



Universidad de Los Andes

Manual de Usuario

Juan Pablo Reyes Fajardo

Índice

1. Introducción a Marlin y Configuración del Hardware	2
1.1. ¿Qué es Marlin?	2
1.2. Hardware Requerido	2
1.3. Flasheo de Marlin en Arduino Mega	2
2. Comunicación Serial y G-Code	2
2.1. Comunicación Serial	2
2.2. G-Code: Lenguaje de Control	2
2.3. Ejemplo de Secuencia de Comandos	3
3. Automatización de Trayectorias con Python	3
3.1. Instalación de pyserial	3
3.2. Ejemplo de Código Python para Controlar el Sistema de Movimiento	3
4. Conclusión	4

1. Introducción a Marlin y Configuración del Hardware

1.1. ¿Qué es Marlin?

Marlin es un firmware de código abierto ampliamente utilizado en impresoras 3D y otros sistemas de movimiento controlados por microcontroladores. Es altamente configurable y diseñado para trabajar con una variedad de hardware, lo que lo hace ideal para proyectos de automatización y robótica. Marlin convierte los comandos de G-code en señales eléctricas que controlan los motores paso a paso y otros actuadores del sistema.

1.2. Hardware Requerido

Para este sistema de movimiento, necesitarás:

- **Controlador:** Arduino Mega 2560, que actúa como el cerebro del sistema.
- **Drivers de Motores:** TB6600, que amplifican la señal del controlador para mover los motores.
- **Motores Paso a Paso:** Nema23, que son los actuadores físicos que generan el movimiento.

1.3. Flasheo de Marlin en Arduino Mega

Para cargar Marlin en tu Arduino Mega, sigue estos pasos:

1. **Descarga del Firmware:** Ve al sitio web de Marlin y descarga la última versión del firmware.
2. **Configuración:** Abre el archivo `Configuration.h` con un editor de texto y ajusta los parámetros según las especificaciones de tu hardware.
3. **Compilación y Carga:** Abre el Arduino IDE, carga el código de Marlin y selecciona tu placa Arduino Mega. Luego, compila y carga el firmware en el controlador.

2. Comunicación Serial y G-Code

2.1. Comunicación Serial

La comunicación entre tu computadora y el sistema de movimiento se realiza a través de una conexión serial, generalmente a través de un cable USB. Es importante configurar la velocidad de baudios correctamente, comúnmente en 115200 o 250000, para garantizar una comunicación fluida.

2.2. G-Code: Lenguaje de Control

El G-code es el lenguaje utilizado para comunicar a Marlin las acciones que debe realizar. Cada comando de G-code se compone de una letra seguida de uno o más números, y puede incluir parámetros adicionales. Aquí hay una descripción más detallada de algunos comandos importantes:

- **G0 y G1 - Movimiento Lineal:** Ambos comandos se utilizan para mover el sistema de un punto a otro. G0 se utiliza para movimientos rápidos sin importar la trayectoria, mientras que G1 se utiliza para movimientos controlados donde se especifica la velocidad. La sintaxis es G0 X10 Y10 Z10 o G1 X10 Y10 Z10 F1500, donde X, Y y Z son las coordenadas de destino y F es la velocidad en mm/min.
- **G28 - Homing:** Este comando se utiliza para llevar los ejes a su posición de origen. La sintaxis es simplemente G28 o G28 X Y para hacer homing solo en los ejes X e Y.
- **G90 y G91 - Posicionamiento Absoluto y Relativo:** G90 establece el modo de posicionamiento absoluto, donde las coordenadas se interpretan como posiciones fijas. G91 establece

el modo de posicionamiento relativo, donde las coordenadas se interpretan como desplazamientos desde la posición actual. La sintaxis es simplemente G90 o G91.

- **G92 - Establecer Posición Actual:** Este comando se utiliza para definir la posición actual del sistema como una coordenada específica. La sintaxis es G92 X0 Y0 Z0 para establecer la posición actual como el origen.
- **M203 - Establecer Velocidad Máxima:** Permite definir la velocidad máxima a la que pueden moverse los ejes. La sintaxis es M203 X500 Y500 Z500, donde los números representan la velocidad máxima en mm/min para cada eje.
- **M204 - Establecer Aceleración:** Este comando establece la aceleración de los ejes, es decir, qué tan rápido pueden aumentar su velocidad. La sintaxis es M204 P500 R1000 T1500, donde P es la aceleración de impresión, R es la aceleración de retracción y T es la aceleración de desplazamiento.
- **M84 - Desactivar Motores:** Apaga los motores paso a paso para permitir el movimiento manual. La sintaxis es simplemente M84.

Sintaxis de Comandos: Cada comando de G-code debe finalizar con un salto de línea (`\n`) para indicar el final del comando. No es necesario agregar un tabulador (`\t`) al final de los comandos.

2.3. Ejemplo de Secuencia de Comandos

Aquí hay un ejemplo de cómo se vería una secuencia de comandos para activar los motores, hacer homing, mover el sistema a un punto específico y luego desactivar los motores:

```
G28                ; Homing en todos los ejes
G1 X100 Y100 F1500 ; Mover a la posición X100, Y100 a una velocidad de 1500 mm/min
M84                ; Desactivar motores
```

3. Automatización de Trayectorias con Python

Es posible automatizar la generación de trayectorias y el envío de comandos de G-code utilizando Python. Esto se puede lograr mediante la biblioteca `pyserial` para la comunicación serial y la creación de funciones que generen los comandos de G-code correspondientes a las trayectorias deseadas.

3.1. Instalación de pyserial

Para instalar `pyserial`, ejecuta el siguiente comando en tu terminal o prompt de comandos:

```
pip install pyserial
```

3.2. Ejemplo de Código Python para Controlar el Sistema de Movimiento

A continuación, se muestra un ejemplo de código Python que establece la comunicación serial con el sistema de movimiento, envía una secuencia de comandos de G-code para realizar una trayectoria simple y luego cierra la conexión:

```
import serial
import time

# Establecer conexión serial
ser = serial.Serial('/dev/ttyACM0', 115200)
time.sleep(2) # Esperar a que la conexión se establezca
```

```

# Función para enviar comandos de G-code
def send_gcode(command):
    ser.write((command + '\n').encode())
    time.sleep(1) # Esperar a que el comando se ejecute

# Secuencia de comandos de G-code
send_gcode('G28') # Homing
send_gcode('G1 X50 Y50 F1500') # Mover a la posición X50, Y50
send_gcode('M84') # Desactivar motores

# Cerrar conexión serial
ser.close()

```

Este código utiliza la función `send_gcode` para enviar los comandos de G-code al sistema de movimiento. La secuencia de comandos realiza un homing, mueve el sistema a la posición (X50, Y50) y luego desactiva los motores.

4. Conclusión

Con esta guía detallada, esperamos que hayas obtenido una comprensión más profunda de cómo configurar y controlar un sistema de movimiento utilizando Marlin y G-code, así como cómo automatizar trayectorias utilizando Python. Con práctica y experimentación, podrás adaptar y expandir estas técnicas para satisfacer las necesidades de tus propios proyectos de automatización y robótica.