



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE TECNOLOGÍA EN ELECTROMECAÁNICA**

**MONOGRAFÍA, PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO  
EN ELECTROMECAÁNICA**

**AUTORES: Guacapiña Claudio, Alex Miguel**

**Sierra Herrera, Dennis Wladimir**

**DIRECTOR: ING. Murillo Mantilla, Luis Alejandro**

**LATACUNGA**

**2021**





**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

Implementación de un sistema de control de un huerto robotizado utilizando electrónica y software de código abierto para asegurar el control y monitorización eficiente del proceso de producción agrícola.



## *Planteamiento del problema*

El constante crecimiento poblacional mundial ha sido evidente con el paso de los años, esto a su vez trae como consecuencia el incremento de los métodos y la cantidad de producción de alimentos para abastecer el consumo de los mismos en las generaciones futuras. Esto quiere decir, que es necesario incrementar la eficiencia de la agricultura en ciertos sectores. Los procesos de siembra y cosecha no siempre son los más eficientes ya que se están consumiendo más recurso de los necesarios para cultivar algo.



# *Justificación*

- El desarrollo del presente proyecto de investigación es plausible al incremento constante de la población y del relativo desperdicio de recursos para la siembra y cultivo de cualquier tipo de planta es que se ha considerado tomar acciones que afecten positivamente a dichos problemas.
- Con la implementación que se va a realizar del prototipo del huerto robotizado, se tiene como meta el de optimizar los recursos necesarios para el crecimiento de los cultivos, a su vez, con la automatización del mismo, también se busca el ahorrar tiempo de las personas que deseen hacer sus propios cultivos, ya que el huerto al tener un sistema de control automatizado, no requiere que exista una intervención manual constante por parte de alguna persona, y ese tiempo ahorrado podrá ser utilizado en cualquier otra actividad.



# Objetivos

## General

- Implementar el sistema de control a un huerto robotizado, mediante la utilización de elementos electrónicos y software de código abierto, para asegurar el control y monitorización eficiente del proceso de producción agrícola.



# Objetivos

## Específicos

- Determinar los dispositivos necesarios del sistema de control mediante el análisis de las funciones a desempeñar por el sistema robotizado asegurando el correcto desempeño.
- Diseñar el algoritmo de control del sistema asegurando la correcta funcionalidad a través de la utilización de software de código abierto.
- Implementar un sistema de interfaz humano-máquina a través de dispositivos electrónicos de bajo costo y código abierto que permitan el control local del sistema.
- Validar el sistema robotizado mediante la ejecución de pruebas de campo para asegurar el correcto funcionamiento



# *Alcance*

- La presente investigación tiene como alcance realizar un sistema de control para el huerto robotizado, mediante la interacción del usuario por un HMI que será una pantalla matricial de 20 x 4 caracteres, un teclado de 4 x 4 caracteres y su respectivo Arduino Mega. A su vez, ésta irá conectada a un microcontrolador que será un Arduino MEGA con un Shield Ramps 1.4 y sus respectivos drivers de control.
- Respecto a las funciones del sistema de control, se espera que mediante el HMI se pueda controlar y monitorizar todo el proceso de siembra y cultivo de las verduras, exceptuando el proceso de cosecha. Dichas funcionalidades incluirán la perforación del suelo para el posterior depósito de la semilla y el riego del agua necesaria para el crecimiento de la planta.



# *Funcionalidades de la máquina*

El huerto robotizado se basa en una máquina que puede ser doméstica como industrial. La cual se utilizará para la siembra de una variedad de plantas y a su vez el usuario tenga su propio huerto doméstico con una autonomía a través de un sistema de control, creada para que funcione introduciendo parámetros de control y realice tareas asignadas con la programación.





# *Código abierto de Farmbot*

Farmbot como tal está comprometido a mantener transparente toda su documentación, a su vez, al ser un proyecto de código abierto y uso libre, se puede hallar la documentación y especificaciones de los detalles del huerto robotizado.

El firmware que se implementó está bajo la licencia “MIT License”; ésta a su vez garantiza el uso del software de manera libre o privada.



# *Lenguaje de programación a utilizar*

Debido a que el software que se va a utilizar para la programación de ambas placas controladoras es Arduino, por ende el lenguaje de programación respectivo será C++. Cabe recalcar que Arduino no maneja C++ de manera pura. Más por el contrario, el lenguaje de programación que maneja es una adaptación que permite que el funcionamiento de las placas a programar sea el adecuado.



## *Firmware para la Ramps 1.4*

El firmware implementado en el Arduino receptor de los códigos para el movimiento de los motores y la activación de las salidas de potencia es el utilizado por Farmbot, también conocido como Farmduino. Su firmware está bajo la licencia de uso libre MIT License, que es una licencia permisiva breve y a su vez simple con sus condiciones de uso.



# Códigos enviados al firmware de Farmduino.

Esos son los códigos que están implementados para que el firmware reciba del otro arduino, existen más comandos programados dentro del firmware para su recepción, pero que no ha sido considerado su implementación dentro del sistema de control.

| Tipo de código: | Número | Parámetros  | Funcionalidad                                                                                                               |
|-----------------|--------|-------------|-----------------------------------------------------------------------------------------------------------------------------|
| G               | 00     | X Y Z A B C | Mover a determinada ubicación, a la velocidad dada para el eje (no tiene que ser una línea recta), en coordenadas absolutas |
| G               | 28     |             | Mover al origen a todos los ejes (orden de los ejes Z, Y, X)                                                                |
| F               | 09     |             | Restablecer el paro de emergencia                                                                                           |
| F               | 22     | P V         | Escribe un valor "V" de un parámetro "P".                                                                                   |
| F               | 41     | P V M       | Establezca un valor "V" en un pin arduino en modo "M" (digital = 0 / analógico = 1)                                         |
| E               |        |             | Paro de Emergencia.                                                                                                         |



## *Parámetros implementados en los códigos a enviar.*

Dentro de los códigos enviados al Arduino de la Ramps 1.4, existen unos parámetros que irán variando y ellos determinarán la manera en que los motores son movidos, que tipo de entrada será activada, la velocidad de los motores etc.

| Parámetros | Descripción                            | Unidad de medida            |
|------------|----------------------------------------|-----------------------------|
| X          | Movimiento en X                        | Milímetros                  |
| Y          | Movimiento en X                        | Milímetros                  |
| Z          | Movimiento en X                        | Milímetros                  |
| A          | Velocidad en X                         | Pasos / segundo             |
| B          | Velocidad en X                         | Pasos / segundo             |
| C          | Velocidad en X                         | Pasos / segundo             |
| Q          | Queue number(número en cola o de cola) | #                           |
| P          | Parametro/ número de pin               | #                           |
| V          | Valor de número                        | #                           |
| M          | Modo (Establece el modo del pin)       | 0 = digital / 1 = analógico |

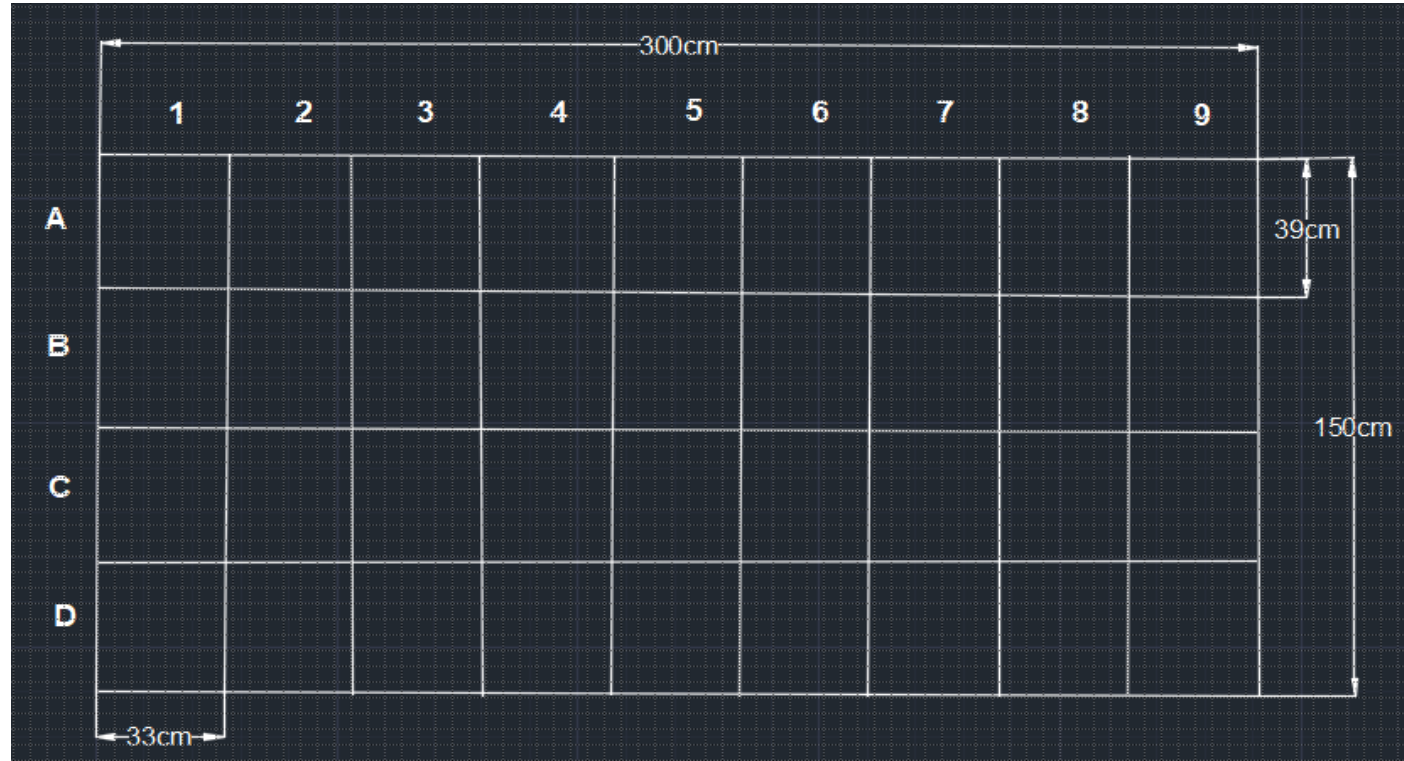


# *Sistema de sembrado*

- Este sistema consta de una distribución de la zona de sembrado, la dimensión es básicamente rectangular 1.5 m de ancho x 3 m de largo. La ubicación donde los ejes va a depositar la semilla es un cuadrado, comienza desde el cuadrante A1 hasta el cuadrante D8, Se puede observar donde van a ir plantadas las semillas y como va ir distribuido el espacio a la hora que el usuario quiera plantar en su huerto robotizado. La columna 9 irá vacía, esto por tema de limitaciones físicas con la placa del carro del Eje X; en la mencionada columna irá cualquier material que el usuario crea conveniente tener al alcance del Huerto Robotizado.



# Sistema de sembrado



# *Interfaz Humano - Máquina*

- Diseño del Menú

Para el manejo del Huerto Robotizado se creó el menú para determinar las funciones que va a permitir realizar la máquina. Al iniciar la pantalla de la LCD recibirá la función menú\_intro (); al usuario con el mensaje de introducción escrito en la pantalla con el texto “BIENVENIDO”;“INICIANDO”;” HUERTO ROBOTIZADO”.





# *Diseño del Menú*

- “A: SEMBRAR “
- “B: REGAR “
- “C: REFERENCIACION “
- “D: DESACTIVACIÓN Y ACTIVACIÓN“



# *Código para el Arduino del HMI*

El Arduino que cumple la función de HMI y a su vez de enviar la información respectiva al Arduino de la Ramps 1.4 es el principal en el proceso de control. Sus funcionalidades son las siguientes:

- Recibir las teclas pulsadas por el usuario en el teclado matricial de 4 x4.
- Reflejar la información respectiva en la pantalla LCD de 20 x 4 caracteres.
- Recibir las señales de los finales de carrera para la adecuada referenciación de los ejes del huerto robotizado.
- Enviar por el puerto serial los datos respectivos al otro Arduino para el respectivo movimiento de los motores y a control de las salidas de potencia.



# Funciones principales

| Tipo | Nombre          | Definición                                                                                                                                |
|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| void | Setup           | Es la función que se ejecuta una sola vez al encender o reiniciar el arduino.                                                             |
| char | Loop            | Contiene el código que se ejecutará de manera repetitiva, de ahí su nombre de "loop" que significa bucle.                                 |
| char | opcion_a        | Es todo el código que se ejecutará al seleccionar la opción A en el teclado matricial, es decir, abarca todo el proceso para el sembrado. |
| void | opción_letra_A  | Obtiene la letra respectiva de la coordenada de sembrado.                                                                                 |
| char | opción_numero_A | Obtiene el dígito respectivo de la coordenada de sembrado.                                                                                |
| char | opcion_b        | Es todo el código que se ejecutará al seleccionar la opción B en el teclado matricial. Esto abarca todo el proceso de regado.             |
| void | opción_letra_B  | Obtiene la letra respectiva de la coordenada de regado.                                                                                   |
| void | opción_numero_B | Obtiene el dígito respectivo de la coordenada de regado.                                                                                  |
| void | opcion_c        | Es todo el código que se ejecutará al seleccionar la opción C en el teclado matricial.                                                    |
| char | opcion_d        | Es todo el código que se ejecutará al seleccionar la opción D en el teclado matricial.                                                    |
| void | menu_intro      | Es todo el texto que sale al encender el arduino, esta función se ubica en el <u>setup</u> y por consiguiente se ejecuta una sola vez.    |
| void | opcion_menu_0   | Es la función que recibe las 4 letras posibles para las 4 opciones establecidas.                                                          |
| void | activar_D8      | Activa la salida de potencia D8                                                                                                           |



# Funciones principales

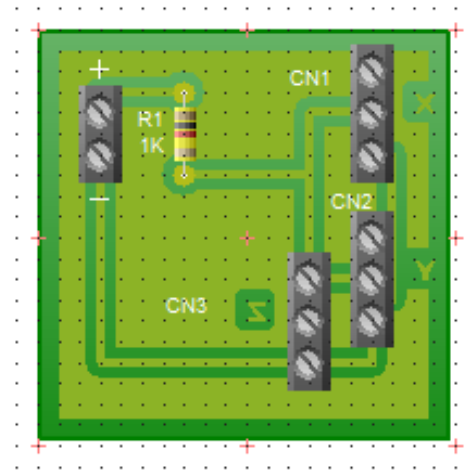
| Tipo | Nombre                | Definición                                                                                                                                                                     |
|------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| void | desactivar_D8         | Desactiva la salida de potencia D8                                                                                                                                             |
| void | activar_D9            | Activa la salida de potencia D9                                                                                                                                                |
| void | desactivar_D9         | Desactiva la salida de potencia D9                                                                                                                                             |
| void | activar_D10           | Activa la salida de potencia D10                                                                                                                                               |
| void | desactivar_D10        | Desactiva la salida de potencia D10                                                                                                                                            |
| void | activacion_manual     | Esta función envía el comando respectivo para activar el firmware de Farmduino y por consiguiente hacer funcionar los motores y las salidas de potencia de la Shield Ramps 1.4 |
| void | paro_emergencia       | Es la función que envía el comando que ejecuta el paro de emergencia.                                                                                                          |
| void | reset_paro_emergencia | Es la función que envía el comando que reinicia el paro de emergencia; y por consiguiente activa nuevamente el uso de la Shield Ramps 1.4.                                     |
| bool | movimiento_motor      | Es aquella función que realiza el movimiento de los motores tanto en los ejes X, Y y Z del huerto robotizado.                                                                  |
| void | check_endstop         | Es la función que verifica que los finales de carrera hayan sido pulsados o no.                                                                                                |
| void | enable_endstops       | Es aquella función que se ejecuta en el setup del Arduino y habilita los pines respectivos de los finales de carrera.                                                          |
| void | home_all_axis         | Es una función que ubica los 3 ejes en el Cero máquina, esto mediante los finales de carrera y la función check_endstop.                                                       |
| void | regar_agua            | Esta función se encarga de activar y desactivar la electroválvula, permitiendo así el paso del agua por un tiempo determinado.                                                 |
| void | recoger_semilla       | Es la función que realiza el movimiento de los motores para que el brazo robótico pueda recoger la semilla.                                                                    |



# Final de Carrera

- Referenciación

Los finales de carrera muestran que la máquina empieza en el mismo punto, esto es conocido como home o lugar de inicio que realiza el trabajo. Estos finales de carrera se van a distribuir para darle conocimiento a los ejes X, Y, Z cuales van hacer los puntos de referencia que van a tener que trabajar, los finales de carrera van a estar colocados en el inicio de los ejes permitiendo que puedan topar con la estructura y así enviar la señal al Arduino de control.



# Drivers para motores

- Características del motor de los ejes X1, X2. Y

| Motor Nema KS42STH40-1204A |                 |                              |           |                    |       |
|----------------------------|-----------------|------------------------------|-----------|--------------------|-------|
| Paso Ángulo                | Voltaje Nominal | Voltaje de Suministro Lógico | Corriente | Sosteniendo Torque | Peso  |
| 1.8/200 pasos              | 12V - 24V DC    | 3.3V – 5V                    | 1.2 A     | 4 N.cm             | 290 g |

- Características del motor del eje Z.

| Motor Nema 17HS4401 |                 |                              |           |                    |       |
|---------------------|-----------------|------------------------------|-----------|--------------------|-------|
| Paso Ángulo         | Voltaje Nominal | Voltaje de Suministro Lógico | Corriente | Sosteniendo Torque | Peso  |
| 1.8/200 pasos       | 12V - 24V DC    | 3.3V – 5V                    | 1.7 A     | 40 N.cm/4.5 Kgf.cm | 290 g |



# Drivers para motores

- Características del Driver.

| A4988 Driver         |            |
|----------------------|------------|
| Voltaje de Operación | Intensidad |
| 8 - 35 V             | 2 A        |

- Para determinar el voltaje de referencia del eje Z que se va a utilizar, se tendría que tomar en cuenta la corriente que va a ingresar en el motor paso a paso, conociendo la resistencia de sensado que nos brinda el driver A4988, en su placa nos indica una resistencia de  $0.1 \Omega$ . Aplicamos la fórmula que veremos a continuación:

$$V_{ref} = I_{max} * (8 * R_s)$$



# Cálculos de Calibración

Eje Z

$$V_{ref} = I_{max} * (8 * R_s)$$

$$V_{ref} = 1.7 \text{ A} * (8 * 0.1 \text{ } \Omega)$$

$$V_{ref} = 1.36 \text{ V}$$

Donde:

$V_{ref}$  = Voltaje de referencia.

$I_{max}$  = Corriente Máxima

$R_s$  = Resistencia de sensado

El manejo de cada motor va a ser a pasos completos, es decir, sin usar Jumpers en la Ramps, el voltaje de referencia será el 70% del obtenido anteriormente. Por lo tanto:

$$V_{ref} = 1.36 \text{ V} * 0.7$$

$$V_{ref} = 0.952 \text{ V}$$





# Cálculos de Calibración

Eje X1, X2, Y.

$$V_{ref} = I_{max} * (8 * R_s)$$

$$V_{ref} = 1.2 \text{ A} * (8 * 0.1 \text{ } \Omega)$$

$$V_{ref} = 0.96 \text{ V}$$

$$V_{ref} = 1.92 \text{ V} * 0.7$$

$$V_{ref} = 0.672 \text{ V}$$



# *Cálculo de los parámetros a utilizar.*

Para la distancia desplazada se observó que el parámetro numérico que va acompañado de las letras X, Y y Z son la distancia en milímetros. Para corroborar que la distancia desplazada en milímetros sea la misma mandada por el Arduino de control se hizo lo siguiente:

- Se envió un comando desplazando 1000 unidades en el Eje X, 500 en el eje Y y 100 en el eje Z.
- Se tomó como referencia el home como la distancia de origen.
- Se midió el desplazamiento de los ejes con referencia a la distancia del home.

Con esto se concluyó que efectivamente el parámetro que se ingresa en los comandos para mover los motores, es el desplazamiento en milímetros de los ejes.



# *Cálculo de los parámetros a utilizar.*

Para el valor del “Delay” o retraso que tendrán las funciones se hizo un análisis similar al mencionado anteriormente; el proceso consiste en:

- Enviar un comando desplazando 1000 unidades en el Eje X, 500 en el eje Y y 100 en el eje Z.
- Iniciar el cronómetro tan pronto los motores empezaron a girar.
- Detener el cronómetro a lo que los motores dejen de girar.

Con esto se concluyó que el valor del “delay” por cantidad de desplazamiento es el mismo para el eje X como el eje Y. Más por el contrario el valor del Eje Z tomo valores más elevados, considerando que el movimiento de dicho eje debe ser lento.



# Conclusiones

- Para este tipo de dispositivos, se investigó la movilidad que tiene una impresora 3D, ya que los componentes que tienen estas máquinas nos permiten mover los ejes X, Y, Z, los cálculos que se pudieron realizar fueron a los drivers que con una formula obtuvimos los voltajes de referencia para cada motor y poder mover los ejes.
- Las funcionalidades básicas del huerto robotizado, tales como: el proceso de siembra, el proceso de riego, la referenciación de la máquina y la adecuada interacción con el usuario fueron adecuadamente implementadas. El firmware utilizado para el movimiento de los motores es de Farmbot bajo la *MIT License* garantizando así que éste sea un software de código abierto.
- Previo a la implementación humano – maquina, se diseñó un menú con unos parámetros que fueron controlados por una pantalla LCD y un teclado matricial con la finalidad de trabajar con funciones creadas para que cada parámetro realice un trabajo distinto y así el usuario tenga la disponibilidad de controlar su huerto robotizado.
- Ejecutando las funciones respectivas en el Arduino implementado para el control del huerto robotizado, se observó que tanto el proceso de siembra, como de regado y el proceso de referenciación de la máquina fueron exitosamente ejecutados. Garantizando por lo tanto, el adecuado funcionamiento del huerto robotizado.



# *Recomendaciones*

- Corroborar el valor las resistencias que vienen integradas en los drivers A4988, ya que cada componente viene con diferentes parámetros, así depende del voltaje de funcionamiento respectivo de cada driver.
- Optimizar el código implementado en el Arduino de control para garantizar una menor cantidad de líneas de código implementados en el proceso de programación, preservando el adecuado funcionamiento del mismo.
- Realizar diversas validaciones del huerto robotizado en condiciones especiales. Para garantizar así que el funcionamiento del proyecto técnico sea el adecuado inclusive en situaciones que no se podrían llegar a considerar principalmente.



GRACIAS POR SU  
ATENCIÓN

