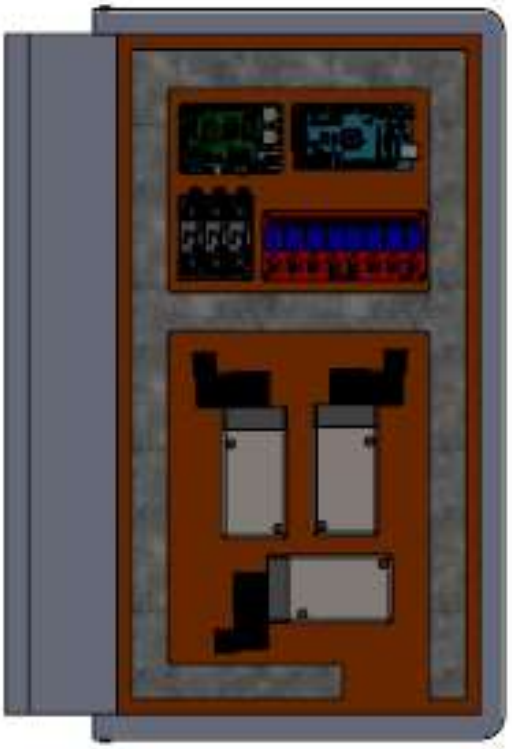
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN

	NOMBRE	FIRMA	FECHA		
DIBUJ.					TÍTULO:
VERIF.					
APROB.					
FABR.					
CALID.				MATERIAL:	N.º DE DIBUJO Válvula 5/2 — A4
				PESO:	ESCALA:1:2 — HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:	ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN

	NOMBRE	FIRMA	FECHA				TÍTULO:
DIBUJ.							
VISTP.							
APROB.							
F.NOR.							
C.NUM.				MATERIAL:			Nº DE DIBUJO: Arduino
							A4
				PESO:			ESCALA:1:3 HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:		ROMPER ARISTAS VIVAS:	NO CAMBIE LA ESCALA	REVISIÓN		
	NOMBRE	FIRMA	FECHA				TÍTULO:
DIBUJ.							
VISTO.							
APROB.							
FABR.							
CALID.				MATERIAL:		Nº DE DIBUJO	A4
						Ensamble	
				PESO:		ESCALA:1:10	HOJA 1 DE 1

Arduino Uno

Arduino Uno R3 Front

Arduino Uno R3 Back

Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front

Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.
The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.
The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Arduino MEGA 2560

CE

Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

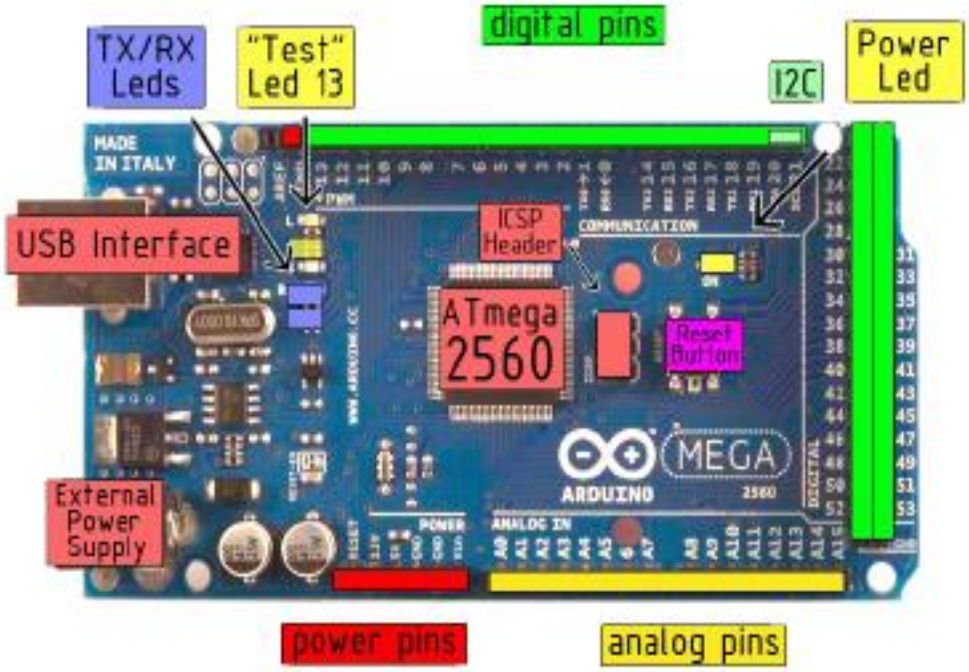
Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 18 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install Windows Install Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

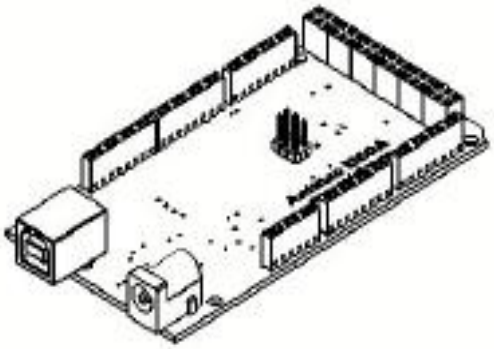
File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

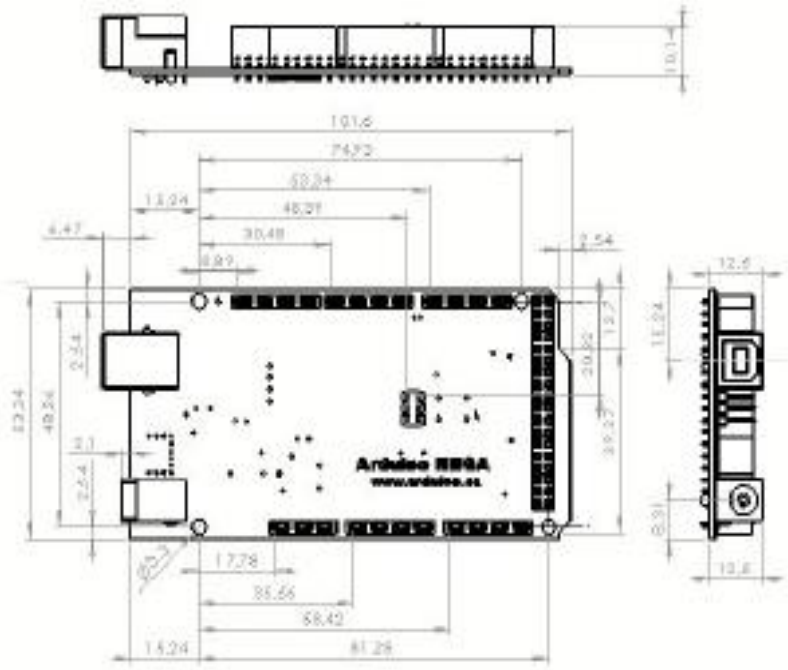
Once you have your skecth you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.

Dimensioned Drawing





radiospares RADIONICS

Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.It. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

IMPATTO ZERO® www.lifegate.it LIFEGATE

RS radiospares RADIONICS

Raspberry Pi

MODEL A+

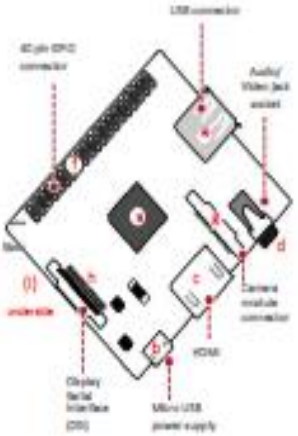
Product Name	**Raspberry Pi Model A+**
Product Description	The Raspberry Pi model A+ features lower power consumption, better audio performance and a 40-pin GPIO connector in an even smaller package.
RS Part Number	883-2699

Specifications

Chip	Broadcom BCM2835 (a)
Core architecture	ARM11
CPU	700 MHz Low Power ARM1176JZFS Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	512MB SDRAM
Operating System	Boots from micro SD card, running a version of the Linux operating system
Dimensions	66 x 56 x 14mm
Power	Micro USB socket 5V, 2A (b)

Connectors:

Digital AV Output	HDMI (rev 1.3 & 1.4) (c)
Analogue AV Output	3.5mm jack (d), Stereo audio, Composite video (PAL, NTSC)
USB	USB 2.0 Connector (e)
GPIO Connector	40-pin 0.1in header compatible with Model A/B 26-pin add-on boards (f)
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2) (g)
Display Connector	15-pin Display Serial Interface (DSI) (h)
Memory Card Slot	Micro SD (i)



Accessories

▲ Camera Module
913-2664

▲ Raspberry Pi power supply
822-6373

▲ 16GB microSD card with NOOBS -
121-3892

▲ Expansion board
772-2974

▲ WiFi dongle
760-3621

▲ 10400mAh Li-ion battery pack
775-7512

Raspberry Pi

Raspberry Pi 2, Model B

Product Name	**Raspberry Pi 2, Model B**
Product Description	The Raspberry Pi 2 delivers 6 times the processing capacity of previous models. This second generation Raspberry Pi has an upgraded Broadcom BCM2836 processor, which is a powerful ARM Cortex-A7 based quad-core processor that runs at 900MHz. The board also features an increase in memory capacity to 1Gbyte.

Specifications

Chip	Broadcom BCM2836 SoC
Core architecture	Quad-core ARM Cortex-A7
CPU	900 MHz
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V, 2A

Connectors:

Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4)
Audio Output	3.5mm jack, HDMI
USB	4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
JTAG	Not populated
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot	Micro SDIO

Raspberry Pi 3 Model B+

Overview

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT

The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

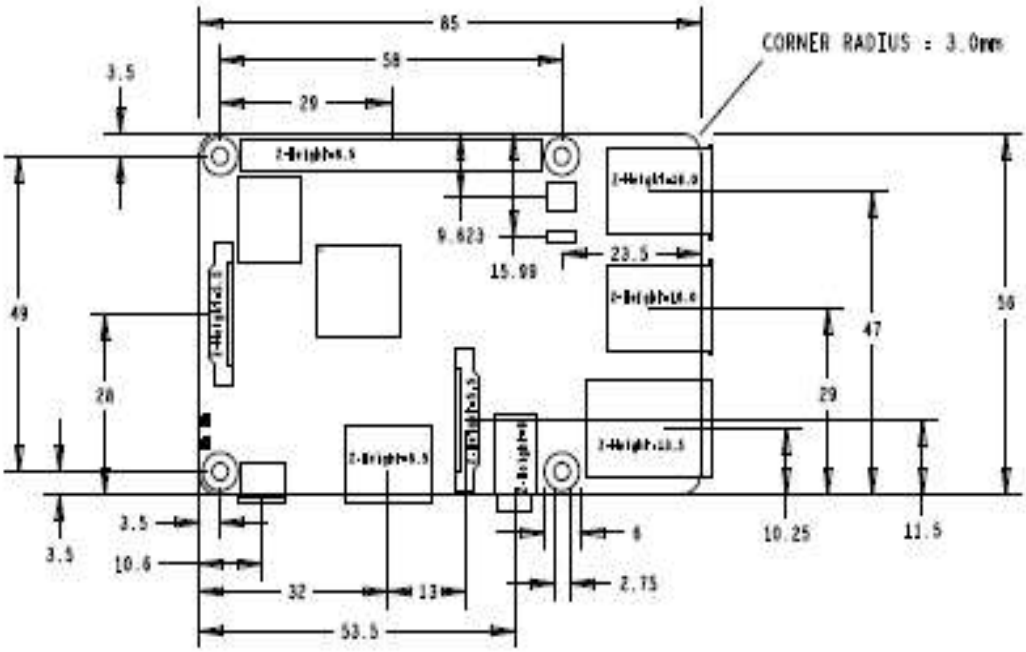
The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

Specifications

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE ■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) ■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	■ 1 × full size HDMI ■ MIPI DSI display port ■ MIPI CSI camera port ■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	■ 5V/2.5A DC via micro USB connector ■ 5V DC via GPIO header ■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.

raspberrypi.org

Physical specifications



Warnings

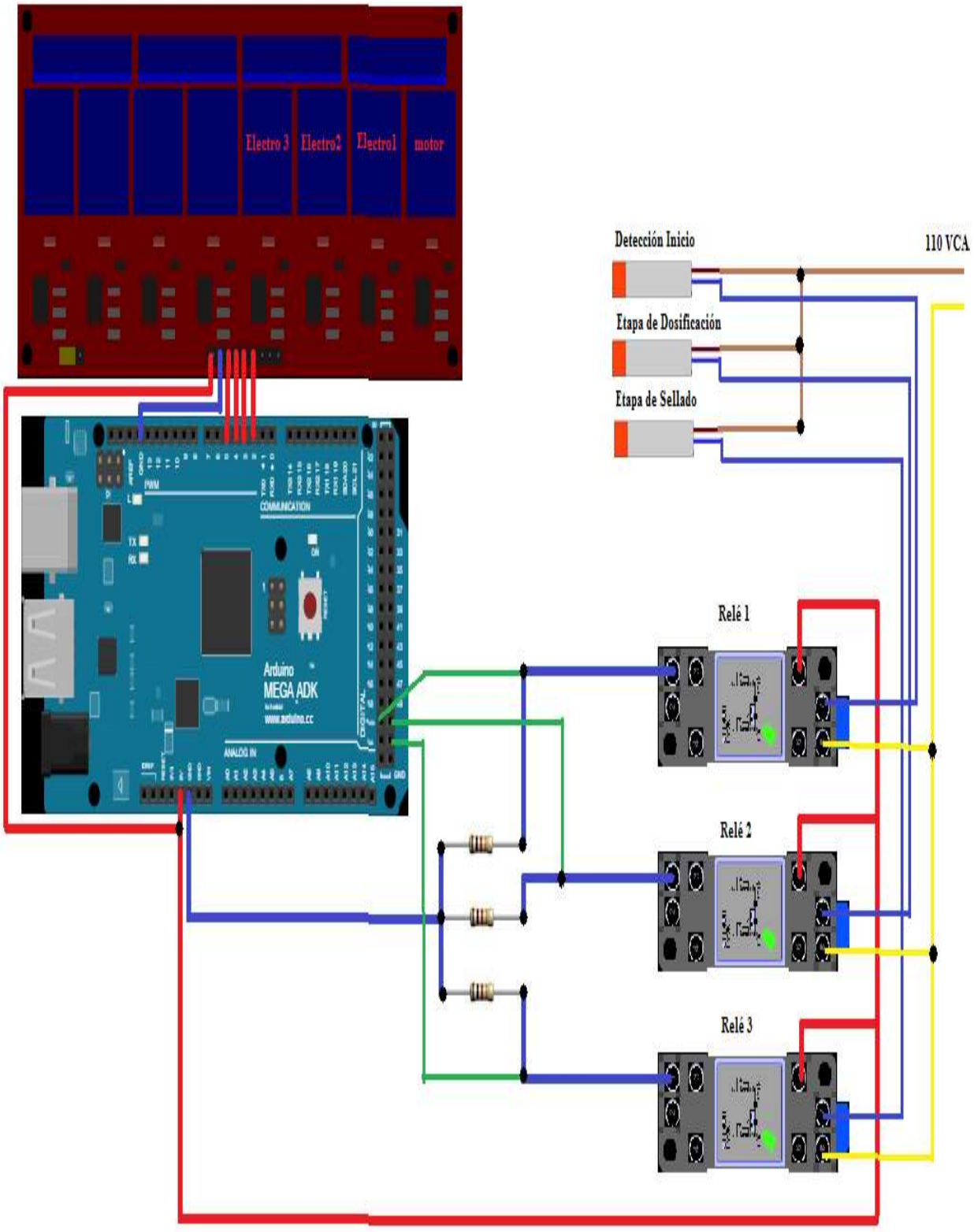
- This product should only be connected to an external power supply rated at 5V/2.5A DC. Any external power supply used with the Raspberry Pi 3 Model B+ shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- Whilst in use, this product should be placed on a stable, flat, non-conductive surface and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance, result in damage to the unit, and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors, and mice when used in conjunction with the Raspberry Pi.
- The cables and connectors of all peripherals used with this product must have adequate insulation so that relevant safety requirements are met.

Safety instructions

To avoid malfunction of or damage to this product, please observe the following:

- Do not expose to water or moisture, or place on a conductive surface whilst in operation.
- Do not expose to heat from any source; the Raspberry Pi 3 Model B+ is designed for reliable operation at normal ambient temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Whilst it is powered, avoid handling the printed circuit board, or only handle it by the edges to minimise the risk of electrostatic discharge damage.

Diagrama de conexiones sistema de control.



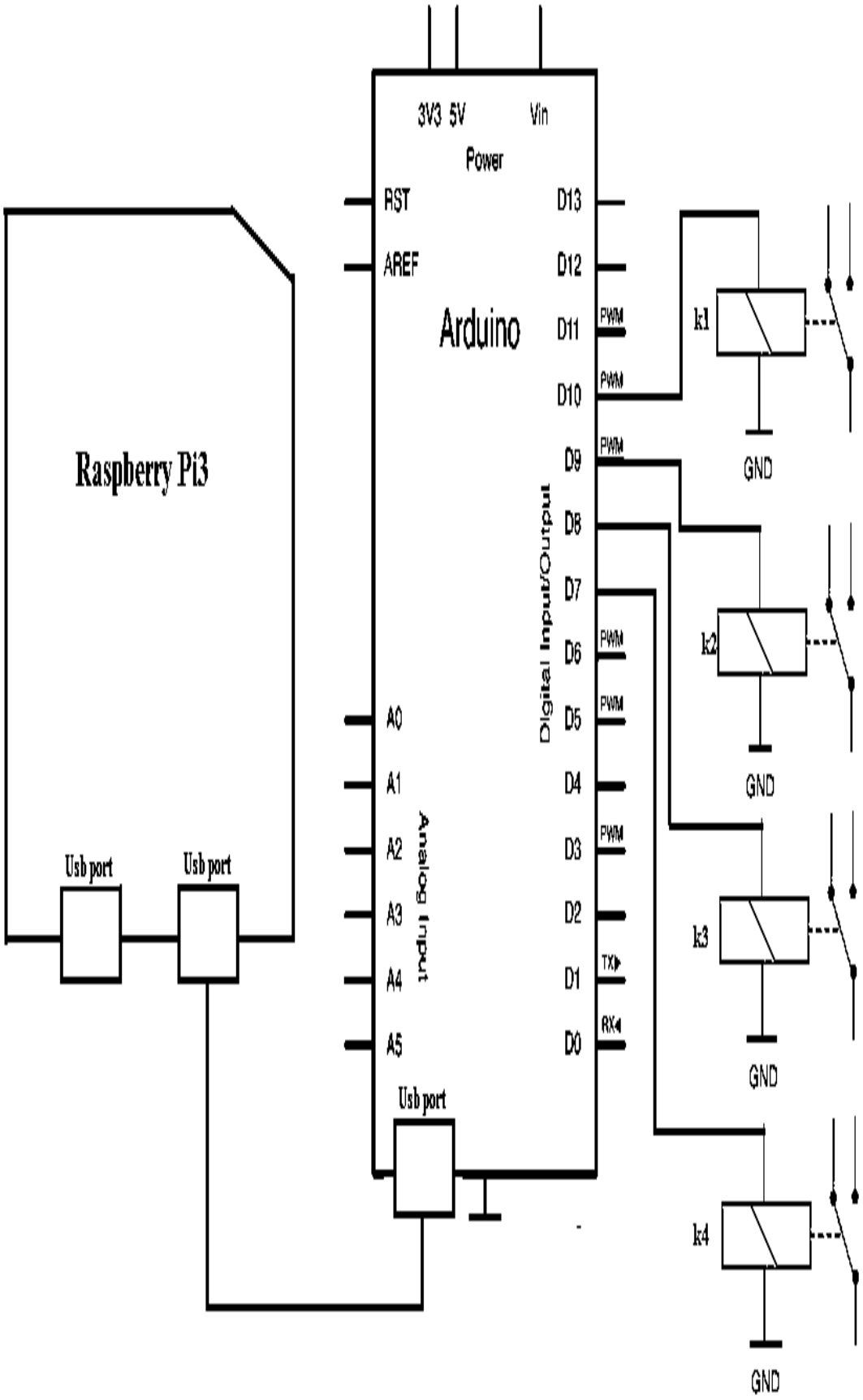


Diagrama electrónico de conexiones – salidas

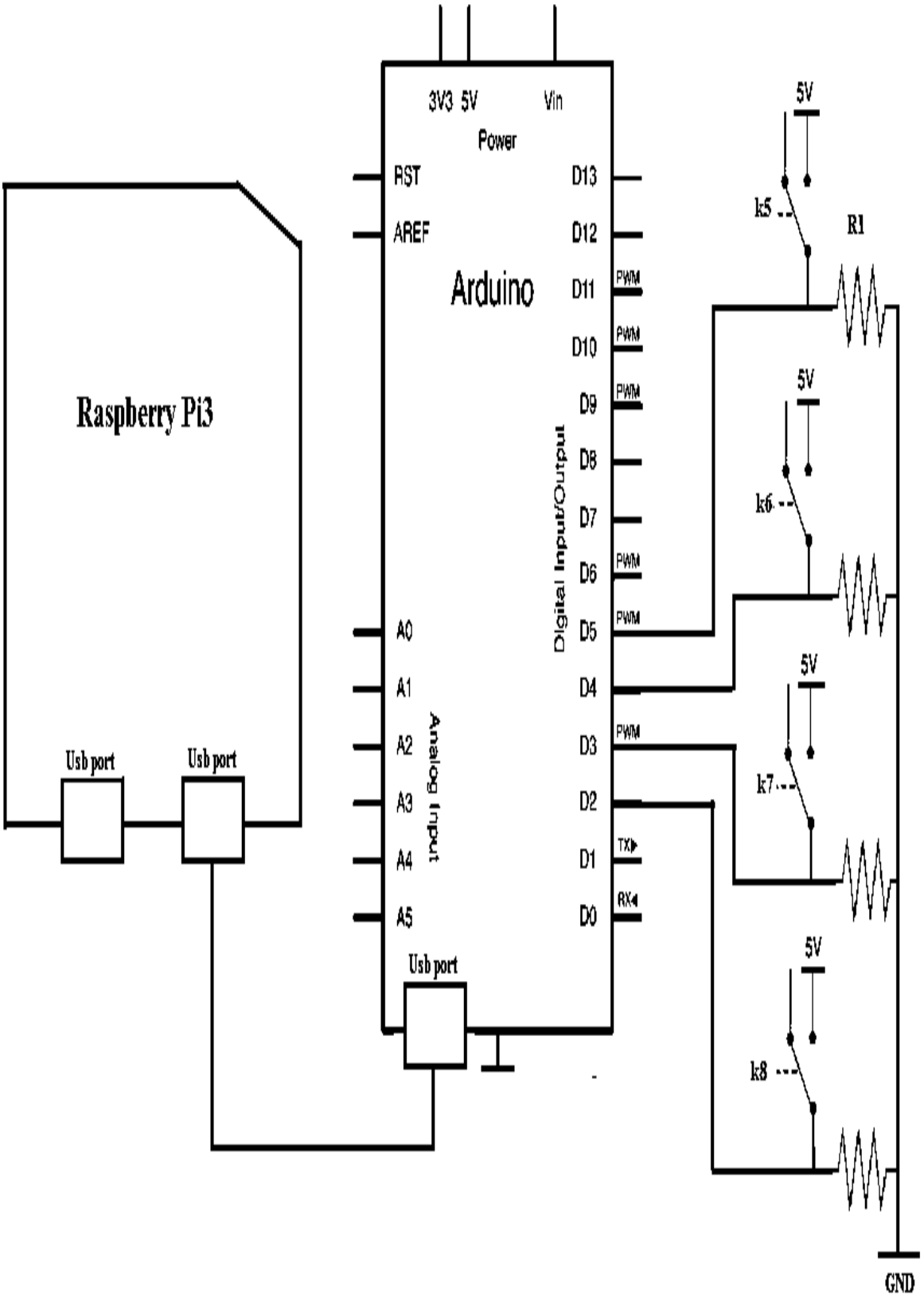


Diagrama electrónico de conexiones - entradas

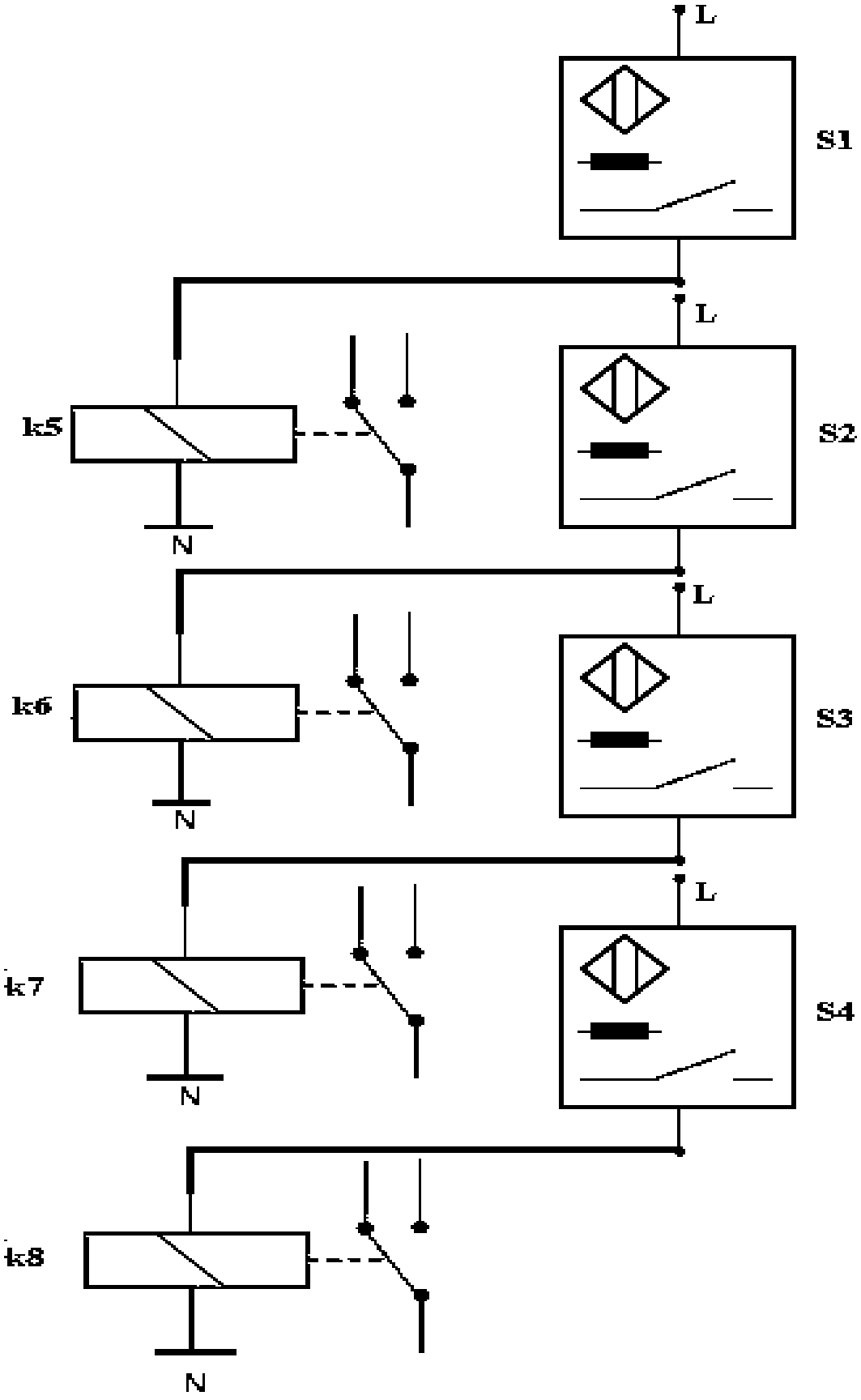


Diagrama eléctrico de conexiones - sensores

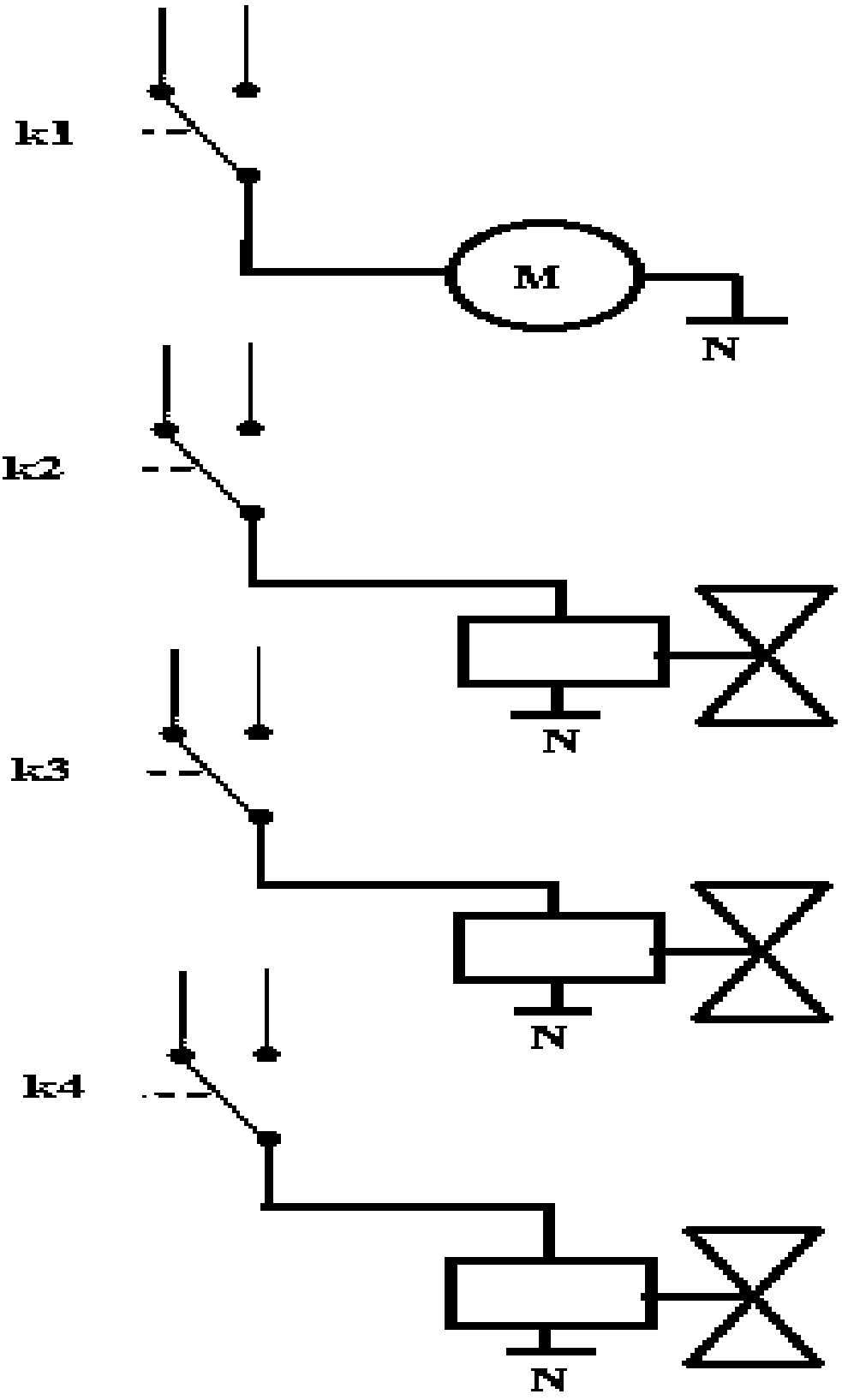


Diagrama eléctrico de conexiones - cargas

CÓDIGO DE ARDUINO.

```
const int RE1 = 2; //Motor
const int RE2 = 3; //Válvula Dosificadora
const int RE3 = 4; //Válvula Tapas
const int RE4 = 5; //Válvula Sellado
const int SE1 = 53; //Re1é de Detecta Tarro
const int SE2 = 52; //Relé de Dosificadora
const int SE3 = 51; //Relé de Tapado
int val = 0;
int val1 = 0;
int val2 = 0;
int cont = 0;
int conta = 0;
```

La función *void setup()* contiene la declaración del tipo de función que desarrollaran los pines de la placa en el circuito es decir si actuarán como entradas (INPUT) o salidas (OUTPUT), además de la inicialización de ciertos recursos como por ejemplo el puerto Serial.

```
void setup() {
        pinMode(RE1, OUTPUT);
        pinMode(RE2, OUTPUT);
        pinMode(RE3, OUTPUT);
        pinMode(RE4, OUTPUT);
        pinMode(SE1,INPUT);
        pinMode(SE2,INPUT);
        pinMode(SE3,INPUT);
        digitalWrite (RE1, LOW);
        digitalWrite (RE2, LOW);
        digitalWrite (RE3, LOW);
        digitalWrite (RE4, LOW);
        Serial.begin(9600); // Habilita el Puerto Serial.
        }
```

La función *void loop()* contiene todo la secuencia de instrucciones para el desarrollo del control del proceso, es un bucle de repetición.

```
void loop()
{
```

```
//Lectura de entradas digitales.
bool val = digitalRead(SE1);
bool val1 = digitalRead(SE2);
bool val2 = digitalRead(SE3);
if (val == HIGH){
  Serial.println('T');
  cont=cont+1;
  digitalWrite(RE1,HIGH);
  delay(1000);
}
 if (cont == 0){
  }
 else if (val1 == HIGH){
    Serial.println('D');
    cont=cont+1;
    digitalWrite(RE1,LOW);
    digitalWrite(RE2,HIGH);
    delay(5000);
    digitalWrite(RE1,HIGH);
    digitalWrite(RE2,LOW);
    Serial.println('F');
    cont=0;
    }
if (conta == 0){

  if (val2 == HIGH){
      Serial.println('S');
      conta=conta+1;
      digitalWrite(RE1,LOW);
      digitalWrite(RE2,LOW);
      digitalWrite(RE3,HIGH);
      delay(500);
      digitalWrite(RE3,LOW);
      delay(1000);
      digitalWrite(RE4,HIGH);
      delay(3000);
      digitalWrite(RE4,LOW);
```

```
            delay(500);
            Serial.println('H');
            digitalWrite(RE1,HIGH);
            delay(3000);
            digitalWrite(RE1,LOW);
            conta=0;
            Serial.println('L');
            }
    }
```

En el código, las líneas programadas evalúan las señales de los sensores (entradas) para gestionar el funcionamiento de los actuadores (salidas). Además, se puede observar en el código líneas específicas como *Serial.println('L')* que imprimen un carácter en el puerto serial el mismo que a posterior será empleado como señal de entrada para el sistema de monitoreo.