**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**
**SEDE CONCEPCIÓN - REY BALDUINO DE BÉLGICA**

# SISTEMA ROBÓTICO CLASIFICADOR DE OBJETOS CON VISIÓN ARTIFICIAL

**Trabajo de Titulación para optar al Título de Técnico Universitario en ROBÓTICA Y MECATRÓNICA**

**Alumno:**
**Guillermo Ignacio Méndez Moncada**

**Profesor Guía:**
**MSc. Felipe Benavides Pantoja**

**2 0 1 8**

## AGRADECIMIENTOS Y DEDICATORIAS

El presente proyecto lo dedico a todas aquellas personas que siempre me han apoyado, en buenos y malos momentos, especialmente mis padres, hermanos, tíos (as), profesores y amigos.

Los padres son claves en la formación de sus hijos, inculcando el respeto, la responsabilidad, honestidad, humildad, entre otros valores, que son fundamentales para poder enfrentarnos de buena forma a un mundo que cada vez tiende a ser más complejo. Así lo han sido ellos para mí, cuidándome, aconsejándome y siempre dándome ánimo para salir adelante. Con la mentalidad de que a base de esfuerzo y perseverancia se pueden conseguir los objetivos que nos proponemos, sin importar lo difícil que pueda ser el camino.

Mis hermanos son una motivación para continuar superándome cada día más, y así ayudar en su formación, para que ellos también puedan superarse a sí mismos para ser profesionales en un futuro no muy lejano.

Agradezco a aquellos amigos, que han estado en las buenas y en las malas apoyándome, sin esperar nada a cambio. Y también a profesores que me han marcado con su sabiduría, con su manera de ver el mundo y de enfrentar problemas o desafíos que se presentan, transformándose incluso en modelos a seguir.

Con este proyecto intento aplicar los conocimientos y las habilidades adquiridas durante el proceso de estudios, con el objetivo de superarme como persona y profesional.

Por Guillermo Méndez Moncada

**RESUMEN**

La robótica industrial puede definirse como el estudio, diseño y uso de robots para la ejecución de procesos industriales. Más formalmente, el estándar ISO (ISO 8373:1994,l. Robots industriales manipuladores – Vocabulario) define un robot industrial como un manipulador programable en tres o más ejes multipropósito, controlado automáticamente y reprogramable. [1]

La visión artificial es una disciplina científica que pretende emular la visión humana a la par que complementar a ésta. Comprende desde la captura de imágenes hasta la interpretación de ellas mediante un sistema artificial "inteligente". [2]

A través del presente proyecto se pretende realizar una investigación en el campo de la experimentación de la robótica industrial y de la visión artificial, adaptada a los ambientes de acceso remoto a través de las tecnologías de la información y la comunicación. En este orden se parte del análisis de los recursos tecnológicos y de infraestructura existente, sobre los cuales se busca implementar un sistema robótico clasificador de objetos con visión artificial. Desarrollando diferentes algoritmos eficientes que puedan ser implementados en una Raspberry Pi y su módulo de cámara, para que en conjunto y haciendo uso del software de programación Python y la biblioteca OpenCV para el procesamiento y análisis de imágenes, los dispositivos realicen la tarea de clasificación de objetos en función de la forma de estos. Además, el conjunto integrado por la Raspberry Pi Modelo 3B con el software desarrollado, debe establecer comunicación por Sockets con el Controlador AX20, para el envío/recepción de datos entre ambos controladores. Este último, procesar la información recibida y enviar las coordenadas de posicionamiento a un Brazo Robótico Nachi SC50F. De manera autónoma, este último se posiciona, realiza la sujeción y traslado de objetos sobre una banda transportadora, dentro de un área de operación específica. Dicho robot, finalizando la operación de clasificación, regresa a su posición inicial.

Lo ideal de este proyecto, es que contribuya en la formación en ingenierías haciendo uso de visión artificial, como también para el desarrollo de investigaciones en cuanto las prestaciones de los robots articulados en la industria de manufactura, en aplicaciones como la electromedicina y en entrenamiento. Sobre el sistema se puede poner en práctica las teorías y el conocimiento construido en el modelado y análisis cinemático y dinámico de brazos robóticos, aplicar diversos algoritmos y estrategias de control, así como el estudio para la planeación y seguimiento de trayectorias con fines específicos.

Se espera que las próximas generaciones de estudiantes conozcan estas herramientas y puedan sacar provecho de ellas para poder realizar futuros proyectos, para así continuar con la innovación en ámbitos tecnológicos y ser un aporte para la sociedad.

Palabras clave: ROBOTICA INDUSTRIAL, VISION ARTIFICIAL, RASPBERRY PI, CONTROLADOR AX20, BRAZO ROBOTICO NACHI SC50F, PYTHON, OPENCV, COMUNICACIÓN POR SOCKETS.

**ÍNDICE**

INTRODUCCIÓN ..... 1
PLANTEAMIENTO DEL PROBLEMA ..... 3
METODOLOGIA ..... 4
OBJETIVOS ..... 6
Objetivo General ..... 6
Objetivos Específicos ..... 6
1. CAPÍTULO 1: HARDWARE ..... 7
1.1 SISTEMA ROBÓTICO ..... 8
1.1.1 CONTROLADOR AX20-01 ..... 8
1.1.1.1 Funciones de control ..... 9
1.1.1.2 Software ..... 9
1.1.1.3 Modo enseñanza ..... 9
1.1.1.4 Aplicaciones ..... 9
1.1.1.5 Especificaciones básicas ..... 10
1.1.1.6 Funciones opcionales ..... 11
1.1.1.7 Funciones estándar ..... 12
1.1.2 BRAZO ROBÓTICO SC50F-01 ..... 14
1.1.1.8 Características del robot ..... 14
1.1.1.2 Aplicaciones ..... 14
1.1.1.3 Especificaciones básicas ..... 15
1.1.1.4 Rangos de operación ..... 16
1.1.3 PILOTO VERDE ..... 16
1.1.4 PILOTO ROJO ..... 17
1.1.5 SELECTOR ..... 17
1.1.6 VENTOSA ..... 18
1.1.7 EJECTOR ..... 18
1.2 SISTEMA DE VISIÓN ARTIFICIAL ..... 19
1.2.1 RASPBERRY PI 3B ..... 19
1.2.1.1 Características ..... 20
1.2.1.2 Conectores ..... 20
1.2.2 CÁMARA RASPBERRY PI ..... 21
1.2.3 CARGADOR RASPBERRY PI ..... 22
1.3 TABLERO DE CONTROL ELECTROPNEUMATICO ..... 23
1.3.1 COMPONENTES MECÁNICOS ..... 24
1.3.1.1 Gabinete ..... 24
1.3.1.2 Riel DIN ..... 24
1.3.1.3 Regleta ranurada ..... 25
1.3.2 COMPONENTES ELÉCTRICOS ..... 25
1.3.2.1 Automático ..... 25
1.3.2.2 Contactor ..... 26
1.3.2.3 Bornera ..... 26
1.3.2.4 Separador ..... 26
1.3.2.5 Tope ..... 27
1.3.3 COMPONENTES PNEUMÁTICOS/ELECTROPNEUMÁTICOS ..... 27
1.3.3.1 Filtro Regulador Lubricador ..... 27
1.3.3.2 Válvula Colizante ..... 28
1.3.3.3 Manifold ..... 29
1.3.3.4 Electroválvula ..... 29
1.4 CORREA TRANSPORTADORA ..... 29
1.4.1 COMPONENTES ..... 30

1.4.1.1 Motor Trifásico ..... 30
1.4.1.2 Sensor Fotoeléctrico O5P51A ..... 31
1.5 MESA DE OBJETOS CLASIFICADOS ..... 32
1.6 OBJETOS ..... 33
1.6.1 Objeto Cuadrado ..... 33
1.6.2 Objeto Cilíndrico ..... 34
2. CAPÍTULO 2: SOFTWARE ..... 1
2.1 RASPBIAN ..... 2
2.2 PYTHON ..... 3
3.2.1 CARACTERÍSTICAS Y PARADIGMAS ..... 4
1.3 OPENCV ..... 5
3.2.1 Aplicaciones conocidas ..... 5
1.4 FILEZILLA ..... 6
3. CAPÍTULO 3: PROGRAMACIÓN ..... 7
3.1 CONCEPTOS BASICOS QUE SE APLICAN AL ROBOT ..... 0
3.2.1 SISTEMA ROBÓTICO ..... 0
3.2 CONTROLADOR AX20-01 ..... 3
3.2.1 PROGRAMA BÁSICO – MODO MANUAL ..... 3
3.2.2 PROGRAMA BÁSICO – MODO AUTOMÁTICO ..... 5
3.3 RASPBERRY PI MODELO 3B ..... 6
3.3.1 INSTALACIÓN DE SISTEMA OPERATIVO ..... 6
3.3.2 INSTALAR OPENCV EN RASPBERRY PI ..... 8
3.3.3 CONEXIÓN DEL MÓDULO DE LA CÁMARA ..... 9
3.3.4 PROGRAMA INICIAL - VISTA PREVIA DE LA CÁMARA ..... 10
4. CAPÍTULO 4: EXPERIENCIAS Y RESULTADOS ..... 12
4.1 EXPERIENCIA 1 – ACTIVAR/DESACTIVAR SALIDAS ..... 14
4.2 EXPERIENCIA 2 – VISUALIZAR ESTADO DE ENTRADAS ..... 17
4.3 EXPERIENCIA 3 – PARTIR/PARAR CORREA TRANSPORTADORA ..... 19
4.4 EXPERIENCIA 4 – SISTEMA POSICIONADOR DE OBJETOS ..... 22
4.5 EXPERIENCIA 5 – USER TASK ..... 34
4.6 EXPERIENCIA 6 – SISTEMA DE VISIÓN ARTIFICIAL ..... 45
4.7 RESULTADOS FINALES ..... 50

## ÍNDICE DE FIGURAS

Figura 1- 1 Robot Industrial de Ensamblaje Automotriz KUKA .....2
Figura 1-2. Flujograma de la metodología llevada a cabo .....4
Figura 1-3. Controlador AX20 .....8
Figura 1-4. Manipulador Robótico Nachi SC50F .....14
Figura 1-5. Rangos de operación del robot SC50F .....16
Figura 1-6. Luz piloto verde.....16
Figura 1-7. Luz piloto rojo .....17
Figura 1-8. Selector .....17
Figura 1-9. Ventosa piab .....18
Figura 1-10. Ejector coax .....18
Figura 1-11. Raspberry Pi Modelo B .....19
Figura 1-12. Módulo cámara .....21
Figura 1-13. Cargador rasberry pi .....22
Figura 1-14. Tablero electropneumatico de control .....23
Figura 1-15. Gabinete.....24
Figura 1-16. Riel DIN .....24
Figura 1-17. Canaleta ranurada .....25
Figura 1-18. Automático 3P 10A .....25
Figura 1-19. Contactor 24V .....26
Figura 1-20. Bornera .....26
Figura 1-21. Separador.....26
Figura 1-22. Tope.....27
Figura 1-23. Filtro regulador lubricador .....27
Figura 1-24. Dimensiones FRL.....28
Figura 1-25. Válvula colizante.....28
Figura 1-26. Manifold .....29
Figura 1-27. Electroválvula.....29
Figura 1-28. Correa transportadora .....30
Figura 1-29. Motor trifásico.....30
Figura 1-30. Sensor fotoeléctrico.....31
Figura 1-31. Mesa de objetos clasificados .....32
Figura 1-32. Objeto cuadrado .....33
Figura 1-32. Objeto redondo .....34
Figura 2-1. Raspbian .....3
Figura 2-2. Python.....3
Figura 2-3. IDE de Python .....4
Figura 2-4. OpenCV .....5
Figura 2-5. Filezilla.....6
Figura 3-1. Diagrama de la configuración básica para el robot (Controlador AX20) .....0
Figura 3-2. Trabajo de enseñanza .....1
Figura 3-3. Operación automática.....1
Figura 3-4. Operando los ejes por separado (operación de eje independiente) .....2
Figura 3-5. Mover la punta del robot en línea recta (operación de coordenadas del robot) .....2
Figura 3-6. On/Off Controlador AX20 .....3
Figura 3-7. Pantalla de inicio .....3
Figura 3-8. Selección de programa .....4
Figura 3-9. Energizar servomotores.....4
Figura 3-10. Ajustar velocidad.....4
Figura 3-11. Interruptor de hombre muerto y control de ejes .....5
Figura 3-12. Grabación de posiciones.....5
Figura 3-13. Selección de número de programa .....6

Figura 3-14. Energizar servomotores ..... 6
Figura 3-15. Descarga de Raspbian ..... 7
Figura 3-16. Descargar ZIP ..... 7
Figura 3-17. SD Card Formatter ..... 7
Figura 3-18. Win32 Disk Imager ..... 8
Figura 3-19. Inicio raspberry pi ..... 8
Figura 3-20. Instalación Open CV en Raspberry Pi ..... 9
Figura 3-21. Configuración de raspberry pi ..... 9
Figura 3-22. Habilitar cámara ..... 10
Figura 3-23. Abrir python ..... 10
Figura 3-24. Primer programa ..... 11
Figura 4-1. Monitor 2 ..... 14
Figura 4-2. User OUTPUTS ..... 15
Figura 4-3. Activación de salidas ..... 15
Figura 4-4. Desactivación de salidas ..... 16
Figura 4-5. Monitor 2 ..... 17
Figura 4-6. User INPUTS ..... 18
Figura 4-7. Ejemplo activar/desactivar entrada 17 ..... 18
Figura 4-8. Creación de programa experiencia 3 ..... 19
Figura 4-9. Programa en modo automático ..... 20
Figura 4-10. Correa transportadora encendida ..... 20
Figura 4-11. Correa transportadora detenida ..... 21
Figura 4-12. Correa transportadora ..... 21
Figura 4-13. Creación de programa experiencia 4 ..... 22
Figura 4-14. Creación de programa experiencia 4 ..... 23
Figura 4-15. Creación de programa experiencia 4 ..... 23
Figura 4-16. Robot en posición inicial ..... 24
Figura 4-17. Objeto cuadrado posicionado ..... 24
Figura 4-18. Selector en pos 1 y pos 0 ..... 25
Figura 4-19. Robot posicionado sobre objeto cuadrado ..... 25
Figura 4-20. Robot posicionado en el objeto cuadrado ..... 26
Figura 4-21. Robot con objeto cuadrado en zona segura ..... 26
Figura 4-22. Robot con objeto cuadrado en zona segura mesa clasificadora ..... 27
Figura 4-23. Robot posiciona objeto cuadrado en mesa clasificadora ..... 27
Figura 4-24. Robot en zona segura mesa clasificadora objeto cuadrado ..... 28
Figura 4-25. Robot en posición inicial ..... 28
Figura 4-26. Objeto redondo posicionado ..... 29
Figura 4-27. Selector en pos 1 y pos 0 ..... 29
Figura 4-28. Robot posicionado sobre objeto redondo ..... 30
Figura 4-29. Robot posicionado en el objeto redondo ..... 31
Figura 4-30. Robot con objeto redondo en zona segura ..... 31
Figura 4-31. Robot con objeto redondo en zona segura mesa clasificadora ..... 32
Figura 4-32. Robot posiciona objeto redondo en mesa clasificadora ..... 32
Figura 4-33. Robot en zona segura mesa clasificadora objeto redondo ..... 33
Figura 4-34. Robot en posición inicial ..... 33
Figura 4-35. Cambiar nivel de protección R314 ..... 34
Figura 4-36. Seteo de nivel de usuario como Especialista ..... 35
Figura 4-37. Comunicación 8 ..... 35
Figura 4-38. Ethernet 2 ..... 36
Figura 4-39. TCP/IP 1 ..... 36
Figura 4-40. Configuración TCP/IP 1 ..... 37
Figura 4-41. Configuración FTP 3 ..... 37
Figura 4-42. Registro de usuario ..... 38
Figura 4-43. Programa USERTASK-A.123 ..... 38

Figura 4-44. Propiedades IPV4 computador personal .....39
Figura 4-45. Filezila conectado al controlador AX20.....39
Figura 4-46. Transferencia de programa.....40
Figura 4-47. Conversión de programa 9 .....40
Figura 4-48. Lenguaje 8 .....41
Figura 4-49. Selección de programa a convertir .....41
Figura 4-50. Conversión de programa finalizada.....42
Figura 4-51. Programa convertido .....42
Figura 4-52. User Task 37.....43
Figura 4-53. Selección del programa User Task.....43
Figura 4-54. User Task iniciado.....44
Figura 4-55. Declaración de librerias.....45
Figura 4-56. Configuración de sistema .....45
Figura 4-57. Capturar imagen .....45
Figura 4-58. Leer imagen.....46
Figura 4-59. Imagen en escala de grises .....46
Figura 4-60. Imagen binarizada .....46
Figura 4-61. Aplicación de filtros en imagen.....46
Figura 4-62. Encontrar contornos .....47
Figura 4-63. Obtener centroide .....47
Figura 4-64. Dibujar contornos y centroide en imagen.....47
Figura 4-65. Resultado de imagen capturada.....48
Figura 4-66. Resultado de imagen en escala de grises.....48
Figura 4-67. Resultado de imagen binaria .....49
Figura 4-68. Resultado de imagen con filtros .....49
Figura 4-69. Resultado Objeto Cuadrado 1 .....50
Figura 4-70. Resultado Objeto Cuadrado 2 .....50
Figura 4-71. Resultado Objeto Cuadrado 4 .....51
Figura 4-72. Resultado Objeto Cuadrado 5 .....51
Figura 4-73. Resultado Objeto Redondo 1.....52
Figura 4-74. Resultado Objeto Redondo 2.....52

## ÍNDICE DE TABLAS

Tabla 1-1. Especificaciones básicas ..... 11
Tabla 1-2. Funciones opcionales ..... 12
Tabla 1-3. Funciones estándar ..... 13
Tabla 1-4. Especificaciones básicas ..... 15
Tabla 1-5. Características piloto verde ..... 17
Tabla 1-6. Características piloto rojo ..... 17
Tabla 1-7. Características selector ..... 18
Tabla 1-8. Características ejector ..... 19
Tabla 1-9. Características Raspberry Pi 3 Modelo B ..... 20
Tabla 1-10. Conectores Raspberry Pi Modelo B ..... 20
Tabla 1-11. Características módulo cámara ..... 22
Tabla 1-12. Características cargador raspberry pi ..... 23
Tabla 1-13. Características gabinete ..... 24
Tabla 1-14. Características riel DIN ..... 25
Tabla 1-15. Características FRL ..... 28
Tabla 1-16. Características correa transportadora ..... 30
Tabla 1-17. Características motor trifásico ..... 31
Tabla 1-18. Características sensor fotoeléctrico ..... 32
Tabla 1-19. Características mesa clasificadora de objetos ..... 32
Tabla 1-20. Características objeto cuadrado ..... 33
Tabla 1-21. Características objeto redondo ..... 34
Tabla 3-1. Selección modo manual ..... 4
Tabla 4-1. Salidas experiencia 1 ..... 14
Tabla 4-2. Entradas experiencia 2 ..... 17
Tabla 4-3. Salida y entradas experiencia 3 ..... 19
Tabla 4-4. Salidas y entradas experiencia 4 ..... 22

# SIGLA Y SIMBOLOGÍA

## SIGLA

- ISO: Organización Internacional de Normalización
- GDL: Grado de libertad
- PLC: Controlador Lógico Programable
- RTOS: Sistema operativo de tiempo real
- HMI: Interfaz hombre maquina
- AC: Corriente alterna
- SLIM: Lenguaje estándar para manipuladores industriales
- IP: Protocolo de Internet
- CSI: Interfaz Serie para Cámaras
- IP: Índice de protección

## SIMBOLOGÍA

- [mA] : MiliAmpere
- [mW] : MiliWatt
- [V] : Volt
- [KVA]: Kilo Volt Amper
- [Km] : Kilómetro
- [m] : Metro
- [mm]: Milímetros
- [s] : Segundo
- [Ω]: Ohm
- [fps]: frames por segundo
- [px]: pixel
- [rpm]: revolución por minuto

# INTRODUCCIÓN

Para dar inicio a este proyecto, es necesario una leve descripción a la siguiente interrogante: ¿Qué es la robótica industrial? "El campo de la robótica industrial puede definirse como el estudio, diseño y uso de robots para la ejecución de procesos industriales.

Los brazos robóticos están diseñados para realizar tareas simples o complejas, con una gran precisión, velocidad y varios otros factores que los hacen mucho más eficientes en comparación al ser humano. El uso de estos robots es relativamente sencillo a nivel de usuario, pero a nivel técnico resulta ser más complejo, sobre todo cuando se exploran nuevos proyectos a ser realizados, en ámbitos de control de calidad o clasificación de objetos, aplicaciones que con el tiempo han ido tomando fuerza, por el uso de nuevos dispositivos tecnológicos (controladores, cámaras, entre otros) que son incorporados en un proceso para trabajar en conjunto con los robots mencionados anteriormente.

Tradicionalmente el hombre ha podido utilizar únicamente sus propios ojos y su cerebro para ver el mundo que le rodea y procesar toda la información que recibe a través de ellos. Sin embargo, las nuevas tecnologías han permitido el desarrollo de la visión artificial, o visión por computador, definida como la ciencia cuyo objetivo es dotar de esa misma capacidad que poseen los humanos a las maquinas u ordenadores. Esta disciplina está basada en la extracción automática, el análisis y la comprensión de información útil que pueda ser proporcionada por una imagen única o una secuencia de imágenes, pero requiere del desarrollo de una base teórica y algorítmica para conseguir la aprehensión visual automática. [3]

Las aplicaciones de visión por computador son numerosas y comprenden ámbitos tan diversos como pueden ser la agricultura, controles de calidad en industria, realidad aumentada, análisis de imágenes médicas, seguridad y vigilancia, transporte, robótica y un gran número de disciplinas más.

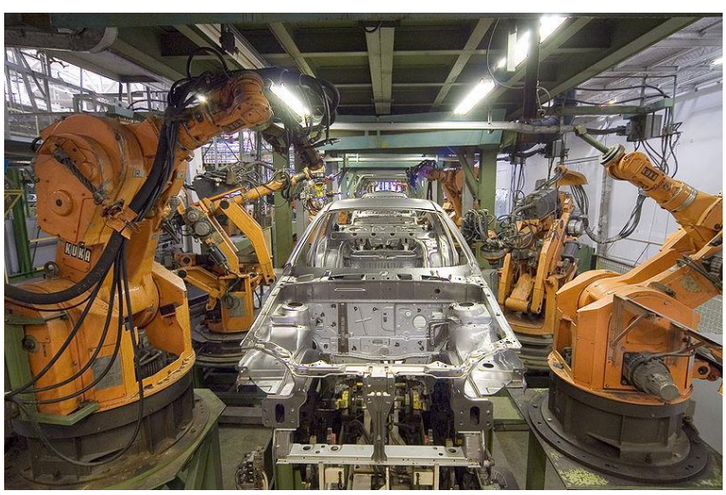
Todo esto ha motivado el desarrollo de este trabajo de título, basado todo él en la robótica industrial y en la visión por computador citada. Un gran número de robots requieren este tipo de característica, tanto para ofrecer una determinada funcionalidad como por seguridad. Los robots precisan de medidas de seguridad especiales para evitar posibles daños a los humanos, con los que están en contacto directo con sus actividades. Estas medidas de seguridad pueden ser implementadas de muchas formas, y una de ellas es la visión artificial, que permite detectar personas, objetos, esquinas y entre otras cosas poder prevenir colisiones. Por tanto, la seguridad es una funcionalidad ineludible del desarrollo de este trabajo de título.

Dicho lo anterior, en este proyecto se pretende indagar en mayor profundidad la visión artificial en manipuladores robóticos para uso industrial, por ello inicialmente es necesario saber ¿en qué consiste la visión artificial en la robótica industrial? consiste en, utilizando cámaras, conseguir que un robot no sólo emule el brazo de un operario, sino también el ojo". De esta forma se puede solventar el problema que surge con algunas industrias que trabajan a la vez en diversas líneas de producto, donde el uso del robot tradicional para la automatización de procesos no es válido.

"El robot debe disponer de diferentes G.D.L y de una capacidad sensorial que le permita posicionarse frente al objeto", asegura, bien por la alta variabilidad de los productos a los que se enfrenta o bien por encontrarse éstos en movimiento o en posiciones aleatorias. Los equipos de visión proporcionan al robot en el instante de aprehensión las coordenadas y orientación del producto, lo que posibilita "una manipulación precisa y fiable". "En líneas en las que el producto va en movimiento o hay varios mezclados el mismo robot puede diferenciar dónde tiene que dirigirse para capturar el producto y procesarle", incide el experto.

Los sistemas de visión artificial parten siempre de una representación digital. "Una cámara actúa como un ojo, el iris corresponde al diafragma, la retina al sensor y el cristalino a la lente, que permite alejar o acercar la cámara del objeto para conseguir un buen enfoque", ha precisado. Frente a la visión humana, la artificial depende mucho de las condiciones lumínicas, puesto que "no puede adaptarse a los cambios como el ojo humano". Por ello, su mayor uso se encuentra en el campo de la industria, donde puede determinarse una iluminación estable.

A continuación, en la Figura 1-1 se presenta un robot industrial, cuya aplicación es el ensamblaje de piezas automotrices:




1 Figura 1- 1 Robot Industrial de Ensamblaje Automotriz KUKA

## PLANTEAMIENTO DEL PROBLEMA

La mecatrónica es una disciplina integradora de las áreas de mecánica, electrónica e informática cuyo objetivo es proporcionar mejores productos, procesos y sistemas industriales. La mecatrónica no es, por tanto, una nueva rama de la ingeniería, sino un concepto recientemente desarrollado que enfatiza la necesidad de integración y de una interacción intensiva entre diferentes áreas de la ingeniería. Esta disciplina tal vez se ha demorado más de la cuenta en darse a conocer masivamente en Chile. En el ámbito industrial, los principales desarrollos se han efectuado en el área de la gran minería del cobre.

Para el desarrollo constante y el aumento de productividad, las industrias utilizan manipuladores robóticos en sus procesos. Y para obtener rendimientos aún mayores de estos robots y garantizar una mejor calidad en los productos y/o servicios que ofrecen estas industrias, se incorporan a los robots sistemas de visión artificial. Sistemas que son de excelente calidad, pero que son limitados por el fabricante y comercialmente son de un costo muy elevado.

Para solucionar el problema de limitaciones y de los costos elevados, este proyecto pretende implementar un nuevo sistema de visión artificial, haciendo uso de actuales tecnologías, considerando una excelente calidad y un bajo costo en el mercado, se decide utilizar una Raspberry Pi 3 Modelo B con su respectiva cámara y así hacer que estos elementos trabajen en conjunto, conformando el sistema de visión artificial de un brazo robótico Nachi SC50F-01 y su Controlador AX20-1101, obteniendo como resultado un proceso industrial, que cumpla con una de las grandes necesidades de las industrias, como lo es la precisión para la selección, clasificación, posicionamiento, traslado, estampado de productos, etc.

## METODOLOGIA

Las etapas que integran esta metodología y que se han llevado a cabo son las que se definen en la Figura 1-2:

2 Figura 1-2. Flujograma de la metodología llevada a cabo

**Planificación inicial:** Es la primera etapa del proyecto y en la que se realiza la planificación global y la división del desarrollo en pequeñas funcionalidades diferenciadas. Se definen las fechas límite, la dedicación y horas invertir, y criterios de evaluación y seguimiento para verificar el cumplimiento de los objetivos.

**Planificación intermedia:** En el desarrollo del proyecto se establece una planificación individual personalizada para cada fragmento. Se definen los mismos elementos que en la planificación inicial, pero ajustados a este "proyecto".

**Diseño:** Se trata de una etapa principalmente teórica. Consiste en la búsqueda de información, análisis de las diferentes formas de llevar a cabo el trabajo, diseño del hardware y del software a desarrollar, en forma bosquejos para el caso del hardware y diagramas para el caso del software, que sirvan como guía, etc.

**Programación:** Es la etapa que conlleva más tiempo ya que comprende la escritura del programa, siguiendo los requisitos de simplicidad, eficiencia, cumpliendo los objetivos establecidos de forma óptima. Además, se incluye en esta etapa la refactorización del

código. La refactorización es una técnica integrada en la Ingeniería de Software que consiste en reestructurar un código de un programa internamente pero sin alterar el comportamiento externo. Es decir, se trata de una “limpieza de código”, eliminando líneas innecesarias, variables que no se usen, comentando las funcionalidades para que puedan ser entendidas por alguien que no ha desarrollado el código, introduciendo sangrías que permitan una lectura más clara, etc.

**Testing:** Dentro del testing y la validación de los resultados existen dos acciones. La primera acción es el testing unitario, que se lleva a cabo junto con la etapa de programación asegurando que el código no contenga errores. Esto permite desarrollar un código cuya sintaxis sea correcta. La segunda acción comprende el test de aceptación o validación de los resultados, que consiste en verificar si el programa desarrollado cumple los objetivos que se definieron en la planificación. Si los resultados no son los esperados es necesario repetir el ciclo, realizando de nuevo la planificación intermedia. Si, por el contrario, los resultados cumplen los objetivos definidos, se publica esa funcionalidad que estará lista para ser implementada posteriormente en el programa final.

# OBJETIVOS

## Objetivo General

- Implementar un sistema robótico clasificador de objetos con visión artificial, para la detección, sujeción y traslado de objetos cuadrados y redondos posicionados sobre una banda transportadora. Clasificándolos de acuerdo a su forma sobre una superficie determinada.

## Objetivos Específicos

- Elaborar programas de movimiento para el control de un brazo robótico en modo manual y automático.
- Incorporar sensores y actuadores al controlador AX20.
- Activar actuadores y visualizar estado de sensores desde el teach pendant.
- Elaborar programas de movimiento incorporando sensores y actuadores.
- Crear programas robot con un editor de texto de un pc.
- Establecer comunicación Ethernet entre un pc y el controlador AX20.
- Transferir programas robot desde un pc al controlador AX20 y viceversa.
- Compilar y convertir programas robot.
- Desarrollar un sistema de visión artificial que permita identificar y obtener la posición objetos cuadrados y redondos.

# 1. CAPÍTULO 1: HARDWARE

En el presente capítulo se detallan los componentes utilizados en el proceso de elaboración del proyecto.

Estos componentes son esenciales para que el sistema mecatrónico funcione como es debido, considerando aquellos del área mecánica, neumática, electrónica/eléctrica e informática.

## 1.1 SISTEMA ROBÓTICO

### 1.1.1 CONTROLADOR AX20-01

El controlador AX20 es un controlador de robot multi-tarea de alto rendimiento de Nachi, que presenta muchas mejoras de hardware y software que brindan un mayor rendimiento, capacidad de movimiento cooperativo, tiempos de ciclo más cortos y una programación más rápida y flexible, basada en menús. Equipado con el PLC de software incorporado, que es una función estándar. El PLC de software se puede usar para controlar una celda de trabajo completa o realizar tareas más simples. El AX20 cumple con el protocolo de circuito de seguridad estándar internacional por seguridad con la que puede contar. El equipo seguro, como el circuito de parada de emergencia doble y el interruptor de habilitación de 3 posiciones, se incluye de serie.

3 Figura 1-3. Controlador AX20

#### 1.1.1.1 Funciones de control

- Movimiento de paso corto de 50mm es un 30% más rápido.
- Circuito de seguridad estándar internacional.
- Detección de interferencia de alta velocidad.
- Unidad totalmente digital con control de movimiento DSP de alta velocidad.
- Mantenimiento simplificado.
- Se extiende al bus estándar de la industria (ISA / PCI)
- Crea fácilmente copias de seguridad completas del sistema para tarjetas multimedia compactas.

#### 1.1.1.2 Software

- Sistema operativo en tiempo real (RTOS) más WindowsNT® integrado.
- Mejora del movimiento del robot (mayor velocidad, mejor planificación del movimiento).
- Herramientas sencillas para desarrollar y validar programas de robots.
- Software PLC controla directamente periféricos.
- Programación fuera de línea.
- Mantenimiento visual con manual en línea.

#### 1.1.1.3 Modo enseñanza

- Fácil de operar
- Diseño ergonómico.
- Pantalla grafica a color mejorada de Windows.
- Pantalla del panel táctil.
- Los botones e indicadores se pueden personalizar
- HMI del panel de interfaz personalizable con hasta 248 teclas

#### 1.1.1.4 Aplicaciones

- Utilizado en todos los productos de robot Nachi

1.1.1.5 Especificaciones básicas

| Ítem | Especificaciones |
|---|---|
| Eje controlado | Simultáneos 7 ejes (máximo 9 ejes opcionales)<br>Total de 18 ejes (gabinete de control extendido) |
| Servo motor/ Dispositivo de posicionamiento | Servomotor AC/ Encoder absoluto |
| Lenguajes de programación | Enseñanza de la reproducción (método de paso estándar/ registro de función)<br>Lenguaje JIS SLIM (opcional)<br>Software PLC IEC1131 (Nachi Soft-PLC) |
| Numero de programas/ Capacidad de memoria | 9.999 programas/ 16MB (160,000 pasos equivalentes) |
| Cables del Robot | Longitud de cable estándar de 5 m (para montaje en piso o montado en estante) |
| (desde el controlador al manipulador) | (10m, 15m 20m y 25m cables son opcionales) |
| Memoria Interna/ Externa | Memoria Flash/ Interfaz de tarjeta Compact Flash |
| Interfaz de usuario | Interfaz del panel PB en la parte frontal del controlador,<br>montaje adicional dentro de la puerta y panel lateral |
| Construcción/ Clasificación de IP | Caja cerrada con clasificación IP54 |
| Sistema de refrigeración | Sistema de enfriamiento directo |
| Voltaje de entrada de alimentación | Sin transformador: AC200V - 220V ± 10% (3 fases, 50 / 60Hz)<br>Con transformador: AC380V ~ 480V ± 10% (3 fases, 50 / 60Hz) |
| Temperatura ambiental/ humedad | 0-45 ° C / 20-85% (sin condensación) |
| Consumo de energía | Potencia máxima durante la reproducción del programa de operación:<br>Serie SC50: Aprox. 3kVA |
| Dimensiones del gabinete del controlador | W450 x H782 x D550mm (sin ruedas de control); Altura: 882 (con ruedas controladoras)<br>W450 x H1195 x D550mm (without controller casters); Height: 1295 (with controller casters) |
| Peso | Aprox. 85 kg sin transformador<br>Aprox. 165 kg con transformador |
| Color recubierto | Munsell 10GY9/1 |
| Función protectora del operador | Enseñar el enclavamiento del modelo / modo de reproducción |

| | | |
|---|---|---|
| | | Interruptor de hombre muerto<br>Botón de parada de emergencia (Panel de operación / Teach colgante / Entrada de señal externa)<br>Señal de enclavamiento de la puerta de la valla de protección (SFP - Safety Plug)<br>Habilitar interfaz de conmutación |
| Función de autodiagnóstico | | Se usa para autodiagnosticar errores del robot y el controlador<br>(disponible para aproximadamente 700 tipos de errores) |
| Función de detección de error | | Se usa para controlar el estado del robot y el controlador en todo momento<br>Se usa para hacer que el robot pare inmediatamente cuando ocurre un error |
| Teach pendant | Pantalla | Pantalla TFT LCD a color de 6.5 pulgadas (pantalla de 256 colores de 640 x 480 con luz de fondo) |
| | Peso TP | Aprox. peso 1,3 kg (sin cables de conexión) |
| | Longitud del cable | Longitud de cable TP estándar 8 m (cables de 15 m, 20 m, 25 m y 30 m son opcionales) |
| | Interruptor de hombre muerto | El interruptor Deadman de tres posiciones y de una sola mano es estándar (opcional para dos manos) |
| | Pantalla táctil | Pantalla táctil digital (opcional) |
| | Ayuda de idioma | El soporte para dos idiomas concurrentes se puede seleccionar de los idiomas que se muestran (todos<br>los idiomas excepto el inglés yel japonés son opcionales): japonés, inglés, alemán, coreano, taiwanés<br>(chino tradicional), portugués, español, francés, chino (chino simplificado), italiano y holandés) |

1 Tabla 1-1. Especificaciones básicas

### 1.1.1.6 Funciones opcionales

| Funciones opcionales | Descripción |
|---|---|
| Tensión de alimentación primaria | Para una tensión de alimentación primaria fuera de la norma (AC200-220V) |
| Interfaz de tarjeta Compact Flash | La ranura compacta de la tarjeta flash está instalada |
| Eje Auxiliar Servo Pistola | Se controla la pistola de soldadura por puntos: especificación de pistola (cantidad de<br>de la aplicación individual (puede ser necesario un amplificador adicional) |

| | |
|---|---|
| Deslice el posicionador en otro eje auxiliar | El posicionador deslizante está controlado por servo (puede ser necesario un amplificador adicional) |
| Posicionador de eje | El posicionador está controlado por servo (puede ser necesario un amplificador adicional) |
| Field bus | Device Net, Profibus-DP, Interlink, RIO, CC-LINK, un máximo de 2 canales de comunicación por controlador.<br>(Ej. 1 Maestro y 1 Esclavo, 2 Maestros o 2 Esclavos) |
| Input / Output | Entrada 32 DC24V (resistencia de entrada 3k, corriente de entrada 3mA o mayor, entrada de estilo de hundimiento)<br>Salida 32 DC24V (voltaje de salida DC24V, +/- 3V a 100 mA) |
| Extender DC I/O | 32/32 o 64/64 adicionales para un total de 64/64 o 96/96 I / O) |
| XYZ Shift | El punto grabado se reproduce con la cantidad de desplazamiento paralelo XYZ |
| Paletizado | Enseñanza de paletizado y despaletizado programado por definición de patrón amigable integrada |
| Lenguaje Robot | Lenguaje JIS SLIM |
| PLC incorporado | Herramienta de software de programación fuera de línea (ICS Triplx ISaGRAF Workbench) |

2 Tabla 1-2. Funciones opcionales

#### 1.1.1.7 Funciones estándar

| Funciones estándar | Detalle |
|---|---|
| Exactitud | Se puede designar una selección de 8 (0-1000 mm) de precisión en posición en cada paso.<br>En posición o camino a través también puede ser designado |
| Designación de herramienta | Se pueden definir 32 herramientas únicas para usar en posiciones enseñadas. |
| Calculadora automática de longitud de herramienta | Cálculo automatizado de la longitud de la herramienta seleccionando el programa enseñado |
| Peso de herramienta automático y calculadora COG | El peso automático de la herramienta y el COG se calculan ejecutando el programa designado |
| Herramienta automática Momento de la calculadora de inercia | El momento automático de inercia de la herramienta se calcula ejecutando un programa designado |
| Detección de errores de auto verificación | Autocomprobación interna continua en tiempo real del robot y el controlador para detectar fallas<br>o errores (700 tipos de errores). El robot se detiene inmediatamente cuando se detecta un error. |

| | |
|---|---|
| Logical I/O | Máximo 2,048 puntos de E / S lógicos para usar con Soft-PLC |
| Editor de programación | 1. Screen Editor – addition, deletion and copy of every move step and function is available;<br>recorded position can also be edited<br>2. Utilidad de copia: el programa grabado y el paso se pueden copiar<br>3. Conversión de programa: condición y velocidad, ángulo de cada eje, desplazamiento paralelo, etc.<br>4. Certificación del programa - directorio de archivos, verificación de archivos<br>5. Deshacer la función para eliminar la última operación de programación pendiente |
| Función de ahorro de energía | Ahorra energía al bloquear los frenos y eliminar la potencia del motor después de que haya transcurrido un tiempo<br>predeterminado sin movimiento. Después de un tiempo prolongado sin movimiento, los ventiladores de refrigeración<br>del controlador se detendrán para ahorrar aún más energía. |
| Mensaje de ayuda (manual incorporado) | Las explicaciones de funcionamiento y funciones se muestran en la pantalla de instrucciones<br>junto con un manual de solución de problemas de pantalla gráfica. |
| Posiciones iniciales y salidas | Se pueden definir hasta 32 posiciones únicas con salida designada cuando el robot está en posición. |
| Buscar/ Interrumpir | Función que permite la detección de la desviación de la posición de la pieza, desplazando las posiciones<br>programadas de los robots) para compensar. (El dispositivo sensor no está incluido). |

3 Tabla 1-3. Funciones estándar

### 1.1.2 BRAZO ROBÓTICO SC50F-01

El robot SC es un robot de 6 ejes, de alta velocidad y una estructura de alta rigidez, que es óptimo para el manejo y otras aplicaciones. La serie de robots SC, son livianos de uso general, y este modelo en particular se caracteriza por tener la velocidad más rápida en su clase de peso.

4 Figura 1-4. Manipulador Robótico Nachi SC50F

#### 1.1.1.8 Características del robot

- Alta velocidad
- Diseño que ahorra espacio
- Bajo mantenimiento
- Las cargas útiles más grandes en clase
- Movimiento preciso
- Diseño liviano
- Reducción del costo del ciclo de vida
- Opciones de instalación flexibles
- Mayor tiempo de actividad

#### 1.1.1.2 Aplicaciones

- Soldadura por arco

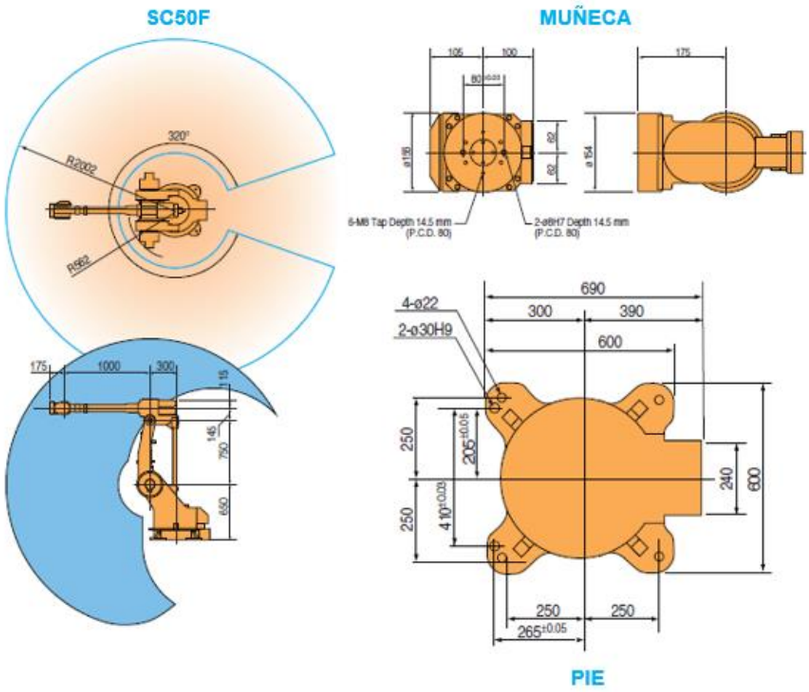
- Desbarbado
- Paletizado
- Manejo
- Montaje
- Manejo de materiales

### 1.1.1.3 Especificaciones básicas

| Ítem | | | | Especificaciones |
|---|---|---|---|---|
| Modelo del Robot | | | | SC50F |
| Construcción | | | | Articulado |
| Numero de ejes | | | | 6 |
| Sistema de manejo | | | | AC Servo Motor |
| Máx.sobre de trabajo | Brazo | J1 | Giratorio | ± 160° (Montaje en pared ±3'°) |
| | | J2 | Adelante y Atrás | ´+ 150° _ -80° |
| | | J3 | Arriba y Abajo | ´+ 132° _ -80° |
| | Muñeca | J4 | Rotación 2 | ± 270° |
| | | J5 | Curva | ± 125° |
| | | J6 | Rotación 1 | ± 450° |
| Máx. Velocidad | Brazo | J1 | Giratorio | 160°/sec |
| | | J2 | Adelante y Atrás | 160°/sec |
| | | J3 | Arriba y Abajo | 160°/sec |
| | Muñeca | J4 | Rotación 2 | 180°/sec |
| | | J5 | Curva | 180°/sec |
| | | J6 | Rotación 1 | 256°/sec |
| Carga Máxima | Muñeca | | | 50kg |
| | Parte superior del brazo | | | 15kg |
| Par de Carga Estatica Permisible | Muñeca | J4 | Rotación 2 | 20kg*m |
| | | J5 | Curva | 20kg*m |
| | | J6 | Rotación 1 | 10kg*m |
| Momento de Inercia Permisible | Muñeca | J4 | Rotación 2 | 8kg*m |
| | | J5 | Curva | 8kg*m |
| | | J6 | Rotación 1 | 2kg*m |
| Repetibilidad de Posición | | | | ± 0.3 mm |
| Temperatura ambiente | | | | 0-45°(32-113°F) |
| Instalación | | | | Piso, invertido, pared |
| Condiciones Ambientales | | | | Temperatura: 0 to 45 °C |
| Masa de Robot | | | | 400 kg |

4 Tabla 1-4. Especificaciones básicas

1.1.1.4 Rangos de operación




5 Figura 1-5. Rangos de operación del robot SC50F

1.1.3 PILOTO VERDE

Piloto compacto IP65 Ø22mm 24Vac/dc Verde c/ led

6 Figura 1-6. Luz piloto verde

| Características |
|---|
| Voltaje de alimentación: 24VAC/VCC |
| Grado de protección: IP65 |
| Diámetro de perforación: 22mm |
| Color: Verde |
| T° de trabajo: -5° a 40°C |

5 Tabla 1-5. Características piloto verde

### 1.1.4 PILOTO ROJO

Piloto compacto IP65 Ø22mm 24Vac/dc Rojo c/ led

7 Figura 1-7. Luz piloto rojo

| Características |
|---|
| Voltaje de alimentación: 24VAC/VCC |
| Grado de protección: IP65 |
| Diámetro de perforación: 22mm |
| Color: Rojo |
| T° de trabajo: -5° a 40°C |

6 Tabla 1-6. Características piloto rojo

### 1.1.5 SELECTOR

Selector 3 posiciones metálico 2 contactos abiertos (NA) completo

8 Figura 1-8. Selector

| Características |
| --- |
| Color: negro. |
| Número de posiciones: 3 mantenidas. |
| Contacto1: 2 contactos abiertos NA. |
| Material: metálico. |
| Intensidad nominal de trabajo Ith: 10 A. |
| Intensidad de trabajo a 220 Vac: 6 A. |
| Tensión máxima de trabajo: 415 V. |
| Resistencia de contacto: 25 Ohm. |
| Grado de protección: IP20. |

7 Tabla 1-7. Características selector

### 1.1.6 VENTOSA

- Adecuada para superficies planas.
- Buena estabilidad y poco movimiento inherente.
- Recomendada cuando la fuerza de elevación es paralela a la superficie del objeto.

9 Figura 1-9. Ventosa piab

### 1.1.7 EJECTOR

piINLINE® generador de vacío MINI Xi, 8-8 mm

10 Figura 1-10. Ejector coax

| Características |
| --- |
| Material: CuZn, PA, SS, Nitrilo (NBR) |
| Nivel de ruido: 68 – 84 dBA |

| |
|---|
| Temperatura: -10 – 80 °C |
| Peso: 21,8 g |
| Presión de alimentación máx.: 0,7 MPa |
| Tecnología patentada de cartuchos COAX®. |
| Capacidad de altos caudales de aspiración en relación con consumo de energía. |
| Diseño en línea liviano con conectores rápidos. |
| Instalación rápida y sencilla directamente en la manguera. |
| Cartucho Si para extra caudal de aspiración. |
| Cartucho Bi para alto rendimiento a presiones bajas de alimentación. |
| Cartucho Xi cuando se requiere alto caudal y alto vacío. |

8 Tabla 1-8. Características ejector

## 1.2 SISTEMA DE VISIÓN ARTIFICIAL

### 1.2.1 RASPBERRY PI 3B

La Raspberry Pi 3 Modelo B es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple de bajo costo, desarrollado en el Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de informática en las escuelas.

11 Figura 1-11. Raspberry Pi Modelo B

1.2.1.1 Características

| Características | Detalle |
|---|---|
| Procesador | Broadcom BCM2387 chipset.<br>1.2GHz Quad-Core ARM Cortex-A53 (64Bit) |
| 802.11 b/g/n Wireless LAN and | EEE 802.11 b / g / n Wi-Fi. Protocol: WEP, WPA WPA2, algorithms AES-CCMP<br>(maximum keylength of 256 bits), the maximum range of 100 meters. |
| Bluetooth 4.1 (Bluetooth Classic and LE) | IEEE 802.15 Bluetooth, symmetric encryption algorithm Advanced Encryption Standard<br>(AES)with 128-bit key, the maximum range of 50 meters. |
| GPU | Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0,<br>hardware-acceleratedOpen VG, and 1080p30 H.264 high-profile decode.<br>Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure |
| Memoria | 1GB LPDDR2 |
| Sistema operativo | Boots desde la tarjeta Micro SD, ejecutando una versión del sistema operativo Linux o Windows 10 IoT |
| Dimensiones | 85 x 56 x 17mm |
| Alimentación | Puerto Micro USB 5V1, 2.5A |

9 Tabla 1-9. Características Raspberry Pi 3 Modelo B

1.2.1.2 Conectores

| Conector | Detalle |
|---|---|
| Ethernet | 10/100 BaseT Ethernet socket |
| Salida de Video | HDMI (rev 1.3 & 1.4)<br>Compuesto RCA (PAL y NTSC) |
| Salida de Audio | Salida de Audio 3.5mm jack<br>HDMI<br>Conector USB 4 x USB 2.0 |
| Conector GPIO | Cabezal de expansión de 40 pines de 2,54 mm (100 mil): cinta de 2x20<br>Proporciona 27 pines GPIO, así como líneas de suministro +3.3 V, +5 V y GND |
| Conector Camera | Interfaz serie de cámara MIPI de 15 pines (CSI-2) |
| Conector Pantalla | Interfaz serie de pantalla (DSI) Conector de cable flexible plano de 15 vías<br>con dos carriles de datos y una línea de reloj |
| Ranura para tarjeta de memoria | Push/pull Micro SDIO |

10 Tabla 1-10. Conectores Raspberry Pi Modelo B

1.2.1.3 Beneficios clave

- Bajo costo
- Formato consistente de carcasa
- Rapidez de procesamiento x 10
- Conectividad agregada

#### 1.2.1.4 Aplicaciones clave

- PC / tableta / laptop de bajo costo
- Aplicaciones de IoT
- Centro de Medios
- Robótica
- Industrial / Domótica
- Servidor / servidor en la nube
- Control de seguridad
- Cámara web y juegos
- Punto de acceso inalámbrico
- Detección / monitoreo ambiental (por ejemplo, estación meteorológica)

### 1.2.2 CÁMARA RASPBERRY PI

El módulo de cámara de alta definición es compatible con la Raspberry Pi modelo A y B. Provee una alta sensibilidad, baja diafonía y captura de imágenes de bajo ruido en un diseño ultra pequeño y liviano. El módulo de la cámara se conecta a la placa Raspberry Pi a través del conector CSI diseñado específicamente para la interfaz a las cámaras. El bus CSI es capaz de velocidades de datos extremadamente altas, y lleva exclusivamente datos de píxeles al procesador BCM2835.

12 Figura 1-12. Módulo cámara

| Características | Detalle |
|---|---|
| Sensor de imagen | Sensor de imagen Omnivision 5647 CMOS en un módulo de enfoque fijo con filtro IR integrado |
| Resolución | 5-megapixel |
| Resolución de imagen fija | 2592 x 1944 |
| Velocidad máxima de transferencia de imágenes | 1080p: 30 fps (codificar y decodificar)<br>720p: 60fps |
| Conexión a la Raspberry Pi | Cable plano de 15 pines, a la interfaz serie de la cámara MIPI de 15 pines dedicada (CSI-2) |
| Funciones de control de imagen | Control de exposición automático<br>Balance de blancos automático<br>Filtro de banda automático<br>Detección de luminancia automática de 50/60 Hz<br>Calibración automática del nivel de negro |
| Rango de temperatura | Funcionamiento: -30 ° a 70 °<br>Imagen estable: 0 ° a 50 ° |
| Tamaño de la lente | ¼” |
| Dimensiones | 20 x 25 x 10mm |
| Peso | 3g |

11 Tabla 1-11. Características módulo cámara

### 1.2.3 CARGADOR RASPBERRY PI

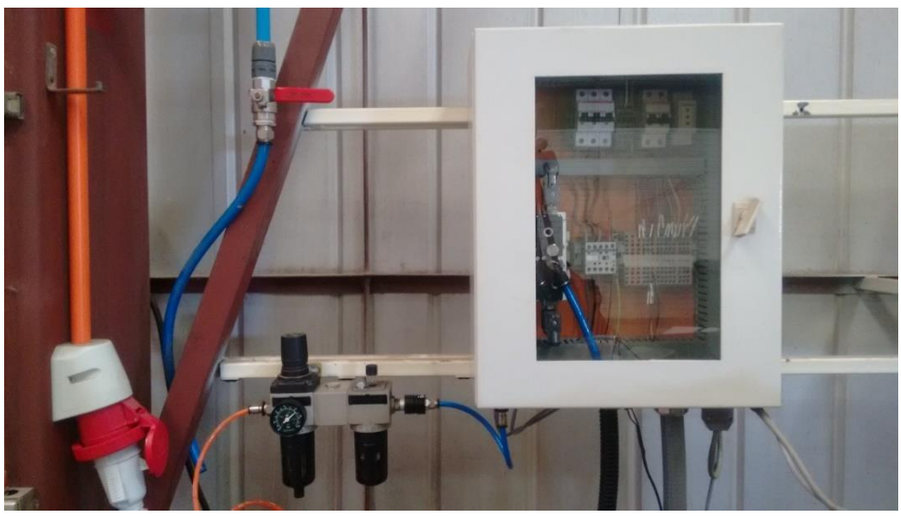
Fuente de alimentación de 3 generaciones, perfecto aplicado a Raspberry Pi 3/2 Modelo B.

13 Figura 1-13. Cargador rasberry pi

| Características |
| --- |
| Salida: Real 5V 2.5A |
| Tipo de enchufe: EU Plug |
| Color: Negro |
| Tamaño del cable: 95CM |
| Diámetro: 3.5mm |
| Voltaje de entrada: AC 100-240V, 50 / 60Hz |
| Salida: DC 5.0V-2500mA |
| Micro USB Interfaz |
| Diseño especial para Raspberry Pi 3/2 Modelo B |

12 Tabla 1-12. Características cargador raspberry pi

## 1.3 TABLERO DE CONTROL ELECTROPNEUMATICO



14 Figura 1-14. Tablero electropneumatico de control

### 1.3.1 COMPONENTES MECÁNICOS

#### **1.3.1.1** Gabinete

Caja metálica 500x400x250mm IP65 R7032 p/vidrio s/placa

15 Figura 1-15. Gabinete

| Características |
|---|
| Conformidad con la norma: CEI EN 62208 |
| Grado de protección: IP65 |
| Tipo de material: Metálico |
| Sitio de instalación: en el interior |
| Método de fijación: piso/pared |
| Temperatura ambiente: - 5 a + 40°C |
| Condiciones atmosféricas-humedad relativa: 50% a 40°C 90% a 20°C |

13 Tabla 1-13. Características gabinete

#### 1.3.1.2 Riel DIN

16 Figura 1-16. Riel DIN

| Características |
| --- |
| Material: acero galvanizado |
| Enchapado: CR + 3 plateado |

14 Tabla 1-14. Características riel DIN

**1.3.1.3** Regleta ranurada

Canaleta ranurada 25x60x2000mm con tapa (4/6) h60

17 Figura 1-17. Canaleta ranurada

1.3.2 COMPONENTES ELÉCTRICOS

**1.3.2.1** Automático

Interruptor 3Px10A 6kA Intensidad de corte nominal de cortocircuito curva C, S203-C10

18 Figura 1-18. Automático 3P 10A

**1.3.2.2** Contactor

A16-30-10-24V-50HZ -  Contactor, 17 A, Panel, 690 VAC, 3PST-NO, 3 polos, 7.5 Kw

19 Figura 1-19. Contactor 24V

**1.3.2.3** Bornera

Borne pour rail din 4 cond. gris 18a - 279-831

20 Figura 1-20. Bornera

**1.3.2.4** Separador

Platine fin et intermediaire mont rail - 280-315

21 Figura 1-21. Separador

**1.3.2.5** Tope

Accesorio para Conector, Tope sin Tornillos- 249-116

22 Figura 1-22. Tope

1.3.3 COMPONENTES PNEUMÁTICOS/ELECTROPNEUMÁTICOS

**1.3.3.1** Filtro Regulador Lubricador

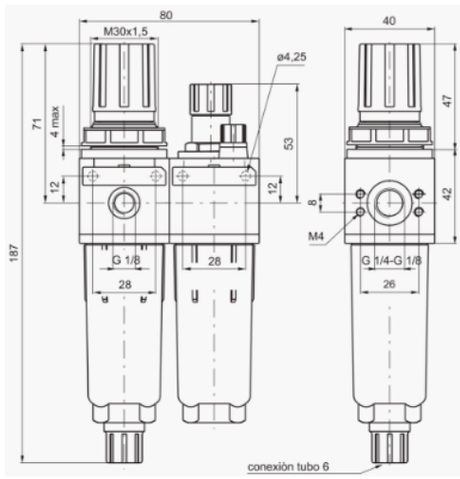
El filtro es un dispositivo neumático que se utiliza para el tratamiento del aire comprimido: filtración, regulación y lubricación.

23 Figura 1-23. Filtro regulador lubricador

| Características |
|---|
| Conexiones: 1/8’-1/4’ |
| Presión máx. de alimentación: 13bar – 1,3MPa |
| Temperatura ambiente máx. (a 10bar): 50°C |
| Ataque manómetro: 1/8’ |
| Peso con cuerpo en tecnopolimero: 295gr. |
| Peso con cuerpo en zamak: 560gr. |

| |
|---|
| Campo de regulación de la presión: 0-2 bar/0-4 bar/0-8 bar/0-12 bar |
| Umbral de filtración: 5µ -20µ - 50µ |
| Capacidad de condensación máx.: 17cm3 |
| Tipo de aceite: FD22 – HG32 |
| Capacidad aceite vaso: 32cm3 |

15 Tabla 1-15. Características FRL




24 Figura 1-24. Dimensiones FRL

**1.3.3.2** Válvula Colizante

Válvula colizante 1/8’, para el paso y corte de aire comprimido.

25 Figura 1-25. Válvula colizante

**1.3.3.3** Manifold

Manifold para electroválvula multi-via de alumínio

26 Figura 1-26. Manifold

**1.3.3.4** Electroválvula

Válvula 5/2’ 1/4’electromecánica Bob/Bob 24V, diseñada para controlar el paso de un fluido por un conducto o tubería.

27 Figura 1-27. Electroválvula

## 1.4 CORREA TRANSPORTADORA

Una cinta transportadora o banda transportadora o transportador de banda o cintas francas es un sistema de transporte continúo formado por una banda continua que se mueve entre dos tambores.

La banda es arrastrada por la fricción de sus tambores, que a la vez este es accionado por su motor.

28 Figura 1-28. Correa transportadora

| Características |
|---|
| Largo: 2250mm |
| Ancho:570mm |
| Alto: 770mm |
| Material estructura: Acero |
| Material correa:  Caucho |
| Color: Gris/Rojo/Blanco |

16 Tabla 1-16. Características correa transportadora

### 1.4.1 COMPONENTES

#### **1.4.1.1** Motor Trifásico

29 Figura 1-29. Motor trifásico

| Características |
| --- |
| Marca: Nord Technik |
| Modelo: MS8015 |
| INS.CL: F |
| IP: 55 |
| S: 1 |
| V. Δ/Y: 220/380 |
| Hz: 50 |
| kW: 0.55 |
| rpm:1390 |
| A. Δ/Y: 2.72/1.57 |
| COSØ: 0.75 |

17 Tabla 1-17. Características motor trifásico

**1.4.1.2** Sensor Fotoeléctrico O5P51A

- Para aplicaciones en áreas peligrosas
- Carcasa robusta con soporte de acero inoxidable
- Muy largo alcance
- Sensibilidad ajustable mediante botones de aprendizaje
- Modo de encendido / oscuridad programable

30 Figura 1-30. Sensor fotoeléctrico

| Características |
| --- |
| Voltaje de operación: 10…30VDC |
| Corriente: 20mA |
| Diseño eléctrico: Salida de Transistor PNP |

| |
|---|
| Rango de detección: 75 a 10000mm |
| Rango ajustable: Si |
| Temperatura ambiente: -10…50°C |
| Dimensiones: 105x30x62mm |
| **Peso: 328g** |

18 Tabla 1-18. Características sensor fotoeléctrico

## 1.5 MESA DE OBJETOS CLASIFICADOS

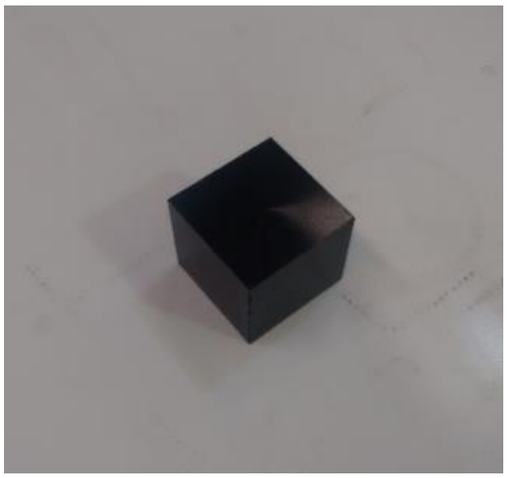
31 Figura 1-31. Mesa de objetos clasificados

| Características |
|---|
| Largo: 860mm |
| Ancho:380mm |
| Alto: 1200mm |
| Material: Acero |
| Color: Amarillo/Negro |

19 Tabla 1-19. Características mesa clasificadora de objetos

## 1.6 OBJETOS

### 1.6.1 Objeto Cuadrado



32 Figura 1-32. Objeto cuadrado

| Características |
|---|
| Largo: 50mm |
| Ancho:50mm |
| Alto: 45mm |
| Material: Tecnil |
| Color: Negro |

20 Tabla 1-20. Características objeto cuadrado

### 1.6.2 Objeto Cilíndrico

33 Figura 1-32. Objeto redondo

| Características |
|---|
| Diámetro: 50mm |
| Alto: 45mm |
| Material: Tecnil |
| Color: Negro |

21 Tabla 1-21. Características objeto redondo

# 2. CAPÍTULO 2: SOFTWARE

## 2.1 RASPBIAN

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Strech (Debian 9.4) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática. El lanzamiento inicial fue en junio de 2012.

Técnicamente el sistema operativo es un port no oficial de Debian armhf para el procesador (CPU) de Raspberry Pi, con soporte optimizado para cálculos en coma flotante por hardware, lo que permite dar más rendimiento en según qué casos. El port fue necesario al no haber versión Debian armhf para la CPU ARMv6 que contiene el Raspberry PI.3.

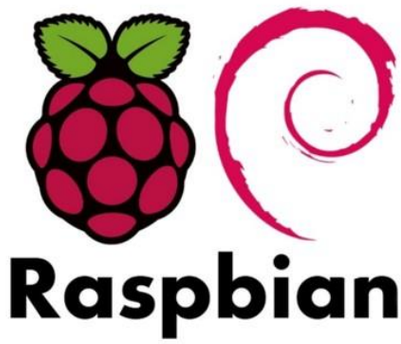
La distribución usa LXDE como escritorio y Chromium como navegador web. Además contiene herramientas de desarrollo como IDLE para el lenguaje de programación Python o Scratch, y diferentes ejemplos de juegos usando los módulos Pygame.

Destaca también el menú "raspi-config" que permite configurar el sistema operativo sin tener que modificar archivos de configuración manualmente. Entre sus funciones, permite expandir la partición root para que ocupe toda la tarjeta de memoria, configurar el teclado, aplicar overclock, etc.

El 17 de diciembre de 2012, junto a la versión 2012-12-16-wheezy-raspbian de Raspbian, se lanzó la tienda de aplicaciones "Pi Store", que en el momento de salida incluía desde aplicaciones como LibreOffice o Asterisk a juegos como Freeciv o OpenTTD. En esta plataforma se puede poner a disposición de todos los usuarios de Raspbian, mediante moderación y posterior lanzamiento, contenidos gratuitos o de pago, como archivos binarios, código python, imágenes, audio o vídeo. Además se quiere incluir documentación acerca del Raspberry Pi como la revista MagPi y tutoriales de proyectos.

Actualmente, la tienda ya no se incluye en el sistema pero se puede instalar ejecutando el comando "sudo apt-get update && sudo apt-get install pistore"

Al ser una distribución de GNU/Linux las posibilidades son infinitas. Todo software de código abierto puede ser recompilado en la propia Raspberry Pi para arquitectura armhf que pueda ser utilizado en el propio dispositivo en caso de que el desarrollador no proporcione una versión ya compilada para esta arquitectura. Además esta distribución, como la mayoría, contiene repositorios donde el usuario puede descargar multitud de programas como si se tratase de una distribución de GNU/Linux para equipos de escritorio. Todo esto hace de Raspberry Pi un dispositivo que además de servir como placa con microcontrolador clásica, tenga mucha de la funcionalidad de un ordenador personal. Lo que lo puede convertir en una alternativa a los ordenadores personales, especialmente para personas con pocos recursos, para la extensión de la informática en países subdesarrollados o para aplicaciones que no soliciten muchos requerimientos. [6]




34 Figura 2-1. Raspbian

## 2.2 PYTHON

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License,3 que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores. [7]

35 Figura 2-2. Python

### 3.2.1 CARACTERÍSTICAS Y PARADIGMAS

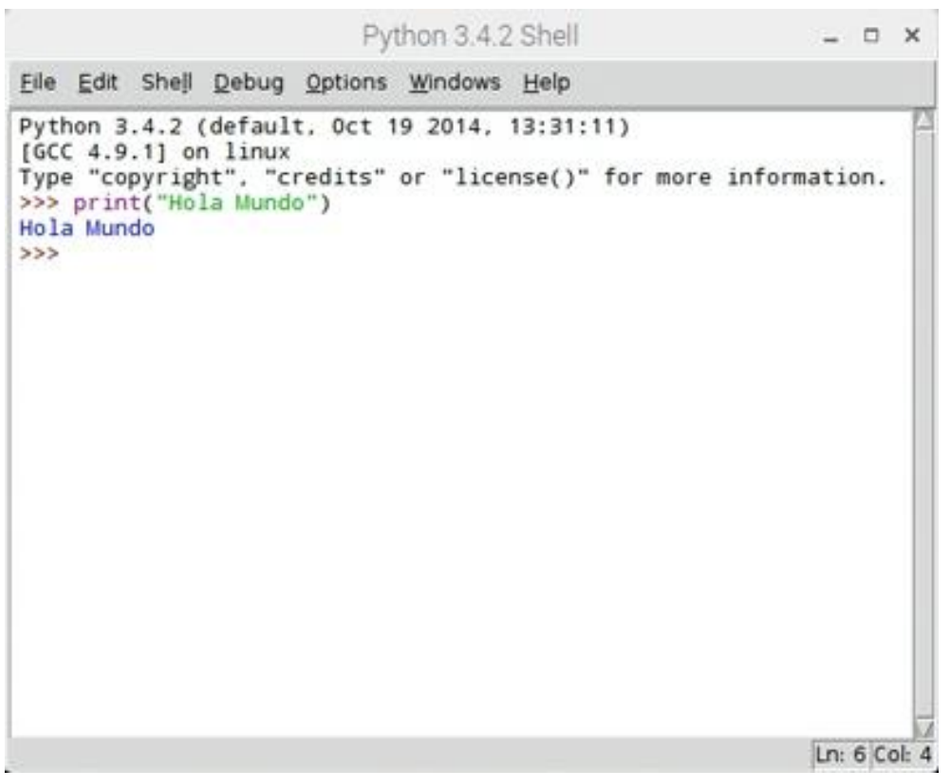
Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostiles a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.




36 Figura 2-3. IDE de Python

## 1.3 OPENCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estérea y visión robótica.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel (IPP). [8]

Aplicaciones conocidas

- **OpenCV** ha sido usada en el sistema de visión del vehículo no tripulado Stanley de la Universidad de Stanford, el ganador en el año 2005 del Gran desafío DARPA.
- **OpenCV** se usa en sistemas de vigilancia de vídeo.
- **OpenCV** es la clave en el programa Swistrack, una herramienta de seguimiento distribuida.

37 Figura 2-4. OpenCV

## 1.4 FILEZILLA

Filezilla es un cliente FTP, gratuito, libre (GPL) y de código abierto. Sustenta FTP, SFTP y FTP sobre SSL. Inicialmente solo diseñado para funcionar bajo Windows, desde la versión 3.0.0, gracias al uso de wxWidgets, es multiplaforma, estando disponible además para otros sistemas operativos, entre ellos Linux, FreeBSD y MacOS X.

Las principales características son Site Manager (Administrador de sitios), Message Log (Registro de mensajes), y Transfer Queue (Cola de transferencia).

El administrador de sitios permite a un usuario crear una lista de sitios FTP con sus datos de conexión, como el número de puerto a usar, o si se utiliza inicio de sesión normal o anónima. Para el inicio normal, se guarda el usuario y opcionalmente la contraseña.

El registro de mensajes se muestra en la parte de arriba de la ventana. Muestra en forma de consola los comandos enviados por FileZilla y las respuestas del servidor remoto.

La vista de archivo y carpeta, situada debajo del registro de mensajes, proporciona una interfaz gráfica para FTP. Los usuarios pueden navegar por las carpetas y ver y alterar sus contenidos tanto en la máquina local como en la remota, utilizando una interfaz de tipo árbol de exploración. Los usuarios pueden arrastrar y soltar archivos entre los ordenadores local y remoto.

La cola de transferencia, situada en la parte de abajo de la ventana, muestra en tiempo real el estado de cada transferencia activa o en cola.

38 Figura 2-5. Filezilla

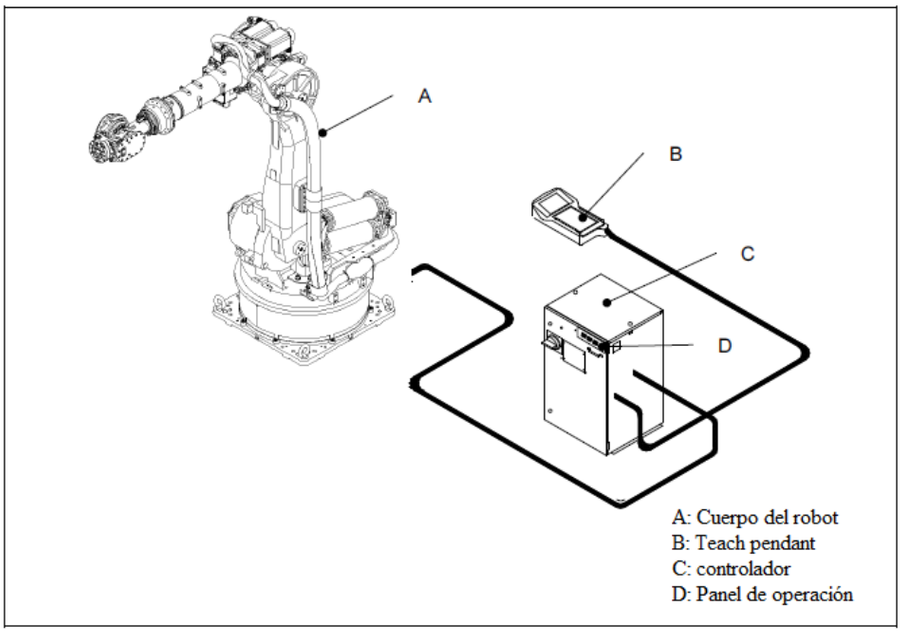
# 3. CAPÍTULO 3: PROGRAMACIÓN

## 3.1 CONCEPTOS BASICOS QUE SE APLICAN AL ROBOT

Esta sección describe los conceptos básicos que se aplican al robot y el conocimiento básico sobre el robot.

### 3.2.1 SISTEMA ROBÓTICO

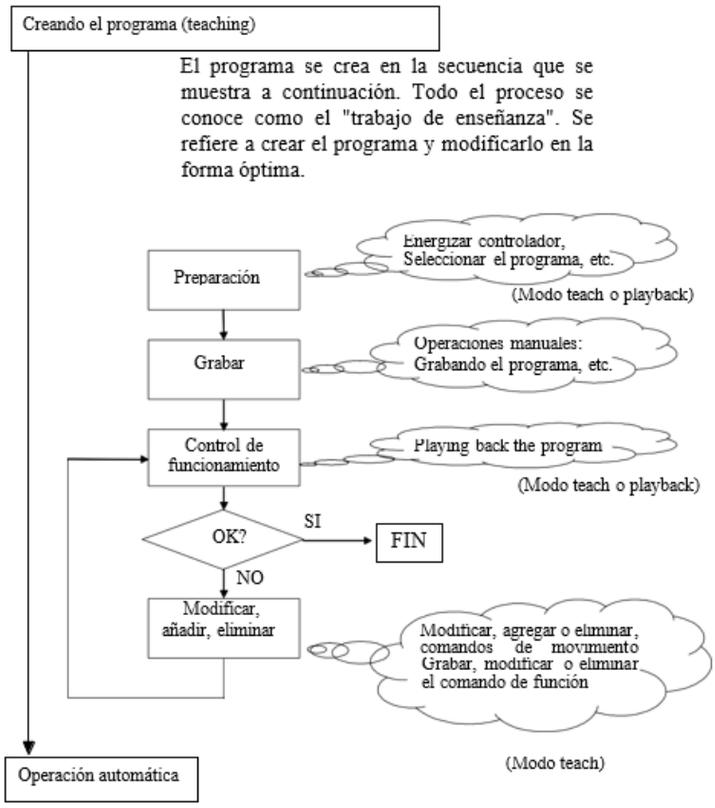
Un sistema de robot normalmente denota una combinación de un robot, un teach pendant y periféricos que están todos conectados a un controlador.




39 Figura 3-1. Diagrama de la configuración básica para el robot (Controlador AX20)

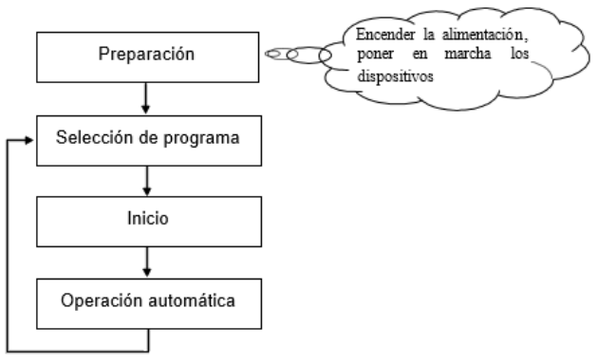
Descripción general: Enseñanza para la operación automática

Para operar continuamente el robot es necesario proceder como se muestra en la figura.




40 Figura 3-2. Trabajo de enseñanza

Después de que se ha completado la creación del programa, se realiza una operación automática.

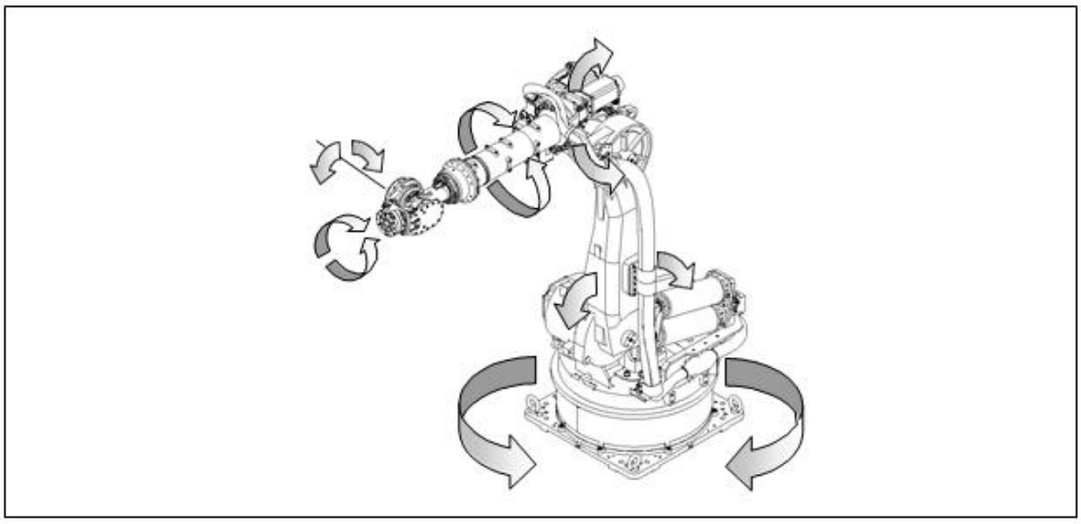



41 Figura 3-3. Operación automática

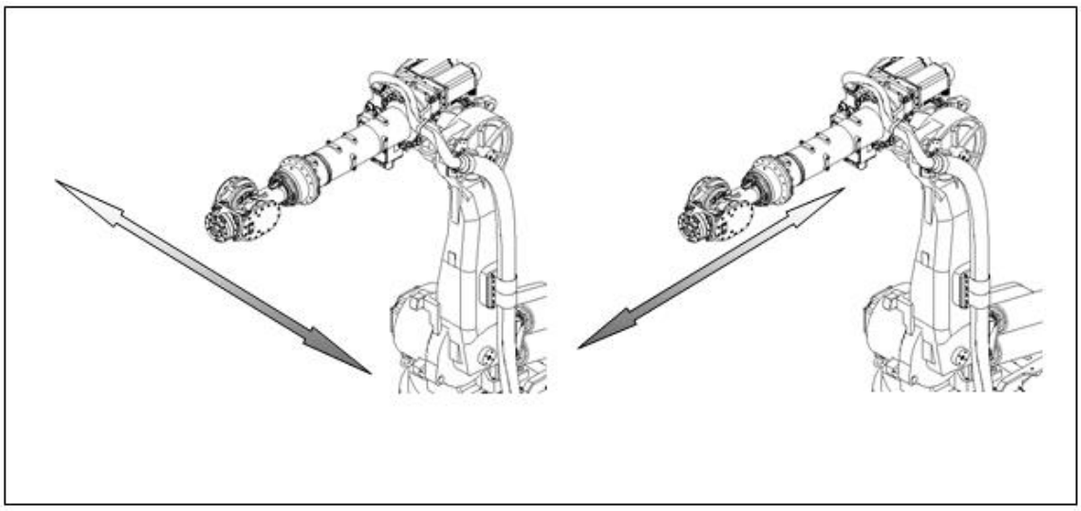
OPERACIÓN MANUAL

La "operación manual" se refiere a mover el robot con el uso del teach pendant. Guíe al robot hasta el punto que se grabará mediante "operación manual".

El funcionamiento manual tiene varios modos, incluido el modo en el que cada eje del "robot" se opera por separado, y un modo en el que la punta del robot se mueve en línea recta.



42 Figura 3-4. Operando los ejes por separado (operación de eje independiente)



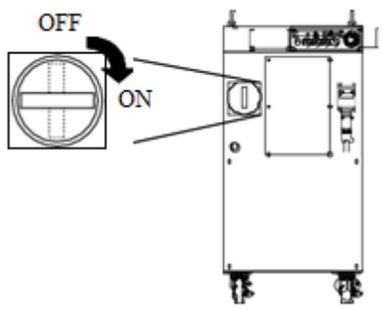
43 Figura 3-5. Mover la punta del robot en línea recta (operación de coordenadas del robot)

## 3.2 CONTROLADOR AX20-01

### 3.2.1 PROGRAMA BÁSICO – MODO MANUAL

**Paso 1:** Energizar y encender Controlador AX20.

Colocar el interruptor en la posición ON. El sistema AX20 se inicia automáticamente y se inicia el autodiagnóstico.



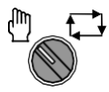

44 Figura 3-6. On/Off Controlador AX20

Si el autodiagnóstico se completa sin que se detecten problemas, se mostrará una pantalla como la que se muestra a continuación en la unidad de programación.

45 Figura 3-7. Pantalla de inicio

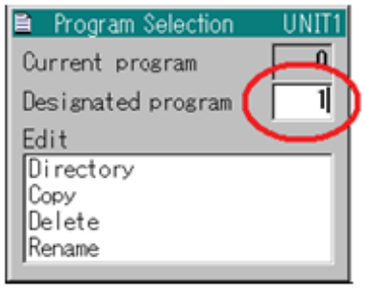
El robot ahora está listo para operar.

**Paso 2:** Seleccionar modo manual en el controlador y teach pendant.

| Mo<br>do | Posición del selector | Pantalla Teach pendant |
|---|---|---|
| **Teach mode** |  | |

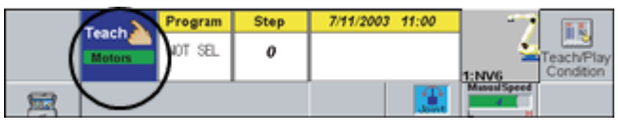
22 Tabla 3-1. Selección modo manual

**Paso 3:** Seleccionar número de programa.



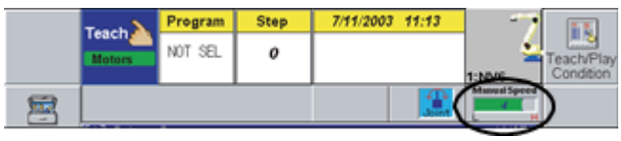

46 Figura 3-8. Selección de programa

**Paso 4:** Energizar servomotores.



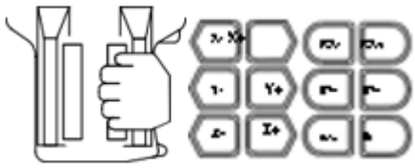

47 Figura 3-9. Energizar servomotores

**Paso 5:** Ajustar velocidad de movimiento manual.



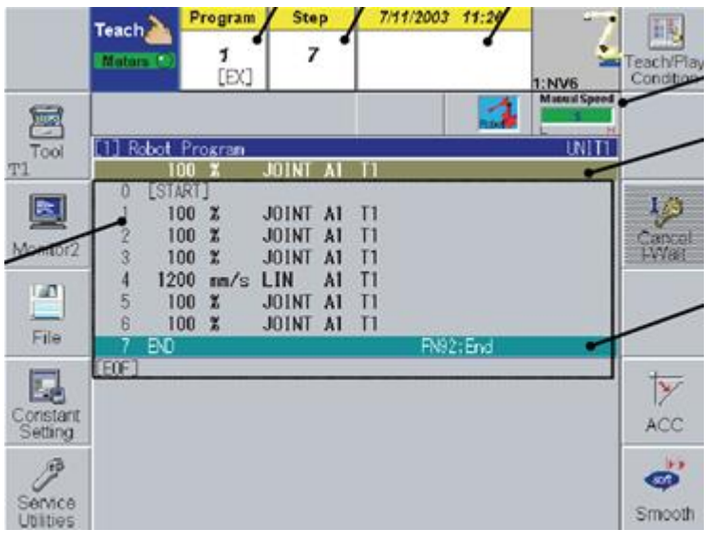

48 Figura 3-10. Ajustar velocidad

**Paso 6:** Presionar interruptor de hombre muerto y el eje del robot que se desee mover.



49 Figura 3-11. Interruptor de hombre muerto y control de ejes

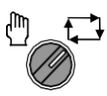
**Paso 7:** Gravar posiciones




50 Figura 3-12. Grabación de posiciones

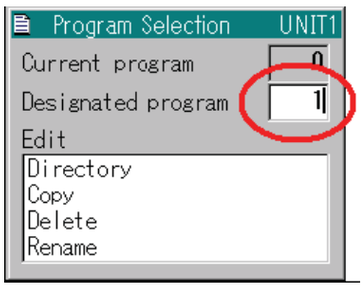
### 3.2.2 PROGRAMA BÁSICO – MODO AUTOMÁTICO

**Paso 1:** Seleccionar modo automático en el controlador y teach pendant**.**

| Mo<br>do | Posición del selector | Pantalla Teach pendant |
|---|---|---|
| **Teach mode** |  | |

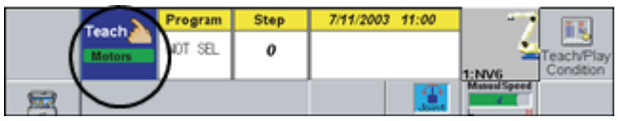
51 Tabla 3-2. Selección modo automático

**Paso 2:** Seleccionar número de programa




52 Figura 3-13. Selección de número de programa

**Paso 3:** Energizar servomotores




53 Figura 3-14. Energizar servomotores

**Paso 4:** Ajustar velocidad de movimiento automático

**Paso 5:** Iniciar programa

## 3.3 RASPBERRY PI MODELO 3B

### 3.3.1 INSTALACIÓN DE SISTEMA OPERATIVO

**Paso 1:** Descargar RASPBIAN

En primer lugar, se debe descargar Raspbian desde la página oficial de raspberry pi.

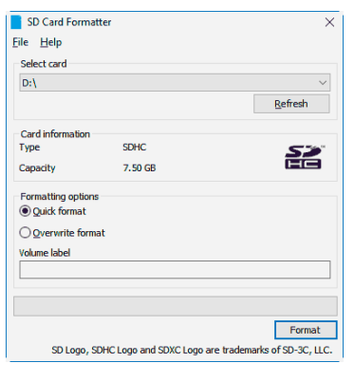
54 Figura 3-15. Descarga de Raspbian

Luego, entre las dos versiones de imagen ISO, elegimos descargar la versión completa, considerando que la versión Lite contiene únicamente los ficheros necesarios para ejecutar RASPBIAN, por lo cual la imagen tiene un tamaño menor. Por lo que al iniciar por primera vez la Raspberry Pi deberá descargarse al inicio.

55 Figura 3-16. Descargar ZIP

**Paso 2:** Formatear tarjeta SD

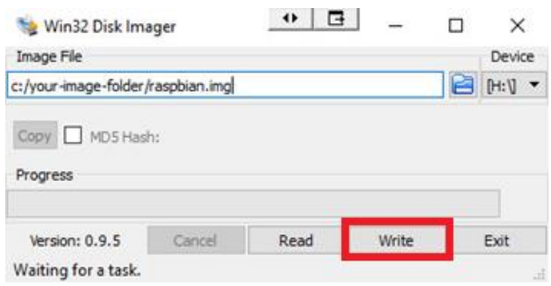
Una vez tengamos la ISO descargada en nuestro PC, debemos utilizar el programa “SD Card Formatter” para formatear la tarjeta de memoria SD, de forma que sea booteable.




56 Figura 3-17. SD Card Formatter

**Paso 3:** Copiar ISO en la memoria SD

Luego, se abre el programa “Win32 Disk Imager”. Se selecciona la imagen ISO y la tarjeta SD donde será copiada.




57 Figura 3-18. Win32 Disk Imager

**Paso 4:** Iniciar Raspberry Pi

Una vez finalizada la copia de la ISO, se introduce la tarjeta SD en la Raspberry Pi, para luego energizar la Raspberry.

58 Figura 3-19. Inicio raspberry pi

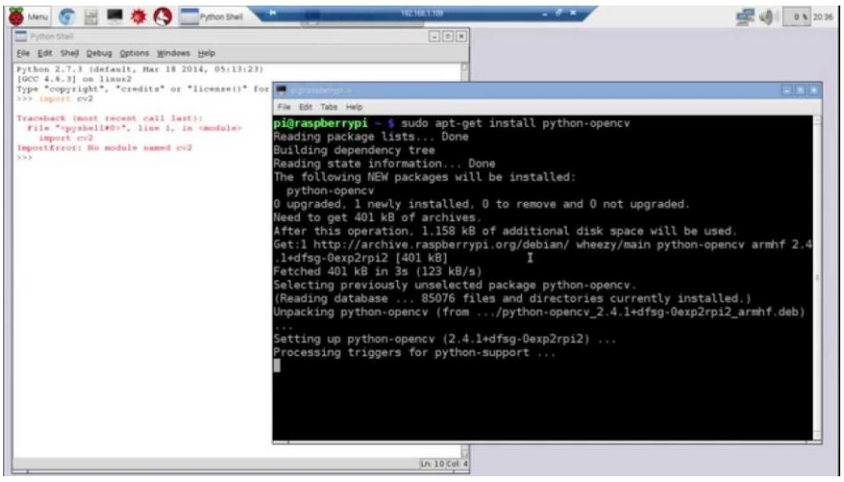
### 3.3.2 INSTALAR OPENCV EN RASPBERRY PI

**Paso 1:** Tener Raspberry Pi configurada y preparada.

**Paso 2:** Abrir la Terminal y escribir:

- sudo apt-get update
- sudo apt-get install python-opencv

- sudo apt-get -f install



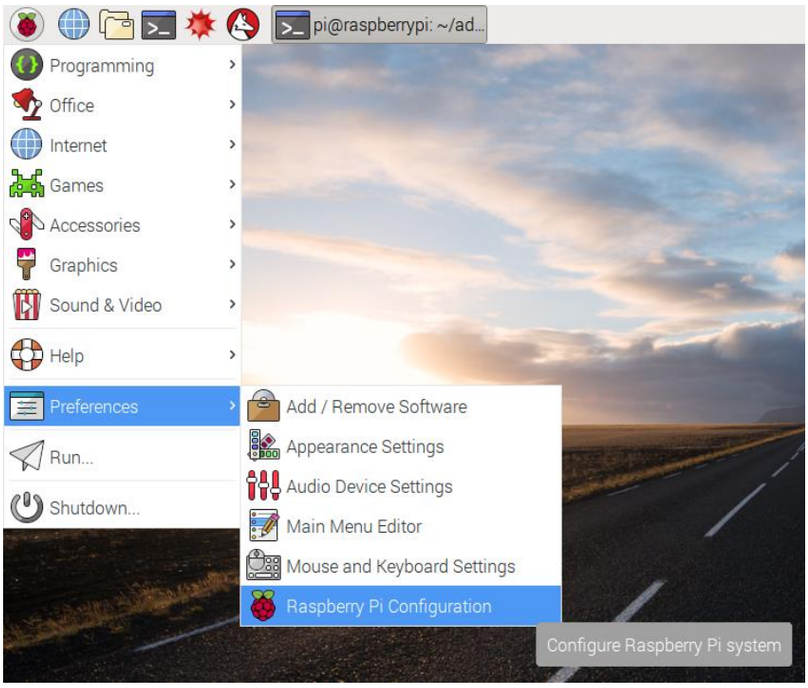

59 Figura 3-20. Instalación Open CV en Raspberry Pi

### 3.3.3 CONEXIÓN DEL MÓDULO DE LA CÁMARA

**Paso 1:** Ubicar el puerto de la cámara y con la Raspberry Pi apagada, conectar la cámara.

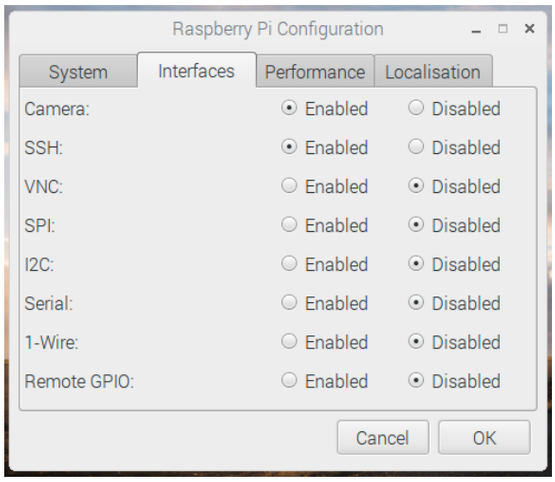
**Paso 2:** Encender Raspberry Pi

**Paso 3:** Abrir la herramienta de configuración de Raspberry Pi en el menú principal:




60 Figura 3-21. Configuración de raspberry pi

**Paso 4:** Habilitar software de la cámara:

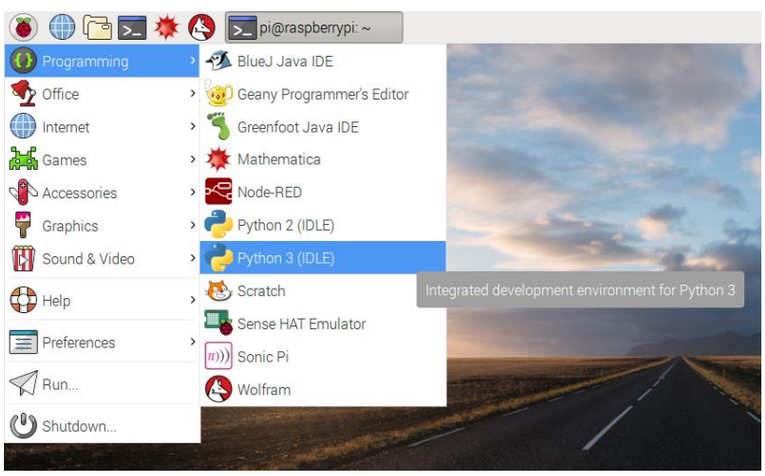



61 Figura 3-22. Habilitar cámara

**Paso 5:** Reiniciar Raspberry Pi.

### 3.3.4 PROGRAMA INICIAL - VISTA PREVIA DE LA CÁMARA

**Paso 1:** Abrir Python 3 desde el menú principal:




62 Figura 3-23. Abrir python

**Paso 2:** Abra un nuevo archivo y guárdelo como “camera.py”. Es importante que no lo guardes como “picamera.py”.

**Paso 3:** Ingresar el siguiente código:

```
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.start_preview()
sleep(10)
camera.stop_preview()
```

63 Figura 3-24. Primer programa

**Paso 4:** Guardar con Ctrl + S y ejecutar con F5. La vista previa de la cámara se muestra durante 10 segundos y luego se cierra.

# 4. CAPÍTULO 4: EXPERIENCIAS Y RESULTADOS

## 4.1 EXPERIENCIA 1 – ACTIVAR/DESACTIVAR SALIDAS

En esta experiencia se pretende interactuar con 5 de las 32 señales digitales de salida de la placa de I/O del controlador AX20, activando y desactivando estas, de forma manual desde el teach pendant a través de las Salidas de usuario, con el objetivo de adquirir conocimientos realizando prácticas de operación con el controlador y sus salidas.

Para la experiencia es necesario utilizar 5 salidas digitales del controlador AX20, a estas salidas se conecta una luz piloto verde, una luz piloto roja, dos bobinas de una electroválvula, una bobina de un contactor, como se detalla a continuación:

| OUTPUT | Descripción |
|---|---|
| 1 | Luz Piloto Verde |
| 2 | Luz Piloto Verde |
| 3 | Electroválvula1-01 |
| 4 | Electroválvula1-02 |
| 5 | Contactor |

23 Tabla 4-1. Salidas experiencia 1

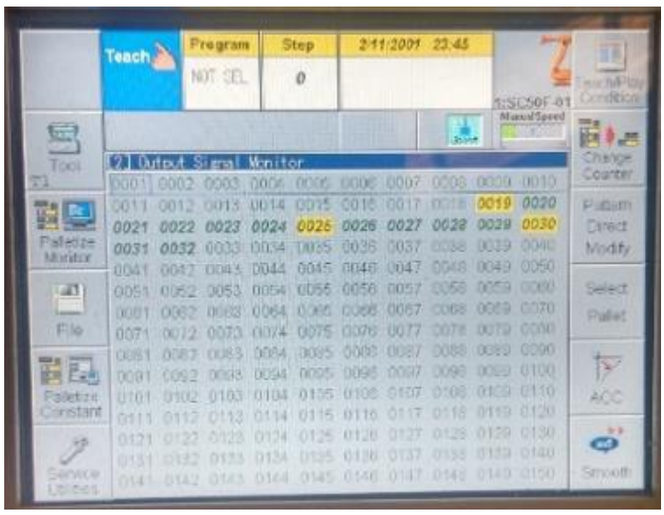
**Paso 1:** Encender el Controlador AX20 y configurar para trabajar en modo manual, al igual que el teach pendant.

**Paso 2:** Presionar Enable + F3 (Monitor 2)

64 Figura 4-1. Monitor 2

**Paso 3:** Seleccionar 8 User OUTPUTS

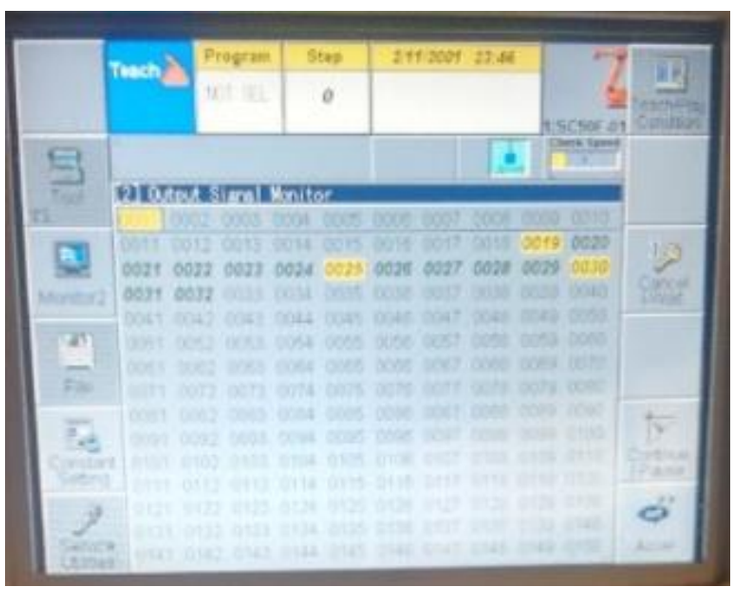
Para seleccionar 8 User OUTPUTS, es necesario presionar 8 y luego ENTER.




65 Figura 4-2. User OUTPUTS

**Paso 4:** Activar salidas

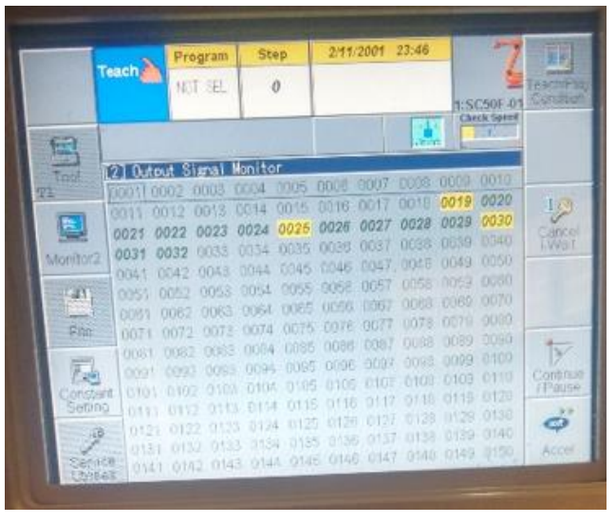
Para activar salidas, es necesario posicionarse con el cursor sobre la salida que se desea activar. Luego presionar ENABLE + ON.




66 Figura 4-3. Activación de salidas

**Paso 5:** Desactivar salidas

Para desactivar salidas, es necesario posicionarse con el cursor sobre la salida que se desea desactivar. Luego presionar ENABLE + OFF.




67 Figura 4-4. Desactivación de salidas

**Resultado**

El resultado fue el esperado, puesto que se consiguió realizar la operación de activar y desactivar los actuadores conectados a las salidas de la placa I/O del controlador AX20.

## 4.2 EXPERIENCIA 2 – VISUALIZAR ESTADO DE ENTRADAS

En esta experiencia se pretende interactuar con 3 de las 32 señales digitales de entrada de la placa de I/O del controlador AX20, activando y desactivando estas, de forma manual por contacto para el caso del selector y de forma manual sin contacto para el caso del sensor fotoeléctrico, visualizando sus estados en el teach pendant a través de las Entradas de usuario, con el objetivo de adquirir conocimientos realizando prácticas de operación con el controlador y sus entradas.

Para la experiencia es necesario utilizar 3 entradas digitales del controlador AX20, a estas entradas se conecta un selector de 3 posiciones y un sensor fotoeléctrico, como se detalla a continuación:

| INPUT | Descripción |
|---|---|
| 17 | Selector posición 1 (luz piloto verde) |
| 18 | Selector posición 2 (luz piloto roja) |
| 19 | Sensor fotoeléctrico |

24 Tabla 4-2. Entradas experiencia 2

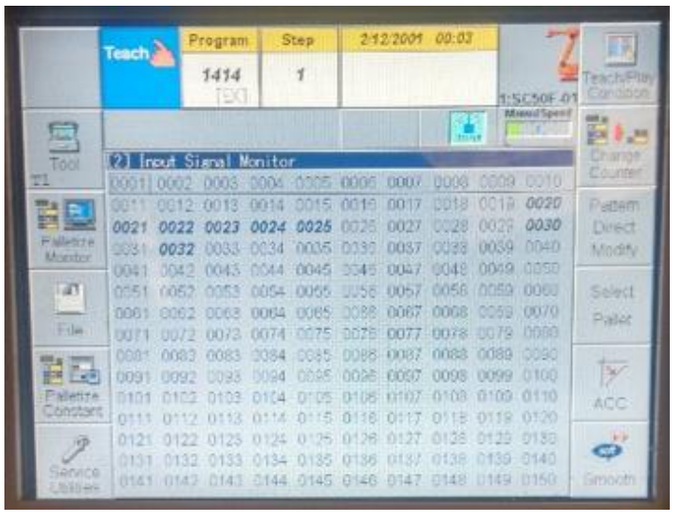
**Paso 1:** Encender el Controlador AX20 y configurar para trabajar en modo manual, al igual que el teach pendant.

**Paso 2:** Presionar Enable + F3 (Monitor 2)

Para seleccionar 7 User INPUTS, es necesario presionar 7 y luego ENTER.

68 Figura 4-5. Monitor 2

**Paso 3:** Seleccionar 7 User INPUTS




69 Figura 4-6. User INPUTS

**Paso 4:** Activar/Desactivar entradas

Para activar las entradas, es necesario manualmente hacer que dichas entradas cambien de estado, y el resultado es reflejado en la pantalla del controlador.

70 Figura 4-7. Ejemplo activar/desactivar entrada 17

**Resultado**

El resultado fue el esperado, puesto que se consiguió realizar la operación de activar y desactivar los sensores conectados a las entradas de la placa I/O del controlador AX20.

## 4.3 EXPERIENCIA 3 – PARTIR/PARAR CORREA TRANSPORTADORA

La siguiente experiencia consiste en crear un programa de robot, para ser ejecutado en modo automático, que realice la función de activar una correa transportadora a través de la señal de un selector en posición 1 y desactivar la correa con la señal del mismo selector en posición 2.

Para llevar a cabo esta experiencia es necesario utilizar una salida y dos entradas del controlador AX20, como se presentan a continuación:

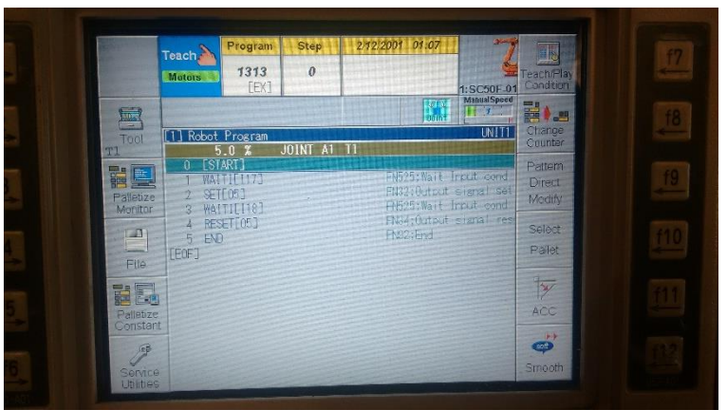
| OUTPUT | Descripción |
|---|---|
| 1 | Luz Piloto Verde |
| INPUT | Descripción |
| 17 | Selector posición 1 (luz piloto verde) |
| 18 | Selector posición 2 (luz piloto roja) |

25 Tabla 4-3. Salida y entradas experiencia 3

**Paso 1:** Encender el Controlador AX20 y configurar para trabajar en modo manual, al igual que el teach pendant.

**Paso 2:** Crear programa

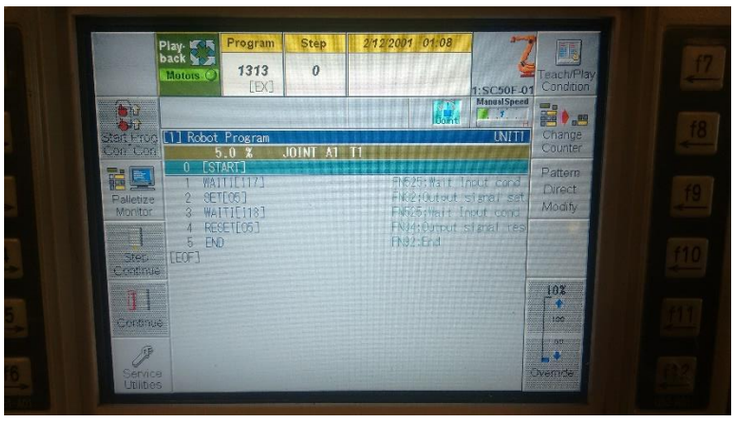
Para la creación del programa es necesario confirmar que los sensores y el actuador están conectados en las entradas y salida definida(s). Para confirmar, revisar EXPERIMENTO 1 Y 2.




71 Figura 4-8. Creación de programa experiencia 3

**Paso 3:** Correr programa en modo automático

Para correr el programa creado en modo automático, se configura el controlador AX20 para trabajar en modo automático, al igual que el teach pendant. Luego se energizan los servomotores y en el panel de control, se presiona el botón gris, para dar inicio al programa.




72 Figura 4-9. Programa en modo automático

**Paso 4:** Encender correa transportadora

73 Figura 4-10. Correa transportadora encendida

**Paso 5:** Detener correa transportadora

74 Figura 4-11. Correa transportadora detenida

**Resultados**

El funcionamiento de la correa transportadora depende de la posición del selector.

Si el selector está en la posición 1, activa la bobina de un contactor, el cual permite o el paso de corriente al motor trifásico que realiza el movimiento de la correa trasportadora.

Si el selector está en la posición 2, desactiva la bobina del contactor, el cual corta o permite el paso de corriente al motor trifásico.

La detención de la cinta se da simplemente por desenergizar el motor trifásico, por lo cual se observa que la correa tarda un momento en detenerse completamente luego de ser desenergizado el motor, esto producto de la velocidad que mantenía en funcionamiento.

75 Figura 4-12. Correa transportadora

## 4.4 EXPERIENCIA 4 – SISTEMA POSICIONADOR DE OBJETOS

La siguiente experiencia consiste en, crear un programa de robot, para ser ejecutado en modo automático, que realice la función de tomar objetos cuadrados y redondos sobre una correa transportadora y posicionarlos en otra superficie, considerando una posición para los objetos cuadrados y otra para los objetos redondos.

Para llevar a cabo este experimento es necesario utilizar los movimientos del robot, 2 entradas y 5 salidas del controlador AX20, como se presentan a continuación:

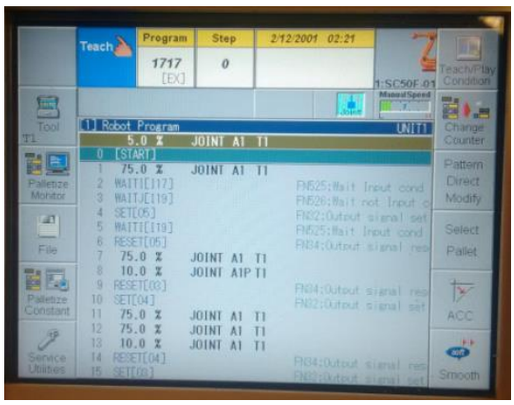
| OUTPUT | Descripción |
|---|---|
| 1 | Luz Piloto Verde |
| 2 | Luz Piloto Verde |
| 3 | Electroválvula1-01 |
| 4 | Electroválvula1-02 |
| 5 | Contactor |
| **INPUT** | **Descripción** |
| 17 | Selector posición 1 (luz piloto verde) |
| 19 | Sensor fotoeléctrico |

26 Tabla 4-4. Salidas y entradas experiencia 4

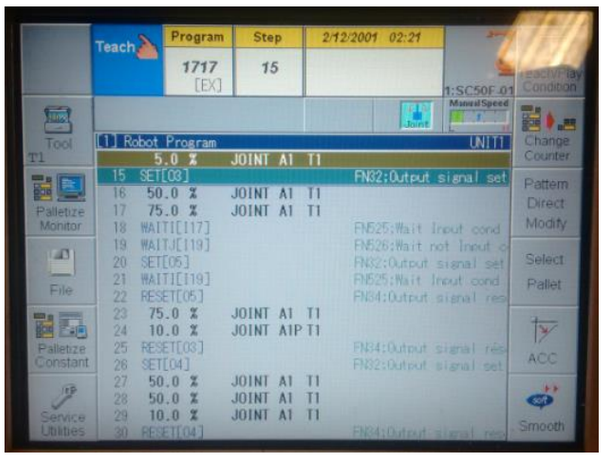
**Paso 1:** Encender el Controlador AX20 y configurar para trabajar en modo manual, al igual que el teach pendant.

**Paso 2:** Crear programa

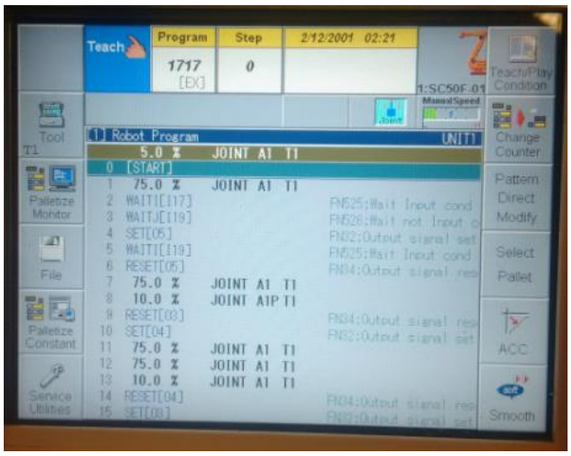
Para la creación del programa es necesario confirmar que los sensores y el actuador están conectados en las entradas y salida definida(s). Para confirmar, revisar EXPERIMENTO 1 Y 2.




76 Figura 4-13. Creación de programa experiencia 4




77 Figura 4-14. Creación de programa experiencia 4




78 Figura 4-15. Creación de programa experiencia 4

**Paso 3:** Correr programa en modo automático

Para correr el programa creado en modo automático, se configura el controlador AX20 para trabajar en modo automático, al igual que el teach pendant. Luego se energizan los servomotores y en el panel de control, se presiona el botón gris, para dar inicio al programa.

**Paso 4:** Posicionar robot

El posicionamiento del robot al inicio del programa, consiste en definir una posición inicial para el robot siempre iniciar un ciclo desde esa posición.

79 Figura 4-16. Robot en posición inicial

**Paso 5:** Iniciar posicionamiento de objeto cuadrado

Para iniciar el posicionamiento de objetos, se posiciona un objeto cuadrado en el punto A de la correa transportadora.

80 Figura 4-17. Objeto cuadrado posicionado

**Paso 6:** Posteriormente, se configura el selector en la posición 1, lo que activa la correa transportadora. Y luego se regresa a su posición 0.

81 Figura 4-18. Selector en pos 1 y pos 0

**Paso 7:** El sensor fotoeléctrico detecta la presencia del objeto cuadrado, lo que detiene la correa transportadora. El robot posiciona la herramienta (ventosa) sobre el objeto cuadrado, a una altura definida (zona segura).

82 Figura 4-19. Robot posicionado sobre objeto cuadrado

**Paso 8:** El robot posiciona la herramienta en el objeto cuadrado y se activa la ventosa para succionar el objeto.

83 Figura 4-20. Robot posicionado en el objeto cuadrado

**Paso 9:** El robot posiciona la herramienta con el objeto cuadrado a una altura definida (zona segura).

84 Figura 4-21. Robot con objeto cuadrado en zona segura

**Paso 10:** El robot posiciona la herramienta con el objeto cuadrado sobre la mesa clasificadora de objetos a una altura definida (zona segura) y en la posición del objeto cuadrado.

85 Figura 4-22. Robot con objeto cuadrado en zona segura mesa clasificadora

**Paso 11:** El robot posiciona la herramienta con el objeto en la posición del objeto cuadrado y se desactiva la ventosa para soltar el objeto.

86 Figura 4-23. Robot posiciona objeto cuadrado en mesa clasificadora

**Paso 12:** El robot posiciona la herramienta (ventosa) sobre el objeto cuadrado, a una altura definida (zona segura).

87 Figura 4-24. Robot en zona segura mesa clasificadora objeto cuadrado

**Paso 13:** El robot regresa a su posición inicial.

88 Figura 4-25. Robot en posición inicial

**Paso 14:** Iniciar posicionamiento de objeto redondo

Para iniciar el posicionamiento de objetos, se posiciona un objeto redondo en el punto A de la correa transportadora.

89 Figura 4-26. Objeto redondo posicionado

**Paso 15:** Posteriormente, se configura el selector en la posición 1, lo que activa la correa transportadora. Y luego se regresa a su posición 0.

90 Figura 4-27. Selector en pos 1 y pos 0

**Paso 16:** El sensor fotoeléctrico detecta la presencia del objeto cuadrado, lo que detiene la correa transportadora. El robot posiciona la herramienta (ventosa) sobre el objeto redondo, a una altura definida (zona segura).

91 Figura 4-28. Robot posicionado sobre objeto redondo

**Paso 17:** El robot posiciona la herramienta en el objeto redondo y se activa la ventosa para succionar el objeto.

92 Figura 4-29. Robot posicionado en el objeto redondo

**Paso 18:** El robot posiciona la herramienta con el objeto redondo a una altura definida (zona segura).

93 Figura 4-30. Robot con objeto redondo en zona segura

**Paso 19:** El robot posiciona la herramienta con el objeto redondo sobre una superficie a una altura definida (zona segura) y en la posición del objeto cuadrado.

94 Figura 4-31. Robot con objeto redondo en zona segura mesa clasificadora

**Paso 20:** El robot posiciona la herramienta con el objeto en la posición del objeto redondo y se desactiva la ventosa para soltar el objeto.

95 Figura 4-32. Robot posiciona objeto redondo en mesa clasificadora

**Paso 21:** El robot posiciona la herramienta (ventosa) sobre el objeto redondo, a una altura definida (zona segura).

96 Figura 4-33. Robot en zona segura mesa clasificadora objeto redondo

**Paso 22:** El robot regresa a su posición inicial.

97 Figura 4-34. Robot en posición inicial

**Resultado**

El funcionamiento del robot y sus componentes fue el esperado, puesto que se cumplió a cabalidad con cada movimiento y acción a realizar, considerando el paso a paso del programa y las medidas de seguridad que correspondían.

## 4.5 EXPERIENCIA 5 – USER TASK

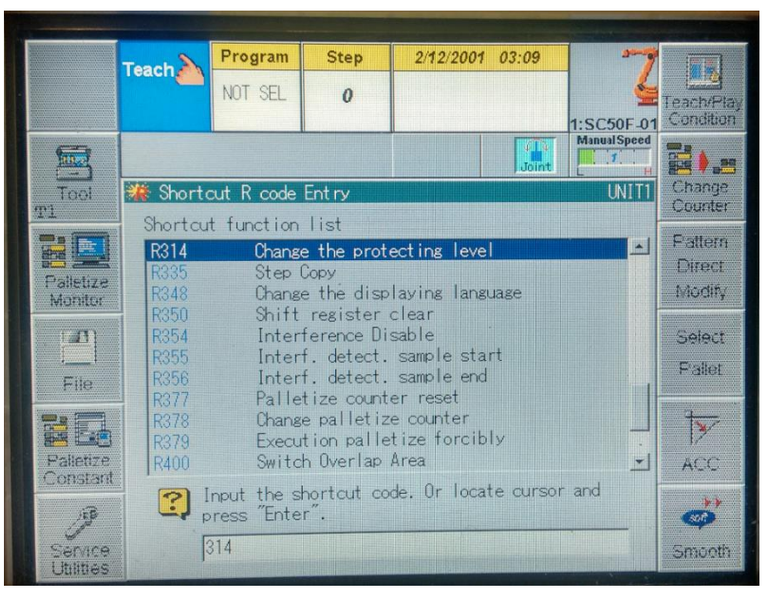
La siguiente experiencia consiste en, crear un programa de robot en un editor de texto (Notepad), y luego establecer comunicación FTP entre un computador personal y el controlador AX, para transferir el programa del pc al controlador. Luego de esto, compilar y convertir el programa para el lenguaje robot y así ejecutarlo desde el controlador como cualquier otro programa creado en el controlador.

Para llevar a cabo este experimento es necesario conectar un cable Ethernet, entre un pc y el controlador AX20.

**Paso 1:** Encender el Controlador AX20 y configurar para trabajar en modo manual, al igual que el teach pendant.

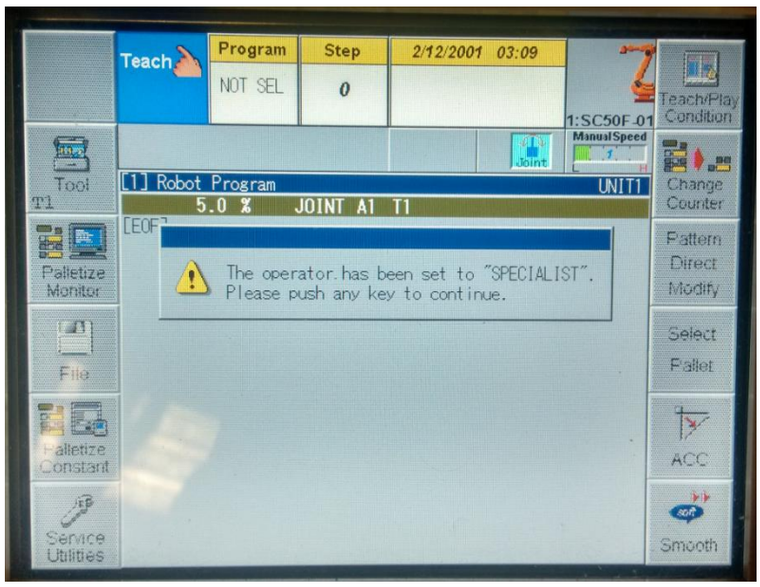
**Paso 2:** Configurar nivel de usuario como Especialista

Para configurar el nivel de usuario como especialista es necesario presionar el botón RESET y luego ingresar el número 314.




98 Figura 4-35. Cambiar nivel de protección R314

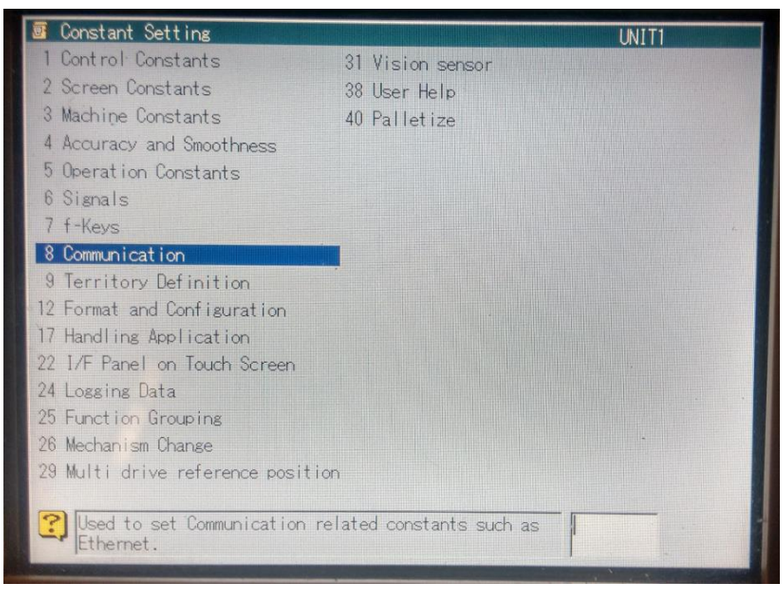
Posteriormente, se ingresa la contraseña de ESPECIALISTA que es: 12345.




99 Figura 4-36. Seteo de nivel de usuario como Especialista

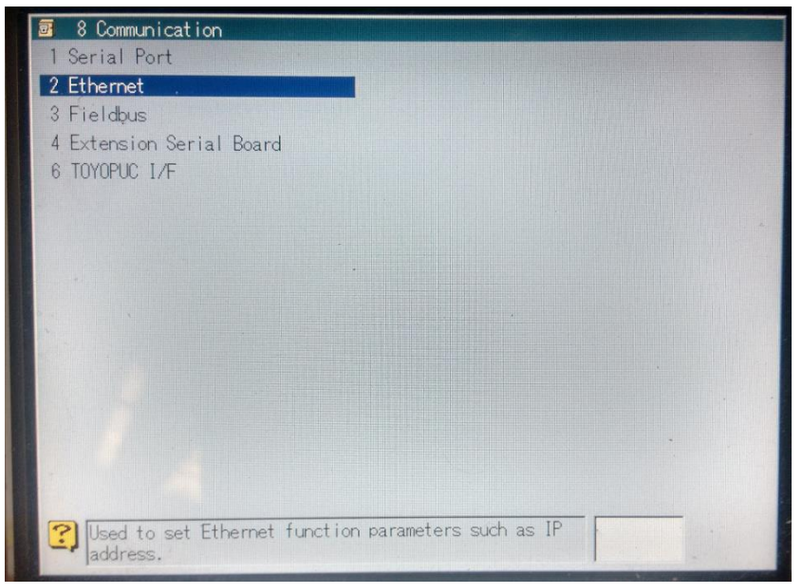
**Paso 3:** Configurar comunicación Ethernet controlador AX

Para seleccionar y configurar la comunicación Ethernet, es necesario presionar ENABLE +F5(Constant Setting). Y luego seleccionar Communication, presionando la tecla 8 y luego ENTER.



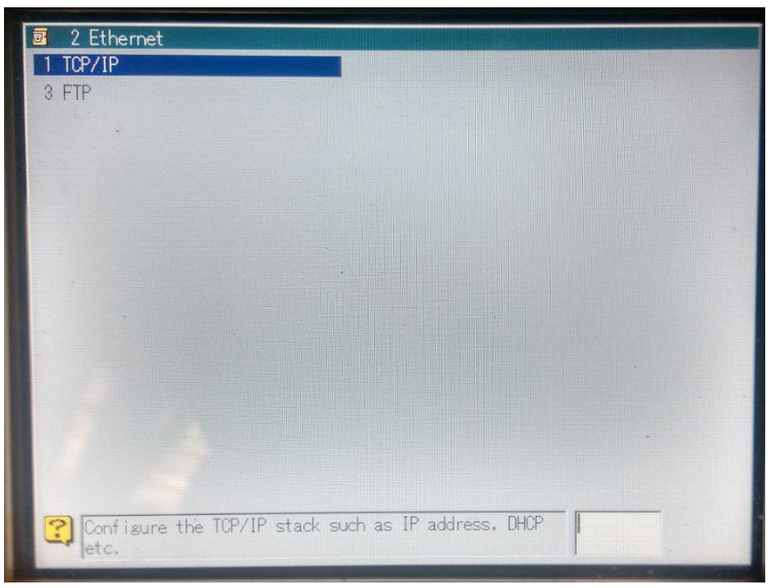

100 Figura 4-37. Comunicación 8

**Paso 4:** Seleccionar Ethernet, presionando la tecla 2 y luego ENTER



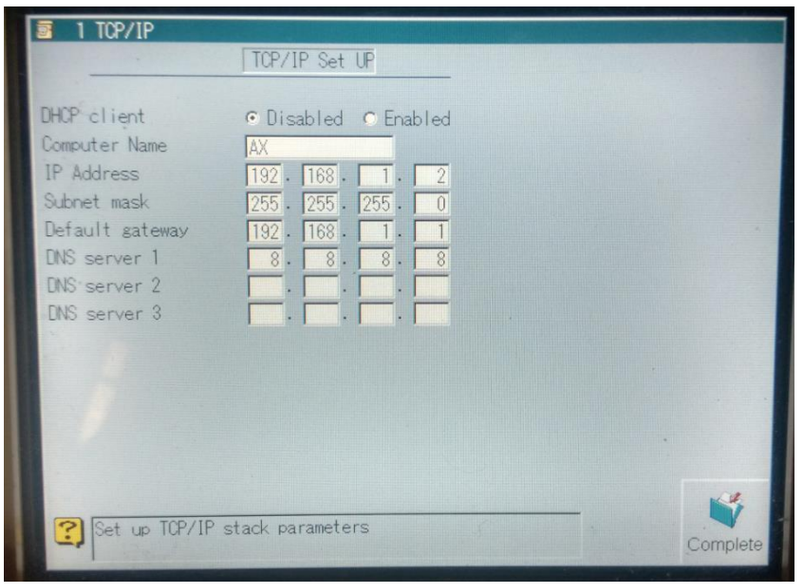

101 Figura 4-38. Ethernet 2

**Paso 5:** Seleccionar TCP/IP, presionando la tecla 1 y luego ENTER




102 Figura 4-39. TCP/IP 1

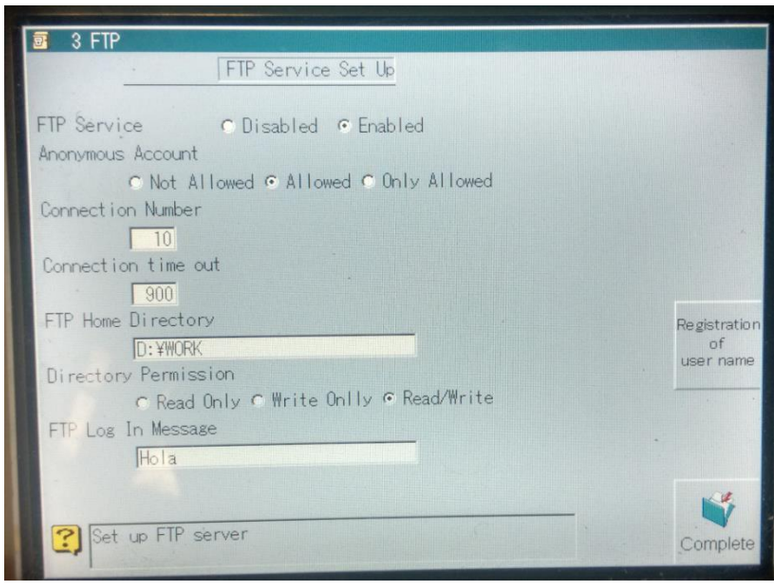
**Paso 6:** TCP/IP: Configuración de parámetros




103 Figura 4-40. Configuración TCP/IP 1

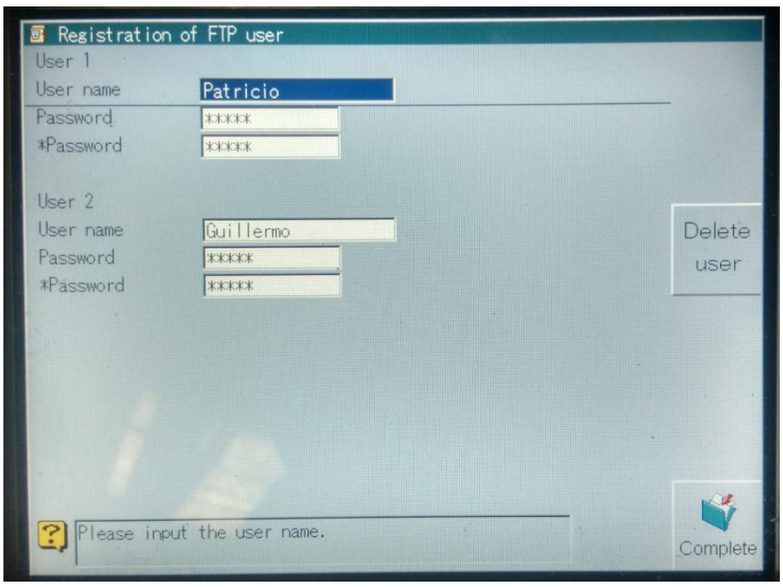
**Paso 7:** Seleccionar FTP, presionando la tecla 3 y luego ENTER

FTP: Configuración de parámetros




104 Figura 4-41. Configuración FTP 3

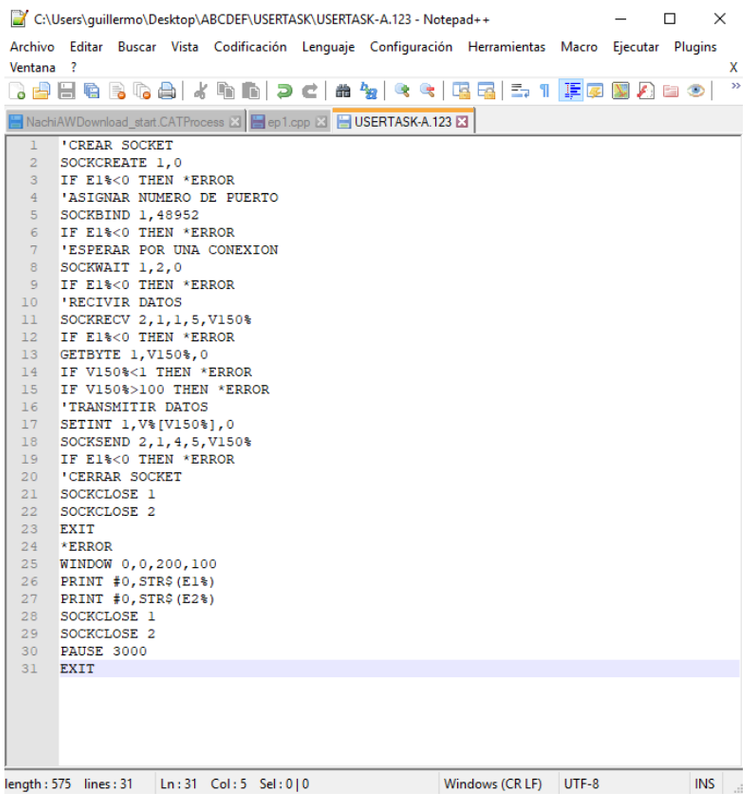
**Paso 8:** Registrar un nombre de usuario




105 Figura 4-42. Registro de usuario

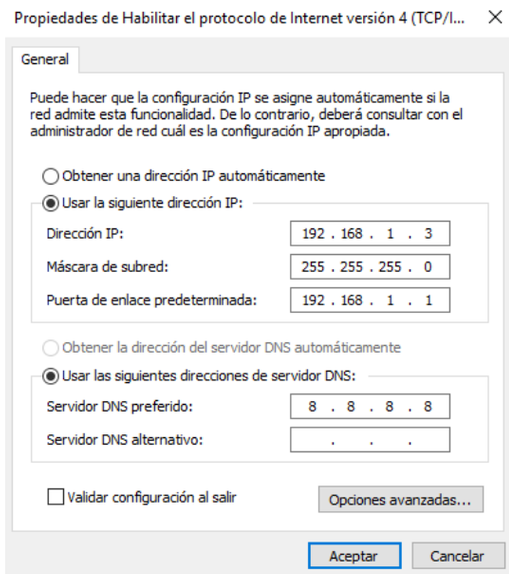
**Paso 9:** Crear programa

Parar apreciar con mayor detalle el programa, revisar el Anexo A

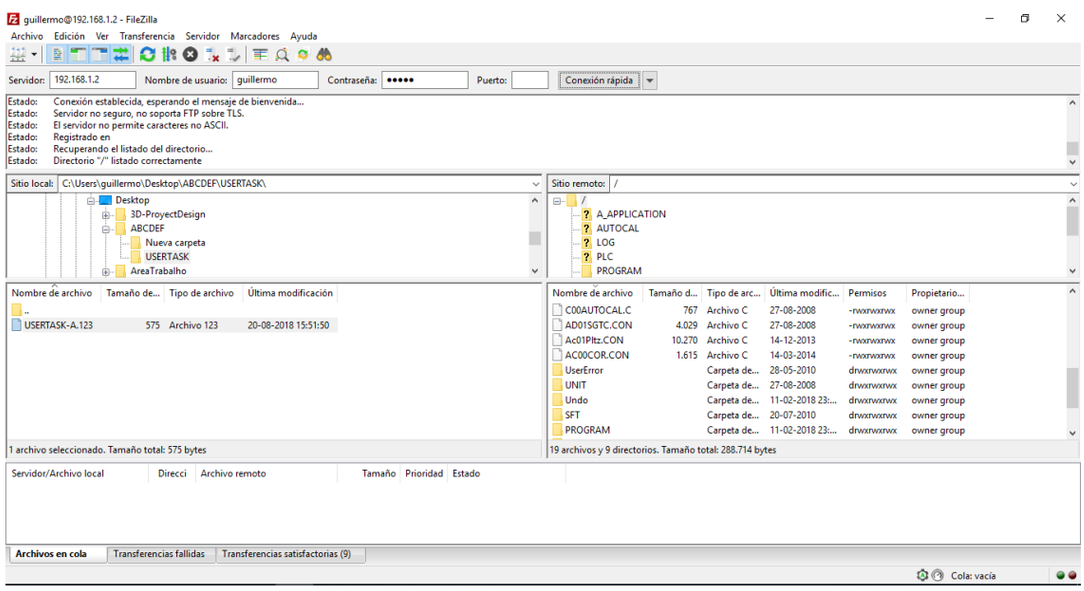



106 Figura 4-43. Programa USERTASK-A.123

**Paso 10:** Configurar comunicación Ethernet computador personal,

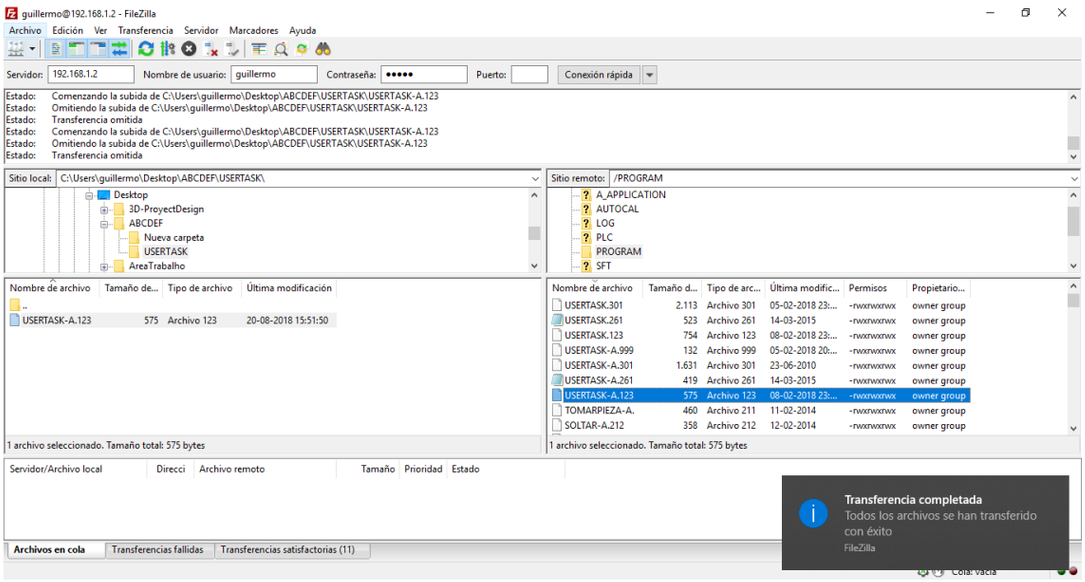



107 Figura 4-44. Propiedades IPV4 computador personal




108 Figura 4-45. Filezila conectado al controlador AX20

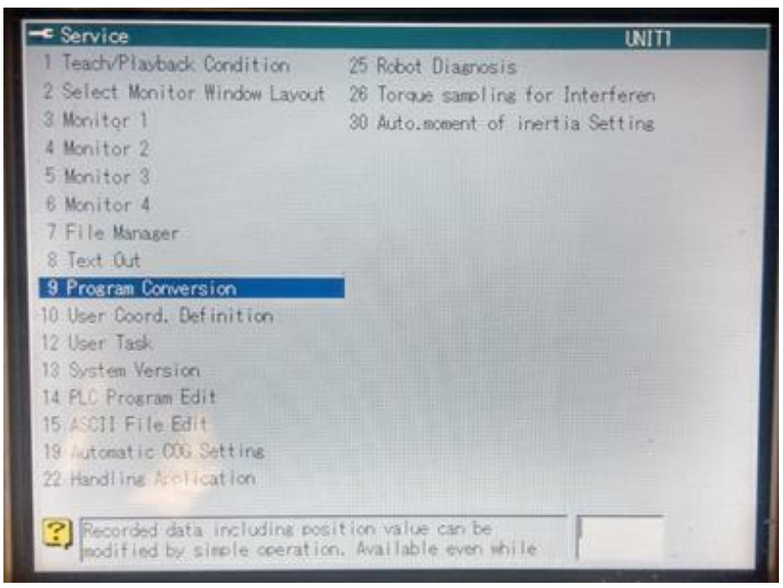
**Paso 11:** Transferir programa USERTASK-A.123 al controlador AX20




109 Figura 4-46. Transferencia de programa

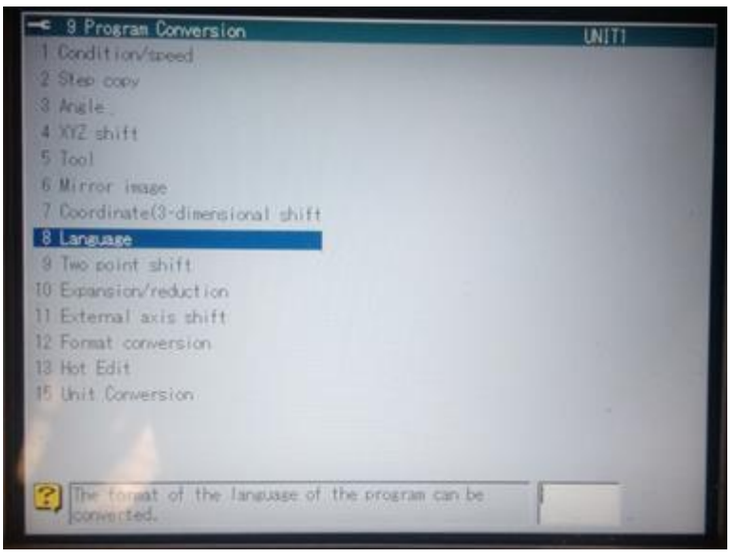
**Paso 12:** Compilar y convertir programa USERTASK-A.123 en controlador AX20.

Para compilar y convertir el programa es necesario presionar ENABLE + F5(Constant setting). Luego presionar 9 y ENTER.



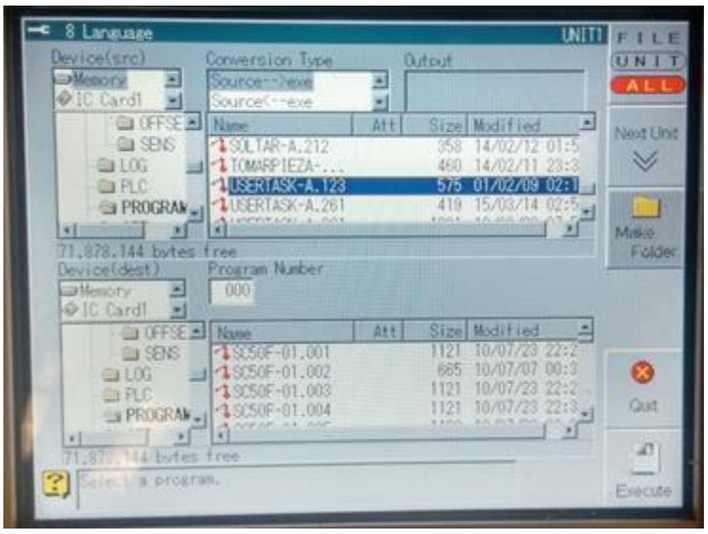

110 Figura 4-47. Conversión de programa 9

**Paso 13:** Posteriormente, seleccionar Lenguage, presionando 8 y luego ENTER.



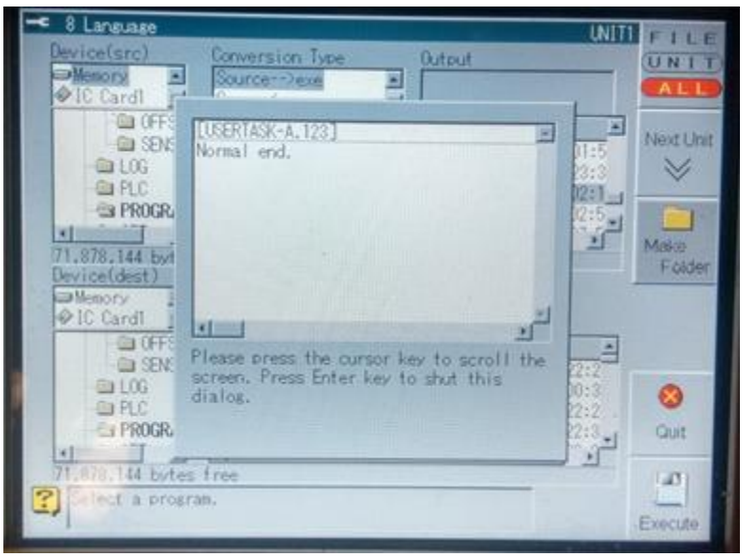

111 Figura 4-48. Lenguaje 8

**Paso 14:** Se selecciona el programa USERTASK-A.123 presionando ENTER, posteriormente se presiona F12 (EXECUTE).



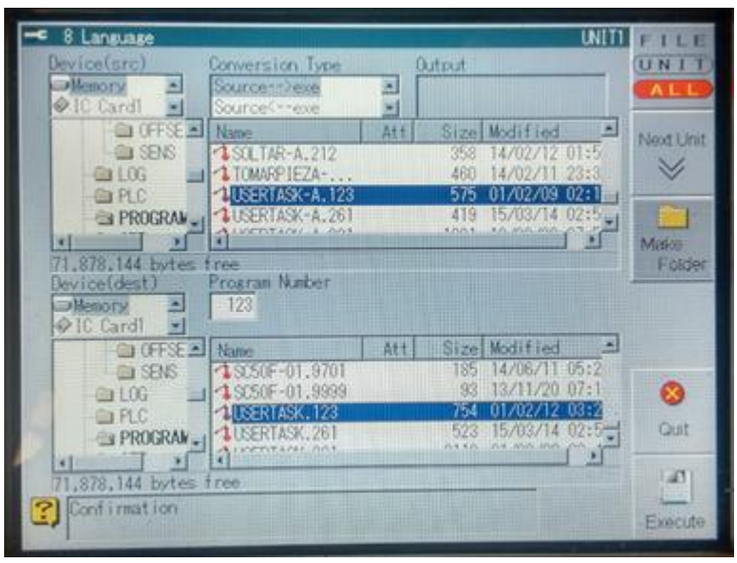

112 Figura 4-49. Selección de programa a convertir

**Paso 15:** Programa compilado y convertido al lenguaje del robot con éxito.




113 Figura 4-50. Conversión de programa finalizada

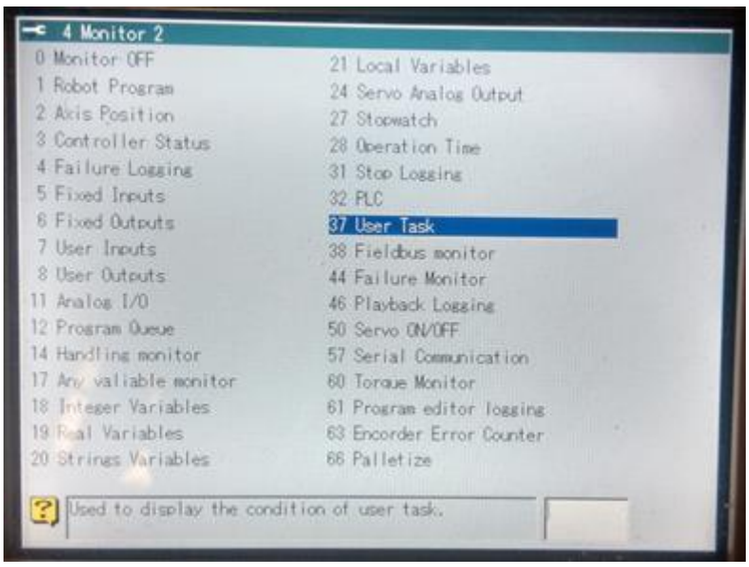
**Paso 16:** Programa USERTASK-A.123 en formato programa robot USERTASK.123




114 Figura 4-51. Programa convertido

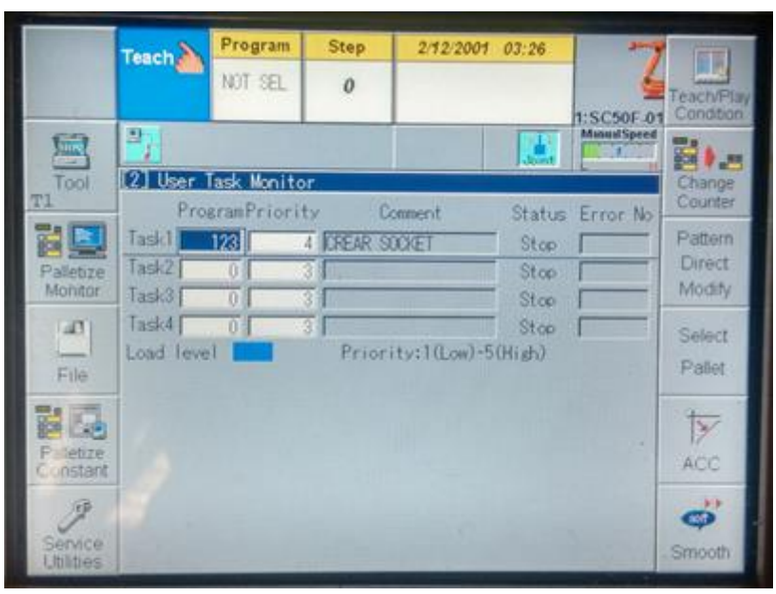
**Paso 17:** Correr el programa USER TASK

Para correr el programa User Task creado de forma manual, se presiona ENABLE + F2 (Monitor2). Luego, se selecciona 37 User Task. Ingresando el numero 37 y luego ENTER para confirmar.



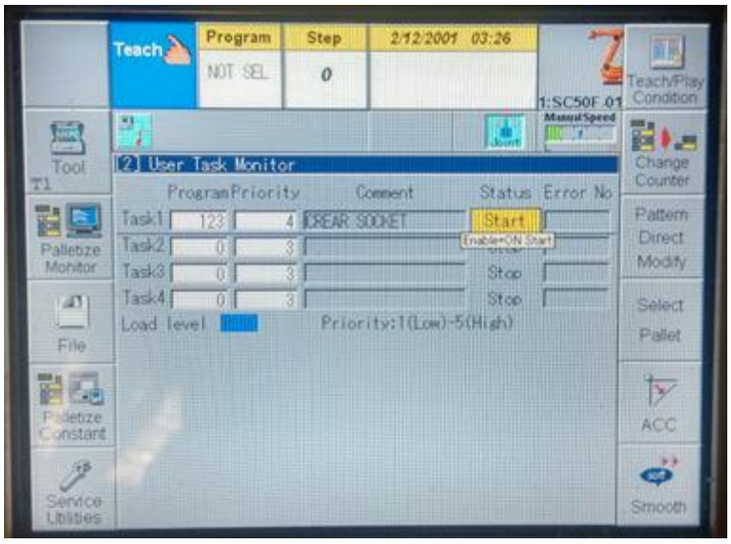

115 Figura 4-52. User Task 37

**Paso 18:** Posteriormente se posiciona el cursor sobre el Task 1(puede ser cualquiera de los cuatro Task disponibles). Luego, se presiona ENABLE + EDIT y se ingresa el número del programa. Automáticamente la casilla Comment se muestra un comentario del programa seleccionado.




116 Figura 4-53. Selección del programa User Task

**Paso 19:** Para iniciar el programa, se posiciona el cursor sobre Stop y luego se presiona ENABLE + ON, y el programa es iniciado, indicando esta acción con la palabra Start y una luz amarilla en la casilla de esta palabra.




117 Figura 4-54. User Task iniciado

**Resultado**

La realización de este experimento permite aplicar conocimientos de redes de comunicación Cliente Servidor. Creando programas en un computador personal y transfiriendo estos al controlador AX20. Luego compilando, convirtiendo y ejecutando estos en el controlador AX20.

Estas funciones son usadas cuando se requiere trabajar operaciones más complejas con el manipulador industrial, permitiendo desarrollar programas de movimiento de robot y/o User Tasks como el del experimento, que son programas que se desarrollan para trabajar en paralelo a un programa de movimiento de robot, realizando operaciones matemáticas, manipulación de variables y/o funciones de comunicación.

## 4.6 EXPERIENCIA 6 – SISTEMA DE VISIÓN ARTIFICIAL

El objetivo del experimento consiste en desarrollar un sistema de visión artificial que detecte y obtenga la posición de objetos cuadrados y redondos sobre una correa transportadora. Es necesario haber realizado los pasos de instalación y configuración inicial del hardware a utilizar (raspberry pi con el modulo cámara). Además, se requiere tener dichos componentes bien posicionados sobre la estructura del sistema de visión, definir la altura ideal, entre la cámara y la superficie donde estarán posicionados los objetos. Y tener una iluminación ambiente lo más estable posible.

A continuación se detalla paso a paso la programación desarrollada y los resultados obtenidos con pruebas realizadas.

**Paso 1:** Declaración de librerías

```
from picamera import PiCamera    # Libreria para funciones del modulo camara
from time import sleep           # Liibreria para funciones de tiempo
import cv2                       # Libreria para funciones de OpenCV
import numpy as np               # Libreria para funciones matematicas
```

118 Figura 4-55. Declaración de librerias

**Paso 2:** Configuración de sistema

```
camara = PiCamera()              # Objeto de Libreria

camara.rotation = 180            # Rotacion de la camara 180 grados
camara.resolution = (640, 480)   # Resolucion de la camara 640x480 pixeles
```

119 Figura 4-56. Configuración de sistema

**Paso 3:** Capturar imagen

```
#3.1. Capturar Imagen
camara.capture("Objeto.jpg")
```

120 Figura 4-57. Capturar imagen

**Paso 4:** Leer imagen

```
#3.2. Leer Imagen
img1 = cv2.imread('Objeto.jpg')
```

121 Figura 4-58. Leer imagen

**Paso 5:** Transformar Imagen a Color a Escala de grises

```
#3.3. Transformar Imagen a Color a Escala de grises
imgra = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
```

122 Figura 4-59. Imagen en escala de grises

**Paso 6:** Transformar Imagen Escala de grises a Binaria

```
#3.4. Transformar Imagen Escala de grises a Binaria
thresh = cv2.adaptiveThreshold (imgray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11,2)
```

123 Figura 4-60. Imagen binarizada

**Paso 7:** Filtrar ruidos de la Imagen

```
#3.5. Filtrar ruidos de la Imagen
#...Dilate...
#Crear un kernel de '1' de 3x3
kernel = np.ones((3,3),np.uint8)
# Se aplica la transformacion: Dilate
thre = cv2.dilate(thresh,kernel,iterations = 1)
```

124 Figura 4-61. Aplicación de filtros en imagen

**Paso 8:** Encontrar contornos

```
#3.6. Encontrar contornos
contours,h = cv2.findContours(thre,cv2.RETR_CCOMP,cv2.CHAIN_APPROX_SIMPLE)
cnt = contours[0]
```

125 Figura 4-62. Encontrar contornos

**Paso 9:** Obtener centroide

```
#3.7. Obtener centroide
M = cv2.moments(cnt)
cX = int(M["m10"] / M["m00"])
cY = int(M["m01"] / M["m00"])
print("POS X")
print( cX)
print("POS Y")
print( cY)
```

126 Figura 4-63. Obtener centroide

**Paso 10:** Dibujar contornos y centro del objeto

```
#3.8. Dibujar contornos y centro del objeto
cv2.drawContours(img1, contours, 0, (0,255,0), 3)
cv2.circle(img1, (cX, cY), 3, (0, 0, 255), -1)
```

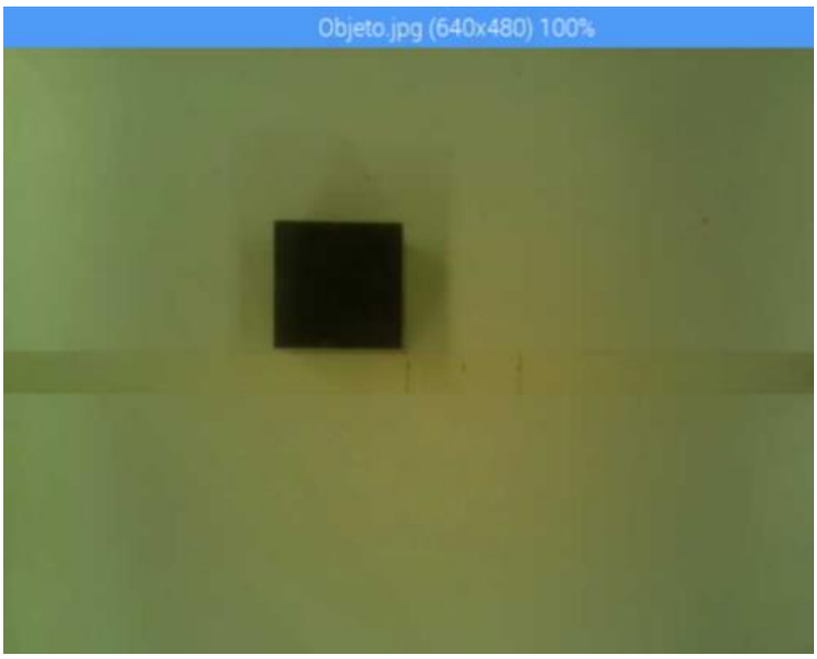
127 Figura 4-64. Dibujar contornos y centroide en imagen

Para ver el código completo, revisar Anexo B.Programa PROX_V7.py

**Resultados**

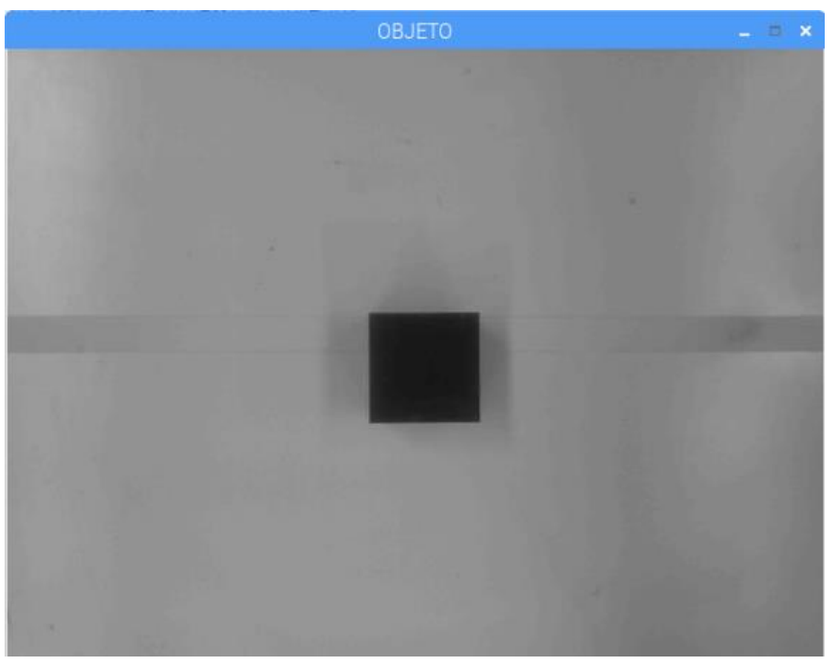
Los resultados que se presentan a continuación, se obtuvieron considerando diferentes opciones de hardware y software.

**Resultado 1: Capturar de imagen**



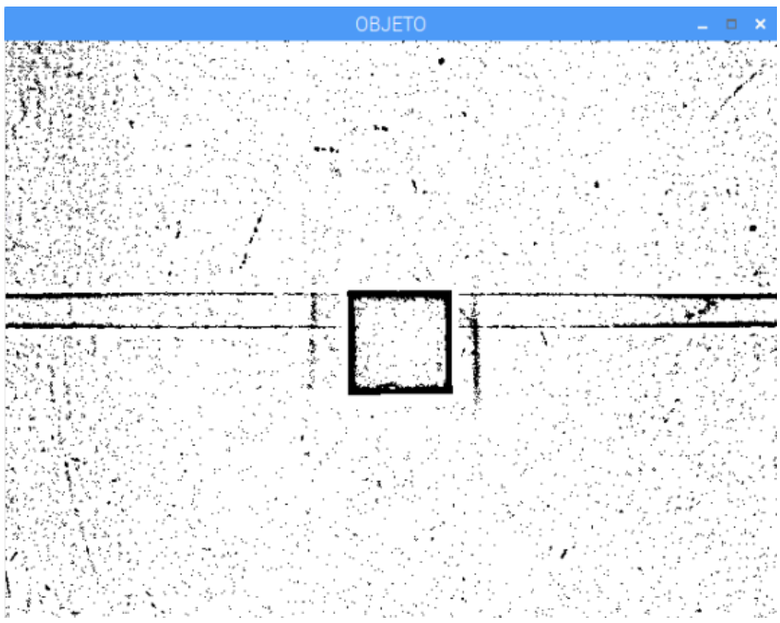

128 Figura 4-65. Resultado de imagen capturada

**Resultado 2: Imagen en escala de grises**



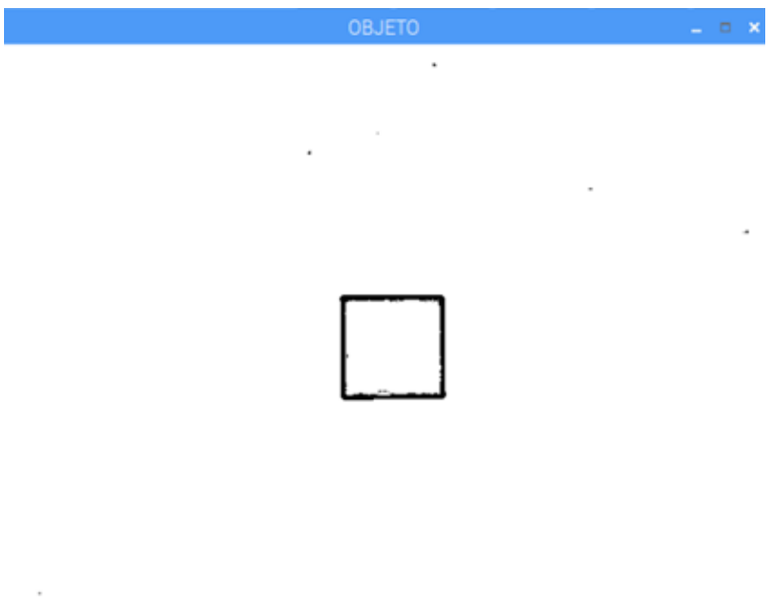

129 Figura 4-66. Resultado de imagen en escala de grises

**Resultado 3: Imagen binaria**




130 Figura 4-67. Resultado de imagen binaria

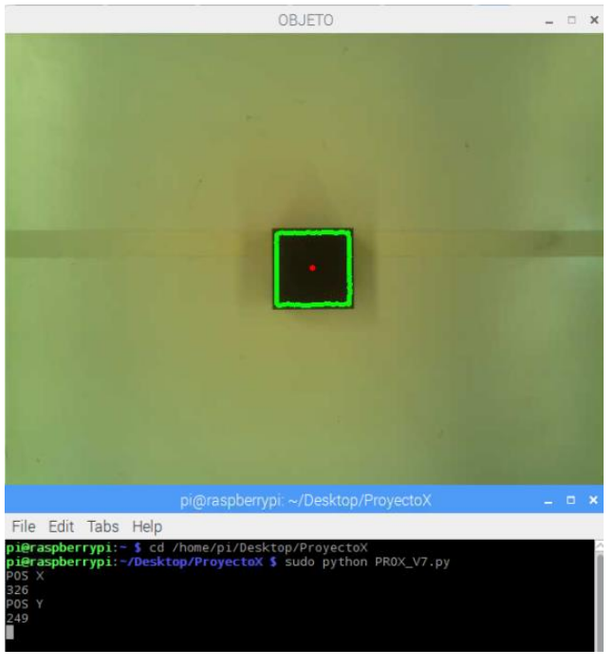
**Resultado 4: Imagen binaria con filtros**




131 Figura 4-68. Resultado de imagen con filtros

## 4.7 RESULTADOS FINALES

Resultado final 1: Objeto Cuadrado 1

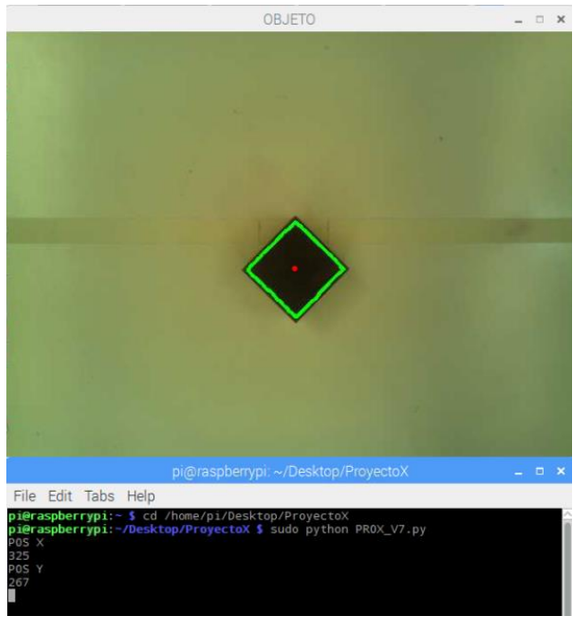



132 Figura 4-69. Resultado Objeto Cuadrado 1

Resultado final 2: Objeto Cuadrado 2

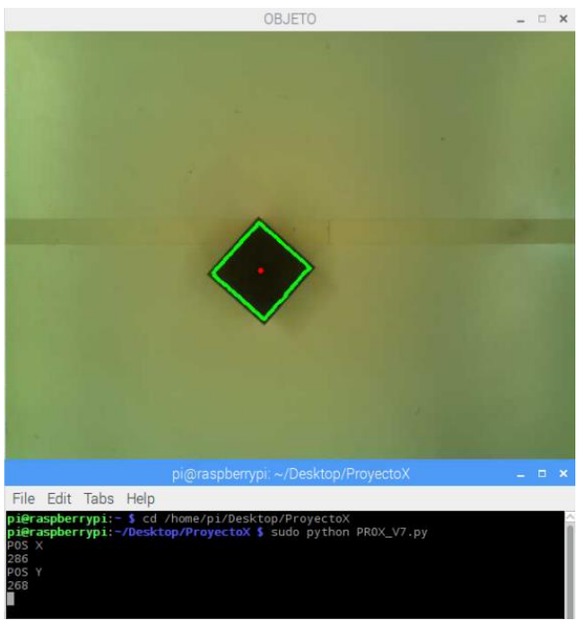
133 Figura 4-70. Resultado Objeto Cuadrado 2

Resultado final 3: Objeto Cuadrado 4



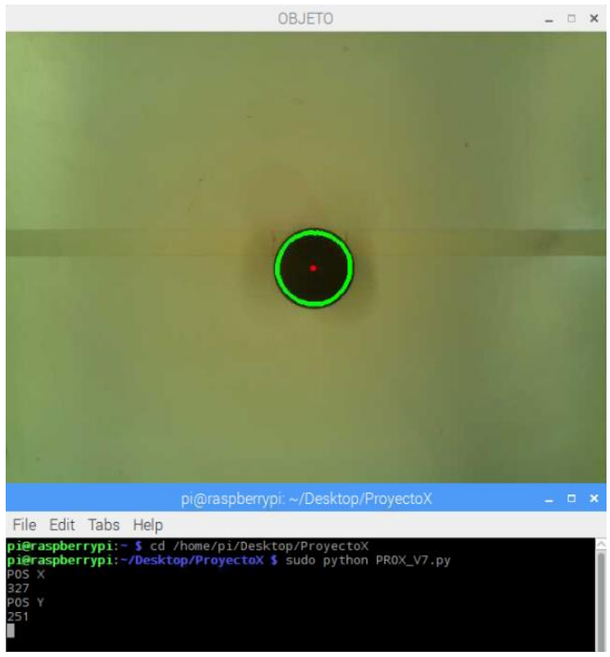

134 Figura 4-71. Resultado Objeto Cuadrado 4

Resultado final 4: Objeto Cuadrado 5



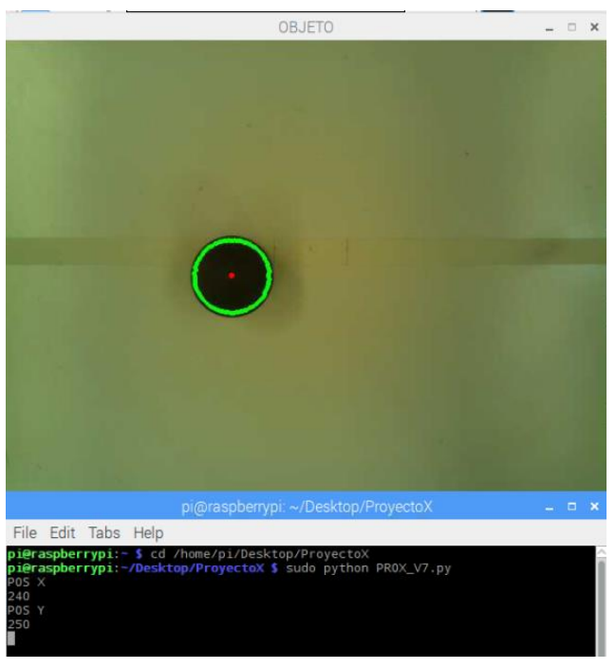

135 Figura 4-72. Resultado Objeto Cuadrado 5

Objeto Redondo 1




136 Figura 4-73. Resultado Objeto Redondo 1

Objeto Redondo 2




137 Figura 4-74. Resultado Objeto Redondo 2

# CONCLUSIONES Y RECOMENDACIONES

- El controlador AX20 y el manipulador robótico SC50F, tienen un cierto grado de complejidad considerable, por lo que para utilizar y trabajar con estos dispositivos se hace sumamente necesario leer y comprender la documentación correspondiente a estos equipos y su funcionamiento.
- Por otro lado para la operación del sistema robótico, se requiere tomar especial cuidado y considerar las medidas de seguridad correspondientes a cada operación que se realice.
- El robot SC50F, para saber la posición de cada uno de sus ejes, utiliza enconders, los cuales son energizados con unas baterías de litio. Si el voltaje de estas baterías es bajo, el robot puede presentar errores y un mal funcionamiento.
- Para la conexión y desconexión de sensores y actuadores, sobre la placa I/O, es necesario tener en cuenta los voltajes de operación de la placa y los elementos externos. Además, de la configuración de salida que estos dispositivos poseen, que puede ser NPN o PNP. Para obtener una mayor información sobre esto, existen diferentes diagramas de conexiones del robot y su controlador, con sus diferentes periféricos.
- La raspberry pi 3 modelo b y su módulo de cámara, son dispositivos de un gran potencial, sin embargo, el funcionamiento de estos puede ser afectado por la estática y/o por un mal uso al momento de realizar conexiones. Además, al trabajar con el módulo cámara es necesario disponer de una iluminación ambiente estable para obtener buenos resultados.
- A pesar de que tanto el controlador AX20 y la raspberry pi incorporan un socket para la comunicación Ethernet, para conseguir establecer comunicación entre ambos, es necesario revisar la documentación existente relacionada con esto. Además, realizar prácticas, tales como, programas y operaciones básicas de comunicación Ethernet, a modo de entender y ambientarse en esta área, para luego pasar a desarrollar la comunicación Ethernet entre el controlador AX20 y las raspberry pi.

# BIBLIOGRAFÍA

[1] Robot industrial - https://es.wikipedia.org/wiki/Robot_industrial

[2] Visión artificial - https://es.wikipedia.org/wiki/Visi%C3%B3n_artificial

[3] The British Machine Vision Association and Society for Pattern Recognition - http://www.bmva.org/

[4] IFR – International Federation of Robotics - https://ifr.org/service-robots/

[6] Raspbian - https://es.wikipedia.org/wiki/Raspbian

[7] Python - https://es.wikipedia.org/wiki/Python

[8] OpenCV - https://es.wikipedia.org/wiki/OpenCV

[9] Comenzando con Pi Camera - https://projects.raspberrypi.org/en/projects/getting-started-with-picamera

# ANEXOS

Anexo A.Programa USERTASK-A.123

```
'CREAR SOCKET
SOCKCREATE 1,0
IF E1%<0 THEN *ERROR
'ASIGNAR NUMERO DE PUERTO
SOCKBIND 1,48952
IF E1%<0 THEN *ERROR
'ESPERAR POR UNA CONEXION
SOCKWAIT 1,2,0
IF E1%<0 THEN *ERROR
'RECIVIR DATOS
SOCKRECV 2,1,1,5,V150%
IF E1%<0 THEN *ERROR
GETBYTE 1,V150%,0
IF V150%<1 THEN *ERROR
IF V150%>100 THEN *ERROR
'TRANSMITIR DATOS
SETINT 1,V%[V150%],0
SOCKSEND 2,1,4,5,V150%
IF E1%<0 THEN *ERROR
'CERRAR SOCKET
SOCKCLOSE 1
SOCKCLOSE 2
EXIT
*ERROR
WINDOW 0,0,200,100
PRINT #0,STR$(E1%)
PRINT #0,STR$(E2%)
SOCKCLOSE 1
SOCKCLOSE 2
PAUSE 3000
EXIT
```

Anexo B.Programa PROX_V7.py

```
#! /usr/bin/python
#-----------------------------------------------------------------------------------------------
# Analisis y procesamiento de Imagen - Detectar y Obtener posicion de objetos
# Proyecto Memoria de Titulo Guillermo Mendez
# Profesor Felipe Benavides P.
# S1 2018
# V7
#-----------------------------------------------------------------------------------------------
#1. DECLARACION DE LIBRERIAS
#-----------------------------------------------------------------------------------------------
from picamera import PiCamera   # Libreria para funciones del modulo camara
from time import sleep          # Liibreria para funciones de tiempo
import cv2                      # Libreria para funciones de OpenCV
import numpy as np              # Libreria para funciones matematicas
#-----------------------------------------------------------------------------------------------
#2. Configuracion de Sistema
#-----------------------------------------------------------------------------------------------
camara = PiCamera()             # Objeto de Libreria

camara.rotation = 180           # Rotacion de la camara 180 grados
camara.resolution = (640, 480) # Resolucion de la camara 640x480 pixeles
#-----------------------------------------------------------------------------------------------
#3. Programa
#-----------------------------------------------------------------------------------------------
#3.1. Capturar Imagen
camara.capture("Objeto.jpg")

#3.2. Leer Imagen
img1 = cv2.imread('Objeto.jpg')

#3.3. Transformar Imagen a Color a Escala de grises
imgra = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
imgray  = cv2.medianBlur(imgra ,5)
#3.4. Transformar Imagen Escala de grises a Binaria
thresh = cv2.adaptiveThreshold (imgray, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11,2)

#3.5. Filtrar ruidos de la Imagen
#...Dilate...
#Crear un kernel de '1' de 3x3
kernel = np.ones((3,3),np.uint8)
# Se aplica la transformacion: Dilate
thre = cv2.dilate(thresh,kernel,iterations = 1)

#3.6. Encontrar contornos
contours,h =
cv2.findContours(thre,cv2.RETR_CCOMP,cv2.CHAIN_APPROX_SIMPLE)
cnt = contours[0]
#3.7. Obtener centroide
M = cv2.moments(cnt)
```

```
cX = int(M["m10"] / M["m00"])
cY = int(M["m01"] / M["m00"])
print("POS X")
print( cX)
print("POS Y")
print( cY)
#3.8. Dibujar contornos y centro del objeto
cv2.drawContours(img1, contours, 0, (0,255,0), 3)
cv2.circle(img1, (cX, cY), 3, (0, 0, 255), -1)

cv2.imshow('OBJETO', img1)        # Mostrar imagen

camara.close()                    # Cerrar camara
cv2.waitKey(0)                    # Espera presionar tecla
cv2.destroyAllWindows()           # Cierra todas las Imagenes
```