

CONTROL DE ROBOT CARTESIANO MEDIANTE PLC S7-1200 DE SIEMENS Y ARDUINO PARA IMPRESIÓN 3D

Alberto Portero, Jesús Lozano, Santiago Salamanca
Escuela de Ingenierías Industriales, Universidad de Extremadura
Av. Elvas s/n. Badajoz. Email: jesuslozano@unex.es

Resumen

Se presenta en este trabajo un sistema de control de un robot cartesiano, mediante la comunicación entre distintos dispositivos controladores, para su funcionamiento como impresora 3D. El sistema de control está basado en un autómata programable S7-1200 de Siemens, una tarjeta Arduino Mega 2560 y un módulo Ethernet Shield 2.0. El programa de control se ha realizado a través de los entornos de desarrollo Arduino IDE y TIA Portal v13. La comunicación entre la tarjeta Arduino y el autómata programable S7-1200 se ha realizado mediante conexión Ethernet, a través del protocolo de comunicación UDP (User Datagram Protocol). Se ha diseñado el sistema de manera que el programa de control lea un archivo G-Code (modelo 3D), lo interprete, y ejecute las instrucciones de movimiento del robot cartesiano. El programa de control no contempla la posibilidad de extrusión, y se centra en el control del movimiento de los ejes y en la comunicación entre dispositivos.

Palabras Clave: robot cartesiano, autómata programable, Arduino, impresora 3D, protocolo UDP, PLC.

1 INTRODUCCIÓN

Una impresora 3D es una máquina capaz de imprimir figuras con volumen (ancho, alto

y largo) a partir de un modelo 3D realizado por ordenador.

El proceso de impresión tridimensional consiste fundamentalmente en crear un prototipo, capa a capa, de manera ascendente. Para ello, la máquina deposita una capa de plástico a una temperatura preestablecida, que se compacta, y repite el proceso hasta completar la pieza. A esta tecnología se le denomina FFF (Fabricación con Filamento Fundido).



Figura 1: Impresora 3D de Tecnología FFF.

El proceso de diseño 3D mediante la tecnología FDM (Fused Deposition Modeling), consiste en el modelado 3D de la pieza. Este modelo 3D sigue un proceso de conversión de formato de archivo, antes de la impresión de la pieza: en primer lugar, se realiza el diseño del modelo 3D en formato de archivo CAD (Diseño Asistido por Ordenador); en segundo lugar, se

exporta a un formato de archivo que consiste en una malla de triángulos cerrada, que define la geometría de objetos 3D, excluyendo información como color, texturas o propiedades físicas, como pueden ser: STL, OBJ, STEP, WRML, entre otros; finalmente, se exporta a un formato de archivo G-Code, el cual puede reconocer la impresora 3D.

El formato G-Code es un lenguaje de descripción de operaciones, concebido en origen para máquinas de control numérico por ordenador (CNC) y, actualmente, muy utilizada en impresoras 3D. El lenguaje G-Code describe el movimiento y las diferentes operaciones que la máquina CNC o impresora 3D debe realizar para la fabricación de la pieza.

Si tenemos en cuenta que un robot industrial, según la definición de la Asociación de Industrias Robóticas (RIA), es un manipulador multifuncional reprogramable, capaz de mover piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas [1], se puede manipular para que su funcionamiento sea igual al de una impresora 3D.

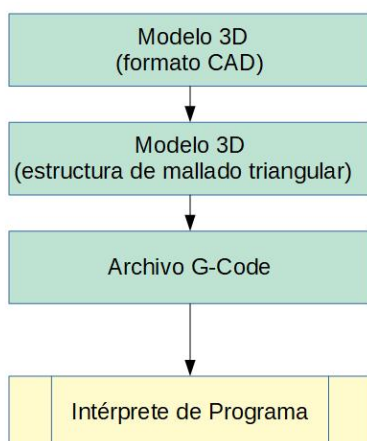


Figura 2: Proceso de conversión de formatos 3D.

Como punto de partida, el control del robot cartesiano se realiza a través de una interfaz

HMI (sistema SCADA), pero no tiene asignada ninguna aplicación. Por ello, se decidió implementar un sistema de control que realice el funcionamiento de una impresora 3D. Para ello, para el desarrollo del sistema se ha decidido utilizar un Arduino Mega 2560 y un módulo de comunicaciones Ethernet Shield 2.0, y un autómatas programable S7-1200 de Siemens, integrados en una red de área local. La arquitectura Cliente-Servidor, nos proporciona la comunicación entre los distintos dispositivos, a través del protocolo UDP (User Datagram Protocol), cuyo modelo de referencia es el modelo TCP/IP (Protocolo de Control de Transmisión / Protocolo de Internet).

2 DESCRIPCIÓN DEL SISTEMA

2.1 ROBOT CARTESIANO

El robot cartesiano está formado por tres ejes (X, Y, Z) de la marca Berger Lahr (actualmente Schneider Electric Motion Deutschland), de longitudes 404 mm, 394 mm y 308 mm, respectivamente, y se muestra en la Figura 3. Los motores utilizados para el movimiento del robot son del tipo paso a paso, cuyo modelo para el eje X e Y es el VDRM31117/50LWCOO y, para el eje Z, el modelo VDRM3910/50LWCOB, ambos de la marca Berger Lahr. Además, para la alimentación de los motores paso a paso, se han utilizado los drivers de potencia TLD011F para el motor Z y TLD012 para los motores X e Y, de la marca Twin Line (Berger Lahr) [2].

La conexión con los motores se realiza a través de al interfaz PULSE-C, a 24 V. Los drivers están alimentados a 24 V y a 230 V. El robot cartesiano posee dos finales de carrera colocados en cada eje, y los modelos utilizados son el BES 526-300-S303-C-05 para los ejes X e Y, y el BES 526-3009-SA2-MO-C-10 para el eje Z, ambos de la marca Balluff.

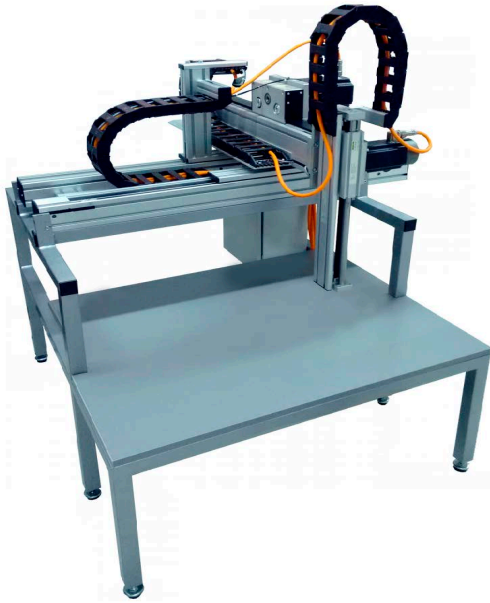


Figura 3: Robot Cartesiano.

El programa de control del robot cartesiano se ha implementado en el autómata programable S7-1200 (CPU 1214C DC/DC/DC) de Siemens.

2.2 SIEMENS S7-1200

La unidad de control que se ha utilizado, como hemos comentado anteriormente, es el autómata programable S7-1200 (CPU 1214C DC/DC/DC) de Siemens, como se puede ver en la Figura 4, cuyas salidas por transistor pueden proporcionar los trenes de pulso necesarios para el movimiento de los motores paso a paso.

Dispone de diez salidas, catorce entradas digitales y dos entradas analógicas. La CPU incorpora un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida, PROFINET integrado, E/S de control de movimiento de alta velocidad y entradas analógicas, todo ello incorporado en una carcasa compacta, conformando así un potente controlador.

La comunicación con el software de programación (TIA Portal v13) se realiza mediante comunicación Ethernet. Permite

programar y establecer comunicación con el protocolo TCP/IP (Transmission Control Protocol / Internet Protocol). Las características técnicas del autómata programable han sido consultadas en el Manual de producto SIMATIC S7-1200 Easy Book [3] y en el Manual de funciones SIMATIC STEP 7 S7-1200 Motion Control V13 [4]. Posee una fuente de alimentación PM 1207 de Siemens, la cual alimenta también a los drivers y a los finales de carrera.



Figura 4: Autómata Programable S7-1200 de Siemens.

2.3 ARDUINO MEGA 2560

Arduino Mega 2560 es una versión ampliada de la tarjeta original de Arduino y basada en el microcontrolador Atmega2560, y se muestra en la Figura 5. Dispone de 54 entradas y salidas, 14 de las cuales se pueden utilizar como salidas PWM (Modulación por ancho de pulso). Además, dispone de 16 entradas analógicas, 4 UARTs (puertas series), un oscilador de 16 MHz, una conexión USB, un conector de alimentación, un conector ICSP y un pulsador para reset de la placa. Se alimenta mediante conexión USB o con una fuente de alimentación externa de 12V.

El Atmega2560 tiene 256 kB de memoria flash para almacenar código (8 kB son usados para el arranque del sistema o bootloader). Además, tiene 8 kB de memoria SRAM y 4 kB de EEPROM, a la cual se puede acceder para leer o escribir con la librería EEPROM. Soporta además la comunicación I2C (TWI) y comunicación SPI, muy importante, dado que es la utilizada para la conexión con el módulo Ethernet Shield 2.0, que se verá más adelante [5].

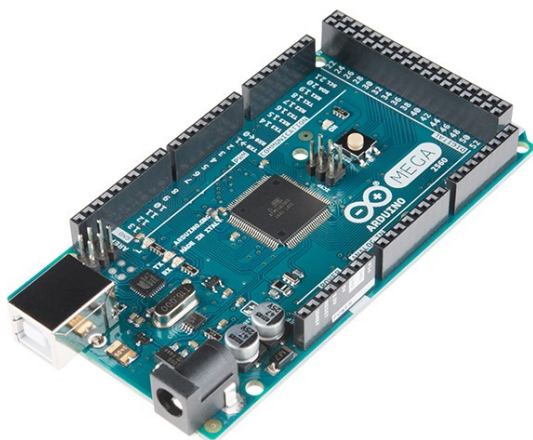


Figura 5: Tarjeta Arduino Mega 2560.

La programación del dispositivo se ha realizado en el entorno de desarrollo de Arduino (Arduino IDE, versión 1.6.1).

2.4 ETHERNET SHIELD 2.0

El módulo Ethernet Shield (versión 2.0) permite a una tarjeta Arduino conectarse a Internet, y se muestra en la Figura 6. Soporta hasta 8 conexiones de sockets simultáneas. Dispone de conectores que permiten conectar, a su vez, otras tarjetas, y apilarlas sobre la tarjeta Arduino en la que se trabaja [6].

Se basa en el chip Ethernet Wiznet W5500, que proporciona una pila de red (IP) con capacidad para TCP y UDP, con todas las funciones del estándar IEEE 802.3 (Ethernet capa física y de enlace de datos). Provee un conector Ethernet estándar RJ45

para la conexión con el autómatas programable S7-1200 de Siemens.

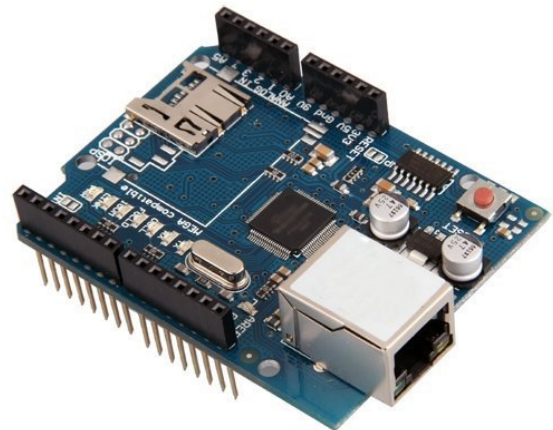


Figura 6: Módulo Ethernet Shield.

La comunicación con la tarjeta Arduino Mega 2560 se realiza mediante la comunicación SPI a través del conector ICSP. El chip Wiznet W5500 actúa como un dispositivo esclavo SPI controlado por el microcontrolador ATmega328 como el SPI maestro. El protocolo SPI necesita al menos cuatro señales: una señal MOSI (Master Out Serial In), una señal MISO (Master In Serial Out), una señal SCK (señal de reloj proporcionada por el maestro) y una señal CS (Chip Select) del esclavo.

El módulo Ethernet Shield 2.0 posee un slot para tarjetas SD, que se gestiona mediante comunicación SPI.

3 PROGRAMA DE CONTROL

Un programa de control es una secuencia de instrucciones que residen en un dispositivo de control y que determinan el comportamiento de un sistema, como por ejemplo, un robot cartesiano. De esta manera, se puede controlar los movimientos de un robot para que funcione como una impresora 3D.

Para ello, el programa de control se ha dividido en tres partes: un intérprete de programa (Arduino Mega 2560), un driver

controlador (Ethernet Shield 2.0); y un programa de usuario (S7-1200 de Siemens).

3.1 INTÉRPRETE DE PROGRAMA

En ciencias de la computación, un intérprete es un programa informático capaz de analizar y ejecutar otros programas. El intérprete ejecuta el programa instrucción a instrucción, sin previa compilación.

Como uno de los objetivos del proyecto, se ha planteado un intérprete de programa que sea capaz de leer archivos G-Code. Las instrucciones se leen desde una tarjeta SD, y traduce el G-Code en instrucciones de movimiento del robot cartesiano. En la Figura 7 se muestra el diagrama de flujo general del intérprete de programa.

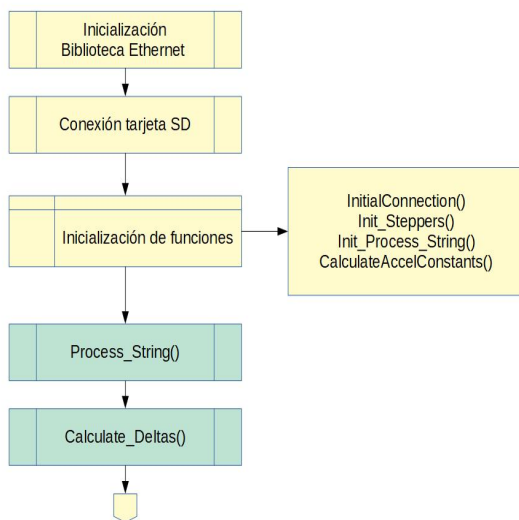


Figura 7: Diagrama de Flujo del Intérprete de Programa.

El intérprete de programa conecta la tarjeta Arduino Mega 2560 con el módulo Ethernet Shield 2.0, mediante de la librería “SPI.h”, conecta con el slot de tarjeta SD mediante de la librería “SD.h” y, finalmente, permite la conexión Ethernet mediante la librería “Ethernet2.h”. El uso de librerías en Arduino es una de las ventajas de estos dispositivos, pues se ahorra tiempo de

programación, y ayuda a centrar el código en el programa al control.

El arranque del programa de control comienza con la conexión con la tarjeta Arduino Mega 2560 y establece parámetros como la dirección MAC y la dirección IP local. Además, comprueba la comunicación con el módulo Ethernet Shield 2.0. La tarjeta SD almacena los modelos 3D en formato de archivo G-Code, tal como hemos comentado antes. Este proceso de arranque se ejecuta una única vez.

El programa de control entra en una función cíclica, que en primer lugar, abre el archivo que contiene el G-Code del modelo 3D. Seguidamente, realiza la lectura de la línea de instrucción, carácter a carácter, y se almacenan en un vector. Almacenados los valores de la línea de instrucción, el vector se envía a una función (Process_String()), que se encarga de su interpretación. Cada instrucción es interpretada por el programa de control, y es capaz de diferenciar entre las distintas instrucciones que el lenguaje G-Code puede contener [7]. Se han implementado funciones como el posicionamiento relativo y absoluto o el cambio de unidades.

Una vez interpretada la línea de instrucción, una función (Calculate_Deltas()) se encarga de los cálculos matemáticos de las variables de posición y velocidad. Las operaciones se realizan con valores de tipo INT o FLOAT, pero el envío de estos datos se realiza con valores de tipo CHAR. Por lo tanto, dicha función realiza la conversión de un tipo de variable a otro.

Obtenidos los datos, el siguiente paso es el envío de los mismos, mediante comunicación Ethernet. Para ello, en el programa de control se ha implementado un driver controlador que gestione el envío y recepción de datos a través del protocolo UDP.

3.2 DRIVER CONTROLADOR

Un driver o controlador de dispositivo es un programa informático que permite al sistema interactuar con un periférico, haciendo la abstracción del hardware y proporcionando una interfaz estandarizada para utilizar el dispositivo.

Por ello, para la comunicación entre la tarjeta Arduino Mega 2560 y el autómatas programable S7-1200 de Siemens, se ha programado un driver que controle el puerto Ethernet (RJ45), posibilitando el envío y recepción de las tramas de datos, y utilizando para ello el protocolo de comunicación UDP (User Datagram Protocol), y cuyo diagrama de flujo se muestra en la Figura 8.

Una vez almacenada la línea de instrucción en un vector, junto con el tamaño de la trama, el driver controlador conecta con el puerto Ethernet, y lo habilita para el envío o recepción de datos. Es importante destacar que para el establecimiento de la comunicación entre la tarjeta Arduino Mega 2560 y el autómatas programable S7-1200 de Siemens, se deben conocer la relación de parámetros entre ambos dispositivos, como son la dirección IP, la dirección del recurso o TSAP (dirección de nivel 4 en el modelo de referencia de la capa ISO 7) compuesta por el Rack (ID de recurso de conexión) y el Slot (ubicación específica).

Una vez que el programa de control comprueba los bits del TSAP, la conexión debe atravesar tres etapas o puertas: una primera conexión TCP; una segunda conexión ISO on TCP; y, finalmente, una negociación de la longitud de la trama del protocolo UDP.

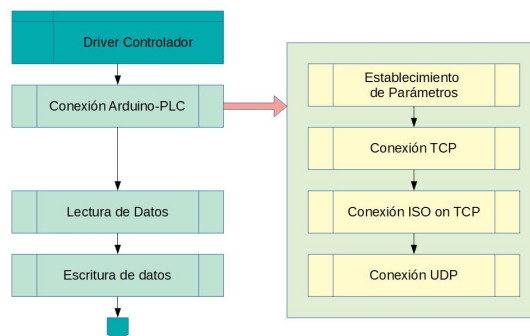


Figura 8: Diagrama de flujo del Driver Controlador.

Llegado a este punto, el programa de control realiza el envío o la recepción de datos y, para ello, se definen dos funciones diferentes: una función de lectura de datos y otra función de escritura de datos: `ReadArea()` y `WriteArea()`, respectivamente. La función de lectura de datos se encarga de la lectura de datos del autómatas programable S7-1200 de Siemens, y establece parámetros como el área de datos al cual se accede, la cantidad de datos que se lee, y la cabecera de la trama de datos. La función de escritura de datos se encarga de enviar los datos al autómatas programable S7-1200 de Siemens, y establece parámetros como la escritura de los bits de la cabecera de la trama de datos o la copia de los datos al búfer de salida.

3.3 PROGRAMA DE USUARIO

El autómatas programable S7-1200 de Siemens contiene el programa de control responsable del control del robot cartesiano. Este programa, también llamado programa de usuario, contiene todas las funciones necesarias para ejecutar la tarea de automatización específica, es decir, las funciones de comunicación y de movimiento del robot cartesiano.

El programa de usuario se ha estructurado en bloques, elaborado según el diagrama de flujo que se muestra en la Figura 9. Un bloque Reset permite acusar “Errores de funcionamiento con parada del eje” y “Errores de configuración”. Permite,

además, resetear aquellas marcas que necesiten ser inicializadas. Seguidamente, el bloque Home coloca la coordenada axial con la posición física del accionamiento. Este bloque realiza la referenciación de los ejes, para que el robot cartesiano pueda realizar movimientos de posicionamiento. Estos bloques se ejecutan una únicamente, al comienzo del programa.

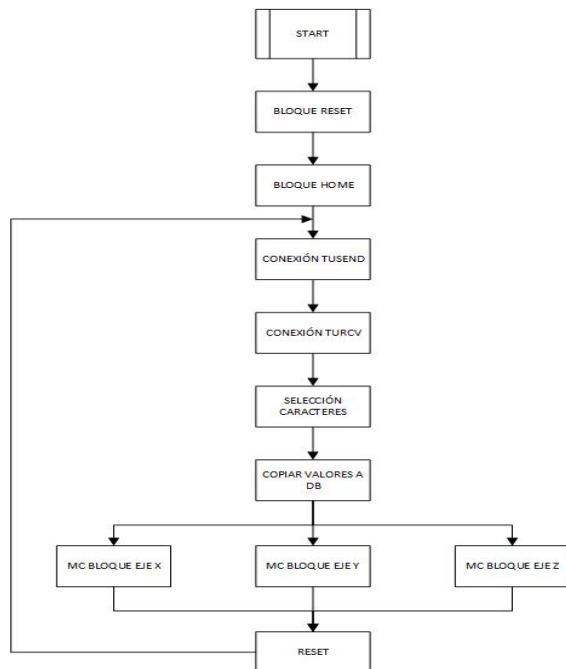


Figura 9. Diagrama de flujo del Programa de Usuario.

Un bloque TUSEND, que envía datos mediante el protocolo UDP, al interlocutor remoto, y establece la comunicación, que la mantiene y vigila automáticamente.

El bloque TURCV recibe datos a través del protocolo UDP, procedente del interlocutor remoto y establece la comunicación, que la mantiene y vigila automáticamente.

El programa de usuario continúa con un bloque llamado Selección de Caracteres, que almacena los datos recibidos por el interlocutor remoto (Arduino Mega 2560). Otro bloque copia los valores almacenados en un Bloque de Datos, el cual utilizan los bloques de Motion Control para el movimiento de los Ejes X,Y,Z. La función de Motion Control de los ejes X,Y,Z

permite realizar el control de los drivers de potencial que, a su vez, controlan los motores del robot cartesiano.

Finalmente, un bloque Reset se encarga de resetear todos los finales de carrera del robot cartesiano, así como la puesta a cero de las marcas utilizadas, estableciendo el programa para una nueva lectura de datos.

4 COMUNICACIÓN ENTRE DISPOSITIVOS

Todos los dispositivos que forma el sistema se han colocado dentro de una red de área local, y cuya comunicación para la transmisión de datos se ha realizado con el protocolo UDP (User Datagram Protocol), como se ha comentado en apartados anteriores.

4.1 ARQUITECTURA CLIENTE-SERVIDOR

La arquitectura Cliente-Servidor consiste básicamente en la realización de peticiones de un dispositivo (llamado Cliente) a otro dispositivo (llamado Servidor) que le da respuesta. En este caso, la tarjeta Arduino Mega 2560 junto con el módulo Ethernet Shield 2.0, actúa como cliente en una red de área local, y el autómata programable S7-1200 de Siemens, actúa como servidor de la misma red. En la Figura 10 se muestra un esquema representativo.

El servidor negocia con su sistema operativo un puerto donde esperar las solicitudes y, espera las peticiones en el puerto que se ha reservado para el servicio que ofrece. El cliente también solicita a su sistema operativo un puerto desde el cual enviar su solicitud y esperar la respuesta. Además, ubica un puerto arbitrario, no utilizado y no reservado, para la comunicación.

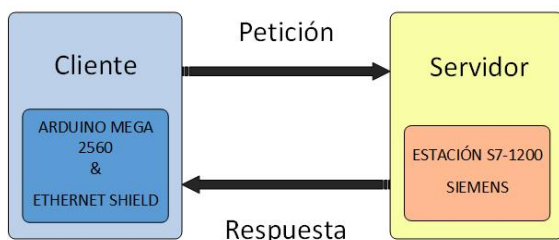


Figura 10. Esquema Arquitectura Cliente-Servidor.

La relación existente entre cliente y servidor es la establecida a través del intercambio de mensajes entre ambos, siendo éste, el mecanismo para la petición y entrega de solicitudes de servicios.

4.2 EL MODELO TCP/IP

El modelo TCP/IP (Protocolo de Control de Transmisión / Protocolo de Internet) es un conjunto de protocolos que permiten la comunicación entre distintos dispositivos pertenecientes a una red, que tiene especial importancia, dado que es el modelo usado para el acceso a una red Ethernet [8].

La tarjeta Arduino Mega 2560 permite conectarse a Ethernet mediante TCP/IP. La implementación de la pila de protocolos de TCP/IP en Arduino se realiza mediante un módulo Shield Ethernet 2.0, que proporciona la capa de “Acceso a red” (Ethernet o WiFi), “Internet” (IP) y “Transporte”. La capa de “Aplicación” se implementa directamente o mediante librerías, en dicha tarjeta.

Para el protocolo TCP/IP, la pila de protocolos del modelo OSI se simplifica para la tarjeta Arduino Mega 2560, como muestra la Figura 11.

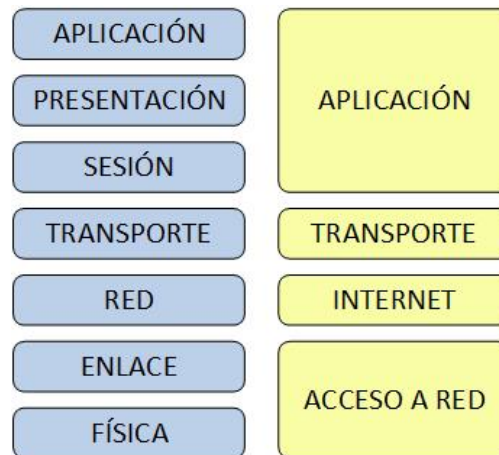


Figura 11. Modelo OSI (izquierda) y simplificación del modelo OSI para Arduino (derecha).

4.3 PROTOCOLO DE COMUNICACIÓN UDP

Como se ha comentado hasta ahora, el protocolo UDP (User Datagram Protocol) es un protocolo del nivel de Transporte (encapsulado de capa 4 o de Transporte del Modelo OSI) basado en el intercambio de datagramas. Permite el envío de datos a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información en el direccionamiento de su cabecera. Además, y como inconveniente, no posee confirmación ni de control de flujo (los paquetes pueden adelantarse unos a otros) ni de comprobación, ya que no hay confirmación de entrega o recepción. Es un protocolo mínimo de nivel de transporte orientado a mensajes, proporcionando una sencilla interfaz entre la capa de red y la capa de aplicación.

Dado que no otorga garantías para la entrega de sus mensajes, el control de la comunicación del envío y recepción de mensajes, debe ser programado por el usuario, para asegurar el funcionamiento del protocolo.

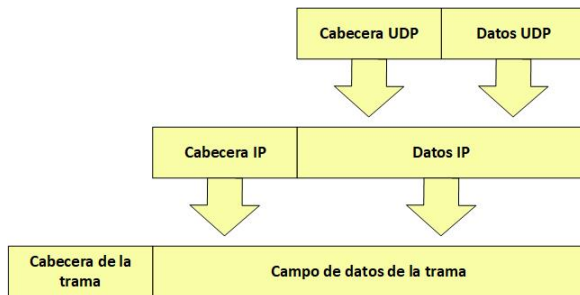


Figura 12: Encapsulamiento de datos.

Durante la transmisión, los datos cruzan cada una de las capas en el nivel del equipo remitente. En cada capa, se le agrega información al paquete de datos, tal como se puede ver en la Figura 12. A esto se llama “Encabezado”, es decir, una recopilación de información que garantiza la transmisión. En el nivel del equipo receptor, cuando se atraviesa cada capa, el encabezado se lee y después se elimina. Entonces, cuando se recibe el mensaje, éste se encuentra en su estado original. En cada nivel, el paquete de datos cambia su aspecto porque se le agrega un encabezado. Por tanto, las designaciones cambian según las capas: el paquete de datos se denomina “mensaje” en la capa de aplicación; el mensaje, después se encapsula en forma de “Segmento” en la capa de transporte; una vez se encapsula el segmento en la capa de internet, toma el nombre de “datagrama”; finalmente, se habla de “trama” en la capa de acceso a la red.

4.4 ESQUEMA DE CONEXIÓN

Llegados a este punto, y teniendo en cuenta las características de la comunicación entre dispositivos que se han visto en apartados anteriores, es de especial importancia el conexionado de los mismos. Para ello, se ha configurado una red de área local, conectando los dispositivos como se muestra en la Figura 13.

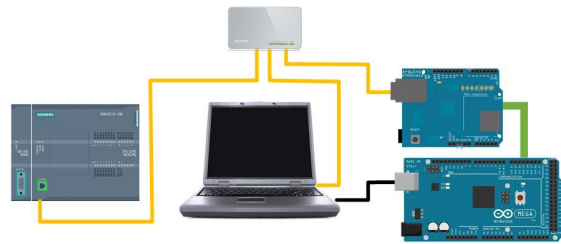


Figura 13. Esquema de conexión entre dispositivos.

Se ha utilizado un Switch que enlaza el módulo Ethernet Shield 2.0 con el autómata programable S7-1200 de Siemens. Además, se ha realizado la conexión a un PC, permitiendo la visualización en tiempo real del envío y recepción de los datos.

5 RESULTADOS

Con el objetivo de comprobar el correcto funcionamiento del sistema de control desarrollado, se han realizado distintas pruebas con el robot cartesiano.

En primer lugar, se ha comprobado que la lectura de la tarjeta SD se realiza correctamente, conforme al programa de control.

Se ha verificado, además, que la transmisión y recepción de los datos entre dispositivos funciona correctamente, asegurando la ejecución de las instrucciones.

Por otro lado, los movimientos del robot cartesiano se realizan conforme a las instrucciones recibidas por la tarjeta Arduino Mega 2560, y a las instrucciones de control del programa de usuario.

Los finales de carrera se resetean cada vez que detecta el final del eje, por lo que se evita que el robot cartesiano quede bloqueado.

En [9] se encuentran disponibles varios vídeos sobre el funcionamiento del sistema.

6 CONCLUSIONES

Se ha realizado el control de un robot cartesiano, para su funcionamiento como impresora 3D. Se ha diseñado el sistema de manera que el Arduino lea un archivo G-Code (modelo 3D) de la tarjeta SD, lo interprete, y calcule las instrucciones de movimiento del robot cartesiano para que el S7 1200 controle. Como tareas futuras se contempla el control de la extrusión del material fundido para el funcionamiento completo de una impresora 3D de gran formato.

English summary

CONTROL OF CARTESIAN ROBOT BY USING SIEMENS S7-1200 PLC AND ARDUINO FOR 3D PRINTING

Abstract

A control system of a cartesian robot is presented in this work, through the communication, among different controllers, for its functioning as a 3D printer. The control system is based on a Siemens S7-1200 programmable controller, an Arduino Mega 2560 card and an Ethernet Shield 2.0 module. The control program was developed through the Arduino IDE and TIA Portal v13 development environments. The communication between the Arduino card and the S7-1200 programmable controller has been made through Ethernet connection, through the UDP (User Datagram Protocol) communication protocol. The system has been designed so that the control program reads a G-Code file (3D model), interprets it and executes the movement instructions of the cartesian robot. The control program does not contemplate the possibility of extrusion, and focuses on controlling the movement of the axes and on communication between devices.

Keywords: cartesian robot, programmable logic controller, Arduino, 3D printer, UDP protocol.

Referencias

- [1] Barrientos, A., Peñin, L.F., Balaguer, C., Aracil, R., (1997) "Fundamentos de Robótica Segunda Edición", McGraw-Hill.
- [2] SIG POSITEC, (2002) "Documentación Técnica Twin Line Drive 01x", Edición d107.
- [3] Siemens, (2009) "Manual de producto SIMATIC S7-1200 Easy Book".
- [4] Siemens, (2014) "Manual de funciones SIMATIC STEP 7 S7-1200 Motion Control V13".
- [5] Arduino Mega 2560, <https://store.arduino.cc/arduino-mega-2560-rev3>.
- [6] Modulo Ethernet Shield 2.0 <https://store.arduino.cc/arduino-ethernet-shield-2>.
- [7] Intérprete G-Code de Arduino, <https://reprap.org/wiki/G-code/es>.
- [8] Modelo TCP/IP, <https://aprendiendoarduino.wordpress.com/tag/direccion-mac/>.
- [9] Portero, Alberto (2017), <https://www.youtube.com/watch?v=APrAguy8Sso>.



© 2018 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC 3.0 license (<https://creativecommons.org/licenses/by-nc/3.0>).