



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Eléctrica

**Diseño e implementación del control de
motores paso a paso mediante sistemas
embebidos**

Autor:

Soladana Monasterio, Roberto

Tutor:

**San Martín Ojeda, Moisés
Departamento de Ingeniería
Eléctrica**



Universidad de Valladolid

DISEÑO E IMPLEMENTACIÓN DEL CONTROL DE
MOTORES PASO A PASO MEDIANTE SISTEMAS EMBEBIDOS



ESCUELA DE INGENIERÍAS
INDUSTRIALES



Agradecimientos

Quiero agradecer a mi familia y a mis amigos confiar en mí, sin su apoyo no hubiera sido posible superar esta etapa de mi vida.

También quiero agradecer a los profesores que he tenido a lo largo de la carrera por sus enseñanzas y, en especial, a Moisés, que me ha guiado en todo momento para la realización de este trabajo.

A todos ellos, GRACIAS.



RESUMEN

El objeto de este trabajo es realizar el estudio y control de un motor paso a paso con la tarjeta Raspberry Pi, utilizando el programa LabVIEW para ello.

Se repasará el funcionamiento de las máquinas eléctricas y, más en profundidad, de los motores paso a paso, su respuesta ante pulsos, la oscilación del rotor y las curvas y parámetros característicos de los mismos.

A continuación, se explicará qué es una Raspberry Pi, en qué consiste el programa LabVIEW y cuáles son los dispositivos a utilizar en el desarrollo práctico del trabajo.

Se diseñará el sistema de control en LabVIEW, incluyéndose el diagrama de bloques y el panel de control donde se visualizarán las entradas y salidas correspondientes para comprobar el buen funcionamiento del programa.

Finalmente, se implementará el programa en la Raspberry Pi con salidas reales como es el propio motor y se comprobará experimentalmente el funcionamiento de los motores paso a paso, además del control remoto del mismo.

PALABRAS CLAVE

Motor paso a paso.

LabVIEW.

Raspberry Pi.

Control.

Dispositivos embebidos.



ABSTRACT

The purpose of this work is to study and control a stepper motor with the Raspberry Pi target and using the LabVIEW program for it..

The operation of electrical machines and, more in depth, of stepper motors, their response to pulses, the oscillation of the rotor and their characteristic curves and parameters will be reviewed.

Next, it will be explained what a Raspberry Pi is, what the LabVIEW program consists of and what are the devices to use in the practical development of the work.

The control system will be designed in LabVIEW, including the block diagram and the control panel where the corresponding inputs and outputs will be displayed to check the correct operation of the program.

Finally, the program will be implemented on the Raspberry Pi with real outputs such as the motor itself and the operation of the stepper motors will be experimentally verified, in addition to the remote control of it.

KEYWORDS

Stepper motors.

LabVIEW.

Raspberry Pi.

Control.

Embedded devices.



Universidad de Valladolid

DISEÑO E IMPLEMENTACIÓN DEL CONTROL DE MOTORES PASO A PASO MEDIANTE SISTEMAS EMBEBIDOS



ESCUELA DE INGENIERÍAS INDUSTRIALES



ÍNDICE PRINCIPAL

1. INTRODUCCIÓN Y OBJETIVOS	19
1.1. Objetivos.....	19
1.2. Justificación del trabajo	19
2. DESARROLLO TEÓRICO	21
2.1. Máquinas eléctricas	21
2.1.1. Máquinas de corriente continua.....	23
2.1.2. Motores paso a paso	30
3. DESARROLLO DEL TRABAJO	83
3.1. Introducción teórica.....	83
3.1.1. Raspberry Pi	83
3.1.2. Lenguaje de programación LabVIEW	85
3.2. Hardware empleado	94
3.2.1 Raspberry Pi	94
3.2.2. Motor paso a paso y driver.....	96
3.2.3. Otros componentes	99
3.3. Software desarrollado	100
4. CONCLUSIONES	141
5. PROPUESTAS DE TRABAJOS.....	145
6. BIBLIOGRAFÍA.....	157





ÍNDICE DE FIGURAS

Figura 1. Funcionamiento básico de un motor. Fuente: [1].....	21
Figura 2. Funcionamiento básico de un generador. Fuente: [1]	21
Figura 3. Reversibilidad de las máquinas eléctricas. Fuente: [2]	22
Figura 4. Aspectos constructivos de una máquina eléctrica. Fuente: [3].....	22
Figura 5. Fuerzas en una espira debido a un campo externo y una circulación de corriente por ella. Fuente: [4]	23
Figura 6. Modelo de motor con excitación independiente. Fuente: [5]	24
Figura 7. Modelo de motor con excitación serie. Fuente: [5]	25
Figura 8. Modelo de motor con excitación shunt. Fuente: [5].....	25
Figura 9. Modelo de motor con excitación compuesta corta y larga, respectivamente. Fuente: [5]	26
Figura 10. Modelo de motor de corriente continua básico. Fuente: [5]	26
Figura 11. Elementos de una máquina de corriente continua. Fuente: [6].....	27
Figura 12. Rectificación de corriente alterna a corriente continua. Fuente: [7]	28
Figura 13. Esquema motor brushless. Fuente: [8].....	28
Figura 14. Tipos de motores brushless: rotor en la carcasa y rotor en el eje, respectivamente. Fuente: [9]	29
Figura 15. Motor paso a paso de imanes permanentes. Fuente: [8].....	31
Figura 16. Motor paso a paso de reluctancia variable. Fuente: [8]	32
Figura 17. Motor paso a paso híbrido. Fuente: [37].....	32
Figura 18. Esquema de control en lazo abierto. Fuente: [10]	33
Figura 19. Esquema de control en lazo cerrado. Fuente: [10].....	34
Figura 20. Efectos en el par debido a la pérdida de pulso. Fuente: [11]	35
Figura 21. Motor paso a paso con encoder incorporado. Fuente: [13]	35
Figura 22. Operación en lazo cerrado con encoder. Fuente: [10].....	36
Figura 23. Conmutación avanzada vs convencional. Fuente: [10]	36
Figura 24. Paso completo con una bobina activa de un motor unipolar. Fuente: [14].....	37
Figura 25. Paso completo con dos bobinas activas de un motor unipolar. Fuente: [14].....	37
Figura 26. Diferencia en la respuesta ante un paso entre una bobina activa y dos bobinas activas. Fuente: [12].....	37
Figura 27. Medio paso en motor unipolar. Fuente: [14]	38
Figura 28. Micropaso en motor unipolar. Fuente: [14]	38
Figura 29. Principio de funcionamiento del micropaso. Fuente: [10].....	39
Figura 30. Motor paso a paso unipolar con 5 y 6 terminales accesibles. Fuente: [8]	40
Figura 31. Control del motor paso a paso unipolar. Fuente: [15]	40
Figura 32. Motor paso a paso bipolar. Fuente: [8]	46
Figura 33. Control del motor paso a paso bipolar. Fuente: [15]	46



Figura 34. Curva típica par velocidad de un motor paso a paso. Fuente: [11]56

Figura 35. Curvas par velocidad del motor paso a paso. Fuente: [17].....57

Figura 36. Curva típica par velocidad, con el efecto de la inercia de una carga. Fuente: [18].....57

Figura 37. Respuesta de un motor paso a paso ante un pulso. Fuente: [11].58

Figura 38. Respuesta del motor paso a paso ante varios pulsos. Fuente: [24]59

Figura 39. Distribución del par estático en un motor de reluctancia variable con una fase energizada. Fuente: [11]60

Figura 40. Respuesta del par para una secuencia de tres pasos, de secuencia de pasos no uniforme. Fuente: [11]61

Figura 41. Efecto de la duración de los pulsos en la corriente por el devanado. Fuente: [10].....62

Figura 42. Circuito para disminuir la constante de tiempo eléctrica. Fuente: [10].....62

Figura 43. Error de posición estática. Fuente: [11].....64

Figura 44. Respuesta típica de un motor paso a paso ante un solo pulso. Fuente: [11].....64

Figura 45. Obtención de K_m . Fuente: [11]65

Figura 46. Respuestas ante frecuencias de paso proporcionales a la frecuencia natural. Fuente: [11]66

Figura 47. Caídas en la característica par velocidad. Fuente: [29]67

Figura 48. Aproximación de par lineal. Fuente: [10]69

Figura 49. Modelo para oscilaciones de un motor paso a paso de un solo paso. Fuente: [10].....69

Figura 50. Motor paso a paso con un amortiguador Houdaille. Fuente: [10] .70

Figura 51. Respuesta típica de un solo paso de un motor paso a paso con un amortiguador Houdaille. Fuente: [11]71

Figura 52. Método de apagado de pulso. Fuente: [11].....72

Figura 53. Método de inversión de pulso. Fuente: [11].....73

Figura 54. Método de retraso de pulso. Fuente: [11]73

Figura 55. Par estático sinusoidal. Fuente: [11].....74

Figura 56. Modelo de un motor paso a paso. Fuente: [11]75

Figura 57. Circuito equivalente aproximado por fase de un motor paso a paso. Fuente: [11].....75

Figura 58. Aplicaciones de motores paso a paso en impresoras 3D. Fuente: [19].....79

Figura 59. Aplicaciones de motores paso a paso en telescopios. Fuente: [20]80

Figura 60. Aplicaciones de motores paso a paso en controladora de discos. Fuente: [23].....80

Figura 61. Aplicaciones de motores paso a paso en mesa de posicionamiento lineal para fresadora CNC. Fuentes: [10] y [21]81



Figura 62. Aplicaciones de motores paso a paso en una fábrica. Fuente: [10]81

Figura 63. Aplicaciones de motores paso a paso en brazos robóticos. Fuente: [22].....82

Figura 64. Raspberry Pi. Fuente: [25]83

Figura 65. Raspbian, Raspberry Pi + Debian. Fuente: [26]83

Figura 66. Sistema embebido. Fuente: [27]84

Figura 67. Logo de LabVIEW y National Instruments. Fuente: [28]85

Figura 68. Panel Frontal. Fuente: Elaboración propia.....88

Figura 69. Diagrama de Bloques. Fuente: Elaboración propia.....89

Figura 70. Paleta de controles. Fuente: Elaboración propia.....89

Figura 71. Ejemplo de entradas y salidas en el panel frontal. Fuente: Elaboración propia.90

Figura 72. Ejemplo de visualización y control en el panel frontal. Fuente: Elaboración propia.90

Figura 73. Paleta de funciones. Fuente: Elaboración propia.....91

Figura 74. Ejemplo de bucles disponibles en el diagrama de bloques. Fuente: Elaboración propia.91

Figura 75. Ejemplo de operaciones booleanas. Fuente: Elaboración propia. .92

Figura 76. Barra de herramientas. Fuente: Elaboración propia.92

Figura 77. Panel frontal con el programa ejecutándose. Fuente: Elaboración propia.93

Figura 78. Raspberry Pi modelo 3B+. Fuente: [30].....94

Figura 79. Pines de la Raspberry Pi. Fuente: [31]95

Figura 80. Motor 28BYJ-48 5V. Fuente: [32]96

Figura 81. Driver ULN2003AN. Fuentes: [33] y [34]97

Figura 82. Transistor Darlington. Fuente: [38]97

Figura 83. Protoboard y componentes. Fuente: [35]99

Figura 84. Estructura interna de la protoboard. Fuente: [36]99

Figura 85. Aspecto del panel frontal. Fuente: Elaboración propia. 100

Figura 86. Sentido de giro y modo de control del panel frontal. Fuente: Elaboración propia. 101

Figura 87. Variables del ciclo y control de velocidad en panel frontal. Fuente: Elaboración propia. 101

Figura 88. Variables del ciclo y control de velocidad en panel frontal en ejecución. Fuente: Elaboración propia. 102

Figura 89. Velocímetro e indicador de velocidad en el panel frontal ejecutándose el programa. Fuente: Elaboración propia. 102

Figura 90. Indicadores y driver en el panel frontal mientras se ejecuta el programa. Fuente: Elaboración propia. 103

Figura 91. Datos del motor y grados en el panel frontal mientras se ejecuta el programa. Fuente: Elaboración propia. 103

Figura 92. Sentido de giro y modo de control. Fuente: Elaboración propia.. 104



Figura 93. Condición de grados a girar \geq grados girados. Fuente: Elaboración propia..... 105

Figura 94. Características del motor. Fuente: Elaboración propia..... 106

Figura 95. Obtención de la frecuencia. Fuente: Elaboración propia..... 107

Figura 96. Control de velocidad mediante dos pulsadores. Fuente: Elaboración propia..... 107

Figura 97. Obtención del tiempo de ejecución de cada ciclo en ms. Fuente: Elaboración propia. 108

Figura 98. Case structure para sentido horario y paso completo con una bobina activa. Fuente: Elaboración propia..... 108

Figura 99. Obtención de pasos/vuelta necesarios para paso completo. Fuente: Elaboración propia. 109

Figura 100. Obtención de la velocidad de giro. Fuente: Elaboración propia.109

Figura 101. Condición IF para la secuencia de activación. Fuente: Elaboración propia..... 109

Figura 102. Conexión en los extremos del bucle while. Fuente: Elaboración propia..... 110

Figura 103. Activación de los pines correspondientes a sentido horario y paso completo con una bobina activa. Fuente: Elaboración propia..... 110

Figura 104. Obtención de los grados girados por el motor en un instante de tiempo. Fuente: Elaboración propia..... 111

Figura 105. Case structure para sentido antihorario y paso completo una bobina activa. Fuente: Elaboración propia..... 111

Figura 106. Secuencia de activación para sentido antihorario y paso completo con una bobina activa. Fuente: Elaboración propia. 112

Figura 107. Case structure para sentido horario y paso completo con dos bobinas activas. Fuente: Elaboración propia..... 112

Figura 108. Secuencia para sentido horario y paso completo con dos bobinas activas. Fuente: Elaboración propia. 113

Figura 109. Secuencia para sentido antihorario y paso completo con dos bobinas activas. Fuente: Elaboración propia. 113

Figura 110. Esquema general de medio paso con sentido horario. Fuente: Elaboración propia. 114

Figura 111. Estructura del IF para el control con medios pasos. Fuente: Elaboración propia. 114

Figura 112. Secuencia para control con medios pasos sentido horario. Fuente: Elaboración propia. 115

Figura 113. Secuencia para control con medios pasos sentido antihorario. Fuente: Elaboración propia. 115

Figura 114. Comparación en la obtención de los pasos/vuelta. Fuente: Elaboración propia. 116

Figura 115. Caso 0 del case structure principal. Fuente: Elaboración propia. 117



Figura 116. Caso 64 del case structure principal. Fuente: Elaboración propia. 117

Figura 117. Función del botón de paro. Fuente: Elaboración propia..... 118

Figura 118. Botón STOP CICLO. Fuente: Elaboración propia..... 118

Figura 119. Creación de un nuevo proyecto en LabVIEW. Fuente: Elaboración propia..... 119

Figura 120. Creación de un nuevo dispositivo. Fuente: Elaboración propia. 119

Figura 121. Elección de la tarjeta Raspberry Pi. Fuente: Elaboración propia. 120

Figura 122. Colocación del VI dentro del apartado Raspberry Pi. Fuente: Elaboración propia. 120

Figura 123. Opción Tools para la configuración de la Raspberry Pi. Fuente: Elaboración propia. 121

Figura 124. Configuración de la Raspberry Pi. Fuente: Elaboración propia. 121

Figura 125. Configuración de la Raspberry Pi finalizada. Fuente: Elaboración propia. 122

Figura 126. Propiedades del apartado Raspberry en el proyecto. Fuente: Elaboración propia. 122

Figura 127. Propiedades generales del apartado Raspberry en el proyecto. Fuente: Elaboración propia. 123

Figura 128. Conexión con la tarjeta Raspberry. Fuente: Elaboración propia. 123

Figura 129. Introducción del programa en la tarjeta Raspberry. Fuente: Elaboración propia. 124

Figura 130. Despliegue satisfactorio del programa en la tarjeta Raspberry. Fuente: Elaboración propia. 124

Figura 131. Bloques disponibles para la interconexión entre LabVIEW y la Raspberry. Fuente: Elaboración propia. 125

Figura 132. Colocación de la entrada. Fuente: Elaboración propia..... 125

Figura 133. Colocación del bloque Digital Write. Fuente: Elaboración propia. 126

Figura 134. Conexiones disponibles en cada bloque. Elaboración propia... 126

Figura 135. Conexión a realizar con el bloque Digital Write. Fuente: Elaboración propia. 126

Figura 136. Conexión a realizar con el bloque Digital Read. Fuente: Elaboración propia. 127

Figura 137. Pines a utilizar de la Raspberry. Fuente: Elaboración propia.... 127

Figura 138. Conexión con la tarjeta Raspberry. Fuente: Elaboración propia. 128

Figura 139. Esquema realizado. Fuente: Elaboración propia..... 128

Figura 140. Montaje real. Fuente: Elaboración propia..... 129

Figura 141. Motor girando a pasos completos con una bobina activa y sentido horario. Fuente: Elaboración propia. 130



Figura 142. Motor girando a pasos completos con dos bobinas activas y sentido antihorario mediante dos pulsadores. Fuente: Elaboración propia. 130

Figura 143. Creación del Web Service. Fuente: Elaboración propia. 131

Figura 144. VI de Lectura. Fuente: Elaboración propia..... 131

Figura 145. VI de Escritura. Fuente: Elaboración propia. 132

Figura 146. POST en el método de Escritura. Fuente: Elaboración propia. . 132

Figura 147. Modificación propiedades del Web Service. Fuente: Elaboración propia. 133

Figura 148. Prueba de funcionamiento del Web Service. Fuente: Elaboración propia. 133

Figura 149. Creación Web Application Project. Fuente: Elaboración propia. 134

Figura 150. Panel frontal NXG. Fuente: Elaboración propia. 134

Figura 151. Diagrama de bloques NXG. Fuente: Elaboración propia..... 135

Figura 152. Indicadores del lugar donde se realiza el control. Fuente: Elaboración propia. 135

Figura 153. Ejecución del control remoto. Fuente: Elaboración propia..... 136

Figura 154. Control remoto activo. Fuente: Elaboración propia..... 136

Figura 155. Creación página web para el control. Fuente: Elaboración propia. 137

Figura 156. Archivos correspondientes creados correctamente. Fuente: Elaboración propia. 137

Figura 157. Página Web del control remoto. Fuente: Elaboración propia... 138

Figura 158. Creación de aplicación en tiempo real. Fuente: Elaboración propia. 138

Figura 159. Modificación de la categoría Information. Fuente: Elaboración propia. 139

Figura 160. Modificación de la categoría Information. Fuente: Elaboración propia. 139

Figura 161. Ejecutable creado en el proyecto. Fuente: Elaboración propia. 140

Figura 162. Esquema de control motor brushless. Fuente: [39] 145

Figura 163. Funcionamiento de un servomotor convencional. Fuente: [40] 146

Figura 164. Control de un servomotor en lazo cerrado. Fuente: [10]..... 147

Figura 165. Funcionamiento de un servomotor de rotación continua. Fuente: [41]..... 147

Figura 166. Control en lazo cerrado de un motor paso a paso. Fuente: [10] 148

Figura 167. Encoder rotatorio. Fuente: [42] 148

Figura 168. Driver para el control con micropasos. Fuente: [43] 149

Figura 169. Esquema de conexión control con micropasos. Fuente: [44]... 149

Figura 170. Circuito equivalente aproximado por fase de un motor paso a paso. Fuente: [11]..... 150

Figura 171. Motor paso a paso con un amortiguador Houdaille. Fuente: [10] 151



Figura 172. Ejemplo de marcha-paro de un motor asíncrono trifásico en CADe_SIMU. Fuente: Elaboración propia. 152

Figura 173. Ejemplo simple de un marcha-paro en LabVIEW con Raspberry Pi, panel frontal. Fuente: Elaboración propia. 152

Figura 174. Ejemplo de marcha-paro en LabVIEW con Raspberry Pi, diagrama de boques. Fuente: Elaboración propia..... 153

Figura 175. Led encendido con el ejemplo del marcha-paro. Fuente: Elaboración propia. 153

Figura 176. Esquema simplificado de un motor paso a paso unipolar y bipolar, respectivamente. Fuente: [8] 154

Figura 177. Puente H para devanado de un motor paso a paso bipolar. Fuente: [45] 154





ÍNDICE DE TABLAS

Tabla 1. Paso completo, una bobina activa, horario I. Fuente: Elaboración propia.....	41
Tabla 2. Paso completo, una bobina activa, horario II. Fuente: [16]	41
Tabla 3. Paso completo, una bobina activa, antihorario I. Fuente: Elaboración propia.....	41
Tabla 4. Paso completo, una bobina activa, antihorario II. Fuente: [16].....	42
Tabla 5. Paso completo, dos bobinas activas, horario I. Fuente: Elaboración propia.....	42
Tabla 6. Paso completo, dos bobinas activas, horario II. Fuente: [16].....	42
Tabla 7. Paso completo, dos bobinas activas, antihorario I. Fuente: Elaboración propia.....	43
Tabla 8. Paso completo, dos bobinas activas, antihorario II. Fuente: [16].....	43
Tabla 9. Medio paso, horario I. Fuente: Elaboración propia.....	43
Tabla 10. Medio paso, horario II. Fuente: [16]	44
Tabla 11. Medio paso, antihorario I. Fuente: Elaboración propia.....	44
Tabla 12. Medio paso, antihorario II. Fuente: [16].....	45
Tabla 13. Paso completo, una bobina activa, horario I. Fuente: Elaboración propia.....	47
Tabla 14. Paso completo, una bobina activa, horario II. Fuente: [16].....	48
Tabla 15. Paso completo, una bobina activa, antihorario I. Fuente: Elaboración propia.....	48
Tabla 16. Paso completo, una bobina activa, antihorario II. Fuente: [16].....	49
Tabla 17. Paso completo, dos bobinas activas, horario I. Fuente: Elaboración propia.....	50
Tabla 18. Paso completo, dos bobinas activas, horario II. Fuente: [16].....	50
Tabla 19. Paso completo, dos bobinas activas, antihorario I. Fuente: Elaboración propia.....	51
Tabla 20. Paso completo, dos bobinas activas, antihorario II. Fuente: [16]...	51
Tabla 21. Medio paso, una bobina activa, horario I. Fuente: Elaboración propia.....	52
Tabla 22. Medio paso, una bobina activa, horario II. Fuente: [16]	53
Tabla 23. Medio paso, una bobina activa, antihorario I. Fuente: Elaboración propia.....	54
Tabla 24. Medio paso, una bobina activa, antihorario II. Fuente: [16].....	55
Tabla 25. Número resultante según la secuencia escogida. Fuente: Elaboración propia.....	106





1. INTRODUCCIÓN Y OBJETIVOS

1.1. Objetivos

Describir brevemente el funcionamiento de las máquinas eléctricas, principalmente las de corriente continua.

Explicar en profundidad el funcionamiento de los motores paso a paso y todos los factores a tener en cuenta a la hora de utilizarlos.

Describir el funcionamiento de la Raspberry Pi, el microcontrolador utilizado en este trabajo.

Explicar brevemente el funcionamiento de LabVIEW, configurar el programa para comunicarse con el microcontrolador y realizar un programa para controlar el motor.

Explicar el funcionamiento de cada conjunto de bloques utilizado en el programa.

Profundizar sobre el control de máquinas eléctricas mediante sistemas embebidos.

Proponer otras alternativas de trabajos con conceptos similares.

1.2. Justificación del trabajo

El objetivo principal de este Trabajo Fin de Grado está concebido con fines didácticos, para mostrar que LabVIEW es un software muy útil e intuitivo a la hora de realizar un programa para controlar un sistema en tiempo real y mostrar el potencial y la variedad de posibilidades del uso de la Raspberry Pi para tal fin.

Otro fin por el que se realiza este trabajo es el uso de un motor paso a paso, ya que es un motor especial que suele ser menos estudiado que los motores convencionales y se explica con facilidad su funcionamiento gracias a las gráficas y los esquemas expuestos en este trabajo.

Además, tiene una parte teórica más específica en la que se describen todos los parámetros a tener en cuenta, como la oscilación del rotor, posibilidad de pérdidas de pulso, tiempo de estabilización y modelos matemáticos de los distintos motores paso a paso.



2. DESARROLLO TEÓRICO

La información principal de este apartado se ha extraído de [3] y [5] para la descripción general de las máquinas eléctricas de corriente continua y de [10], [11], [12], [16] y [17] para el estudio de los motores paso a paso.

2.1. Máquinas eléctricas

Las máquinas eléctricas son dispositivos que convierten energía mecánica en eléctrica y viceversa, debido a la interacción de campos eléctricos y magnéticos que se producen en su interior.

Se puede distinguir entre **motores** y **generadores**.

Los **motores eléctricos** consumen energía eléctrica para producir energía mecánica. Se le inyecta a la máquina una corriente eléctrica mediante una fuente externa y se origina una interacción con el campo de magnético de esta, la cual produce el movimiento. Se obtiene en forma de par en el eje y se utiliza para realizar un determinado trabajo mecánico.

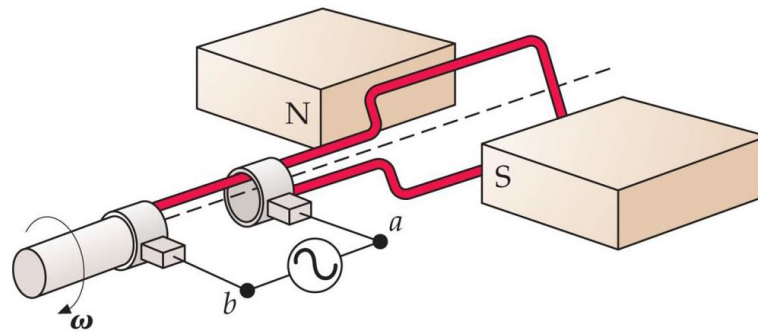


Figura 1. Funcionamiento básico de un motor. Fuente: [1]

Los **generadores eléctricos** aportan energía eléctrica por medio de energía mecánica. Se origina el movimiento de una bobina dentro de un campo magnético a partir de otras formas primarias de energía, resultando una f.e.m. inducida que al aplicarla a un circuito externo produce una corriente.

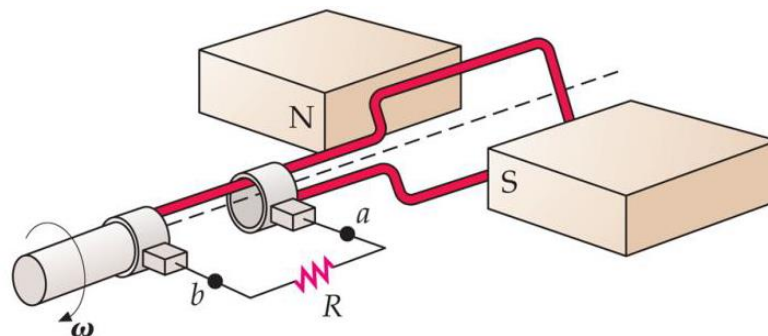


Figura 2. Funcionamiento básico de un generador. Fuente: [1]

No tiene por qué existir diferencia física entre un motor y un generador, pueden ser la misma máquina funcionando de diferente manera, es decir, las máquinas eléctricas son **reversibles**, como se puede observar de manera clara en la figura 3.

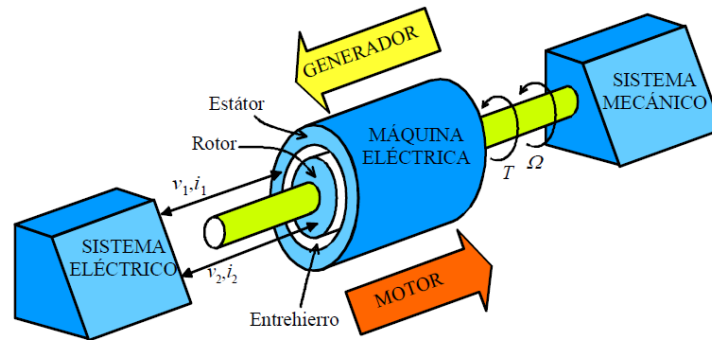


Figura 3. Reversibilidad de las máquinas eléctricas. Fuente: [2]

De manera general, una máquina eléctrica rotativa se compone de dos partes, el estator y el rotor.

El **estator** es la parte fija de la máquina y tiene forma de cilindro hueco. En él se encuentra el devanado del estator. En su interior se coloca el rotor.

El **rotor** es la parte móvil. Puede ser cilíndrico o de polos salientes. Es de un material ferromagnético en el cual se coloca el devanado rotórico. Tiene un hueco en el centro donde se sitúa el **eje**, el cual está unido al rotor y apoyado sobre unos rodamientos.

En la siguiente imagen se muestran las partes de una máquina eléctrica, aunque más adelante se verán las peculiaridades constructivas de cada tipo de máquina.

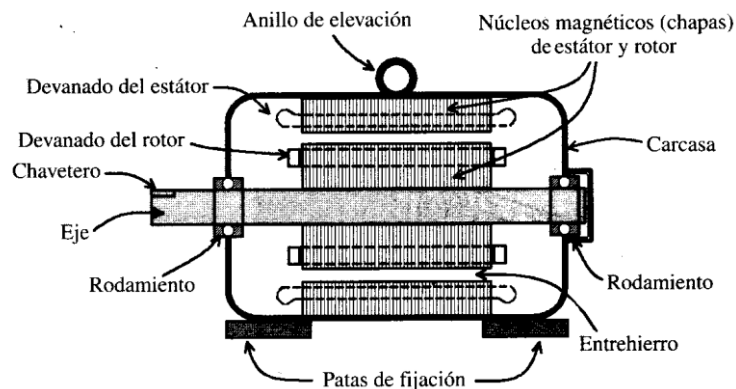


Figura 4. Aspectos constructivos de una máquina eléctrica. Fuente: [3]

El espacio de aire que separa el estator del rotor se denomina **entrehierro**. Es necesario para que pueda existir el movimiento e impide que ambas partes rocen entre sí. En el entrehierro es donde se producen las interacciones electromagnéticas que permiten la conversión de energía eléctrica en mecánica y viceversa.

También se pueden clasificar las máquinas eléctricas en máquinas de corriente continua y de corriente alterna, aunque se explicará el funcionamiento solamente las de corriente continua, que son las que atañen este trabajo.

2.1.1. Máquinas de corriente continua

Una máquina de corriente continua es una máquina eléctrica que funciona con magnitudes de tensión y corriente continuas.

El funcionamiento básico de un **motor** de corriente continua se explica a continuación:

Se alimenta al inductor (estator) con corriente continua, este produce un flujo que recorre todo el circuito magnético de la máquina de acuerdo a la Ley de Ampère: $\oint \vec{H} \cdot d\vec{l} = N_{exc} \cdot I_{exc}$.

Si mediante una fuente de tensión externa de corriente continua se alimenta al inducido (rotor), circulará una corriente por él. Al situarse en el interior de un campo magnético se producirán unas fuerzas según la Ley de Laplace:

$F = \int I i \cdot d\vec{l} \wedge \vec{B}_{exc}$, las cuales crearán un par que lo hará girar.

En bornes del inducido se produce una fuerza contraelectromotriz (f.c.e.m.), oponiéndose a la circulación de la corriente, por lo que el inducido absorbe energía eléctrica.

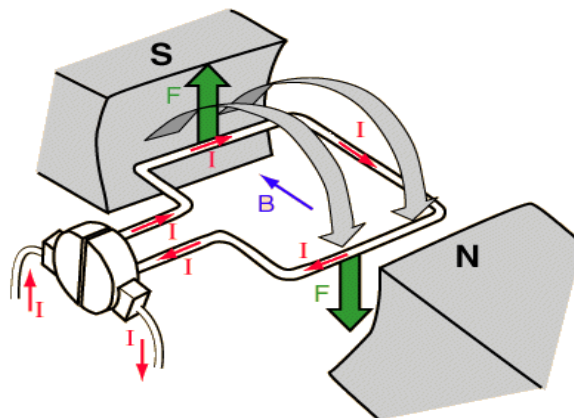


Figura 5. Fuerzas en una espira debido a un campo externo y una circulación de corriente por ella.
Fuente: [4]

En el caso de que la máquina funcione como **generador**, consumiría energía mecánica obtenida en el eje mediante una fuente externa. En bornes del estator se crearía una fuerza electromotriz (f.e.m.), la cual provocará circulación de corriente si se conectan a una carga externa, es decir, producirá energía eléctrica.

Las máquinas de corriente continua tienen distintas **configuraciones**, dependiendo de la conexión entre el inducido y el inductor.

Pueden ser:

De excitación independiente

El inductor y el inducido no están eléctricamente unidos, se alimentan con fuentes de tensión diferentes. Esto hace que el control de la velocidad de la máquina sea más sencillo, ya que se puede variar la corriente de excitación sin modificar la del inducido.

Se muestra a continuación el esquema de la configuración.

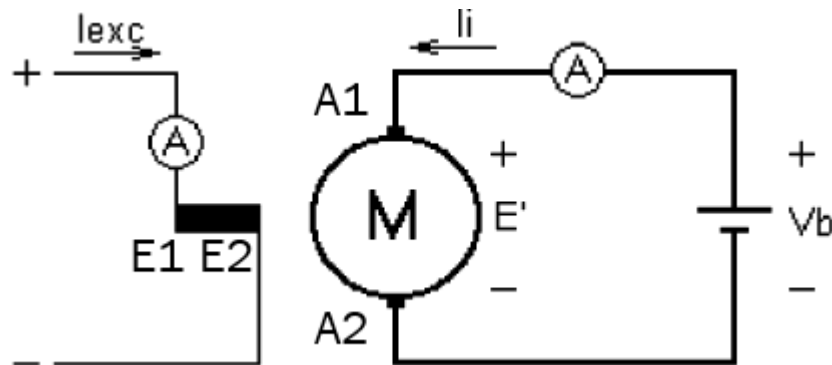


Figura 6. Modelo de motor con excitación independiente. Fuente: [5]

En las siguientes configuraciones, el inductor y el inducido se encuentran unidos eléctricamente. Son máquinas autoexcitadas, pues el inductor utiliza la energía de la misma fuente que alimenta al inducido.

De excitación serie

El inductor y el inducido se conectan en serie, por lo que circulará la misma corriente. Se podría variar la corriente de excitación si se colocara, por ejemplo, una resistencia variable en paralelo tal como se muestra en la siguiente figura

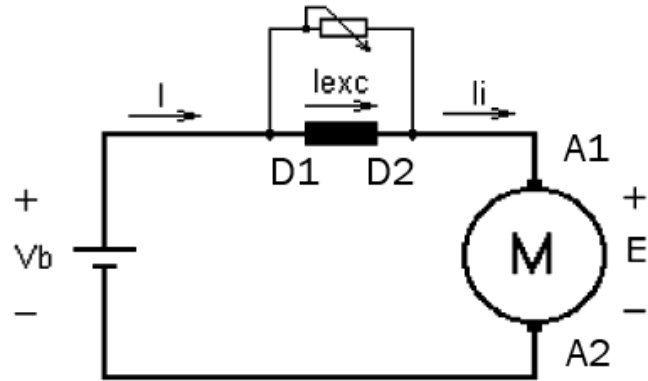


Figura 7. Modelo de motor con excitación serie. Fuente: [5]

Al estar conectados en serie, el devanado inductor deberá tener baja resistencia para disminuir lo máximo posible la caída de tensión en el mismo (bajo número de espiras y alta sección).

De excitación shunt o en derivación

Con la excitación shunt, el inductor y el inducido se conectan en paralelo. La tensión será prácticamente la misma, dependiendo de la resistencia del reóstato colocado en el inductor. Este deberá tener alta resistencia para que la corriente de excitación sea pequeña, por lo que tendrá, como el caso de la excitación independiente, gran número de espiras de baja sección.

El esquema es el siguiente:

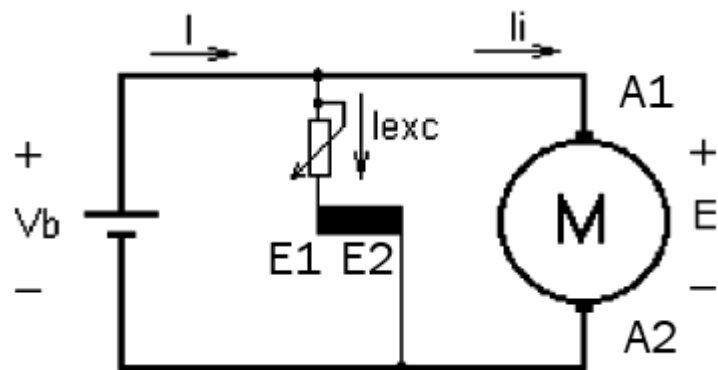


Figura 8. Modelo de motor con excitación shunt. Fuente: [5]

De excitación compuesta

En esta configuración se combinan las características de la excitación serie y shunt para obtener las ventajas de ambas configuraciones. Se pueden conectar de dos formas, dependiendo de la situación del devanado shunt con respecto a la del inducido.

Excitación compuesta larga, en la que el devanado shunt está alejado del inducido.

Excitación compuesta corta, en la que el devanado shunt está próximo al del inducido.

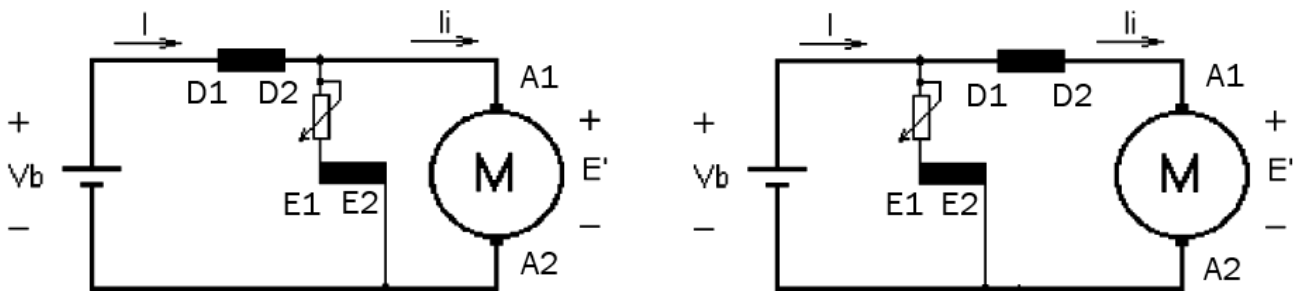


Figura 9. Modelo de motor con excitación compuesta corta y larga, respectivamente. Fuente: [5]

La principal característica de estas máquinas es la facilidad en la regulación de velocidad, ya que es directamente proporcional a la tensión aplicada en el inducido e inversamente proporcional al flujo de excitación (corriente de excitación):

$$E' \approx V_b = k \cdot n \cdot \phi_{exc} \rightarrow n \approx \frac{E'}{k \cdot \phi_{exc}}$$

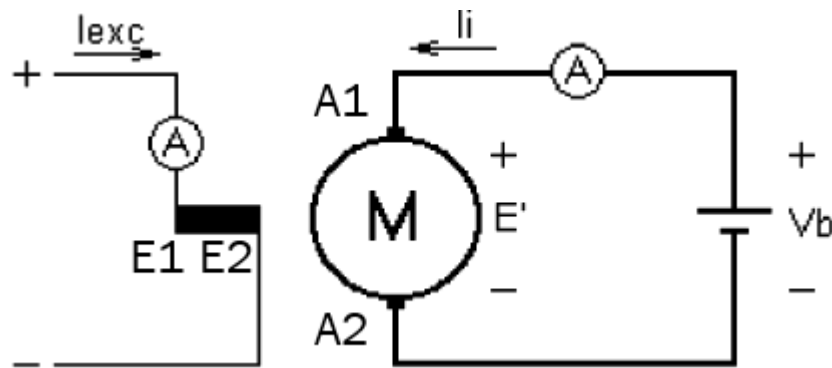


Figura 10. Modelo de motor de corriente continua básico. Fuente: [5]

Más exactamente, la fuerza contraelectromotriz (f.c.e.m.) viene dada por la siguiente expresión:

$$F. c. e. m. = E' = \frac{n \cdot N \cdot \phi_{exc}}{60} \cdot \frac{p}{a} = k \cdot n \cdot \phi_{exc}$$

Para hallar la relación entre la fuerza contraelectromotriz y la tensión aplicada al inducido, se aplican las Leyes de Kirchhoff al circuito del inducido:

$$E' = V_b - R_i \cdot I_i \approx V_b$$

Siendo:

V_b = Tensión aplicada al inducido.

R_i = Resistencia del devanado del inducido.

ϕ_{exc} = Flujo total por polo (Weber).

N = Número de conductores.

n = Velocidad del rotor (r.p.m.).

p = Pares de polos.

a = Pares de circuitos paralelos.

Estas máquinas suelen ser de rotor bobinado y de estator con polos salientes.

En la figura 11 se muestran las partes que conforman una máquina de corriente continua de este tipo.

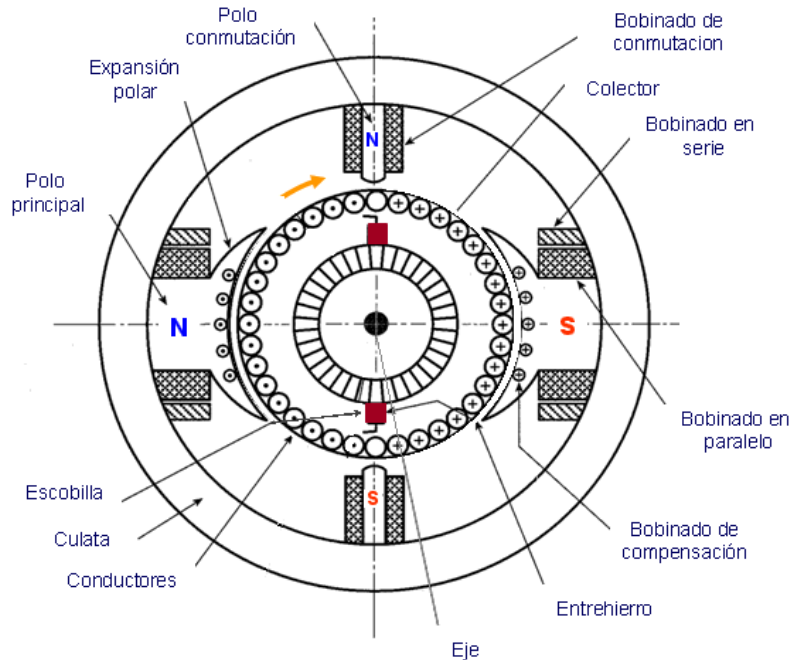


Figura 11. Elementos de una máquina de corriente continua. Fuente: [6]

El **rotor** tiene un devanado cerrado en el cual se inducen fuerzas electromotrices que varían en el tiempo, es decir, las f.e.m. inducidas son alternas. Por ello es necesario el **colector de delgas**.

El colector de delgas es un elemento que se encarga de rectificar esas f.e.m. para convertirlas en corriente continua.

Para conectar eléctricamente el estator y el rotor es necesario el uso de las **escobillas**, situadas en el estator, las cuales reciben la corriente de las delgas mediante un circuito externo.

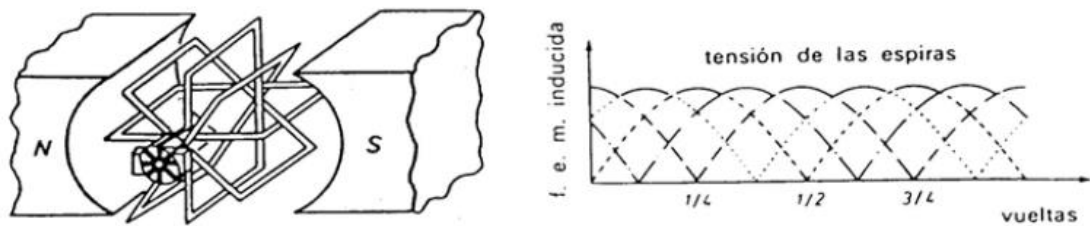


Figura 12. Rectificación de corriente alterna a corriente continua. Fuente: [7]

Existen otros tipos de máquinas de corriente continua que merecen ser destacadas: los motores brushless y los motores paso a paso.

Los **motores brushless** son aquellos que carecen de colector de delgas y de escobillas, por lo que se eliminan las pérdidas por rozamientos y disminuyen la generación de calor y ruido. En su lugar, la conmutación se realiza electrónicamente, tal como se observa en la figura 13.

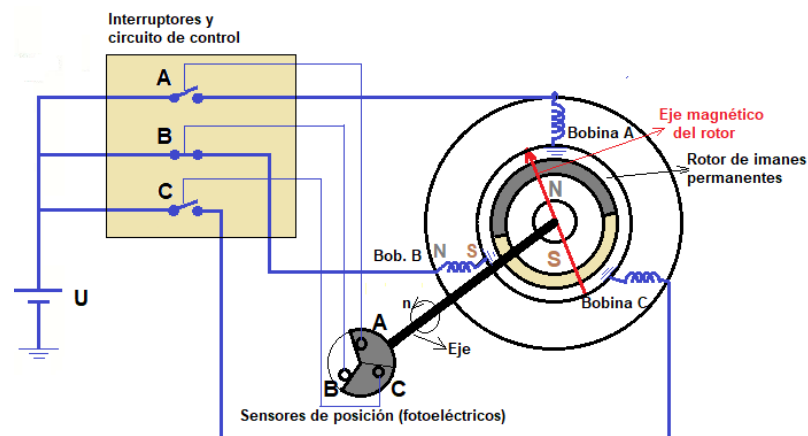


Figura 13. Esquema motor brushless. Fuente: [8]

El rotor está formado por un imán permanente. El estator dispone de devanados que serán alimentados de manera secuencial para crear un campo magnético giratorio en función de la posición del rotor.

Es necesario conocer en todo momento la posición del rotor para que la secuencia de alimentación del estator sea la correcta en cualquier posición del mismo.

Los motores brushless se pueden clasificar dependiendo de la posición del rotor, pueden ser de rotor en carcasa o de rotor en el eje.

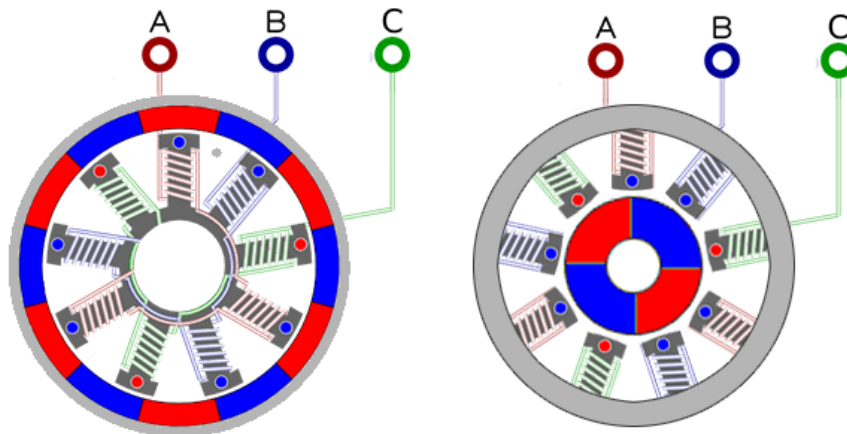


Figura 14. Tipos de motores brushless: rotor en la carcasa y rotor en el eje, respectivamente. Fuente: [9]

Los motores **paso a paso** se explican en el siguiente apartado con más detalle, ya que será el motor al que se someterá a estudio y se realizará el control correspondiente.

2.1.2. Motores paso a paso

2.1.2.1. Definición

Un motor paso a paso es un dispositivo electromecánico que convierte impulsos eléctricos en desplazamientos angulares discretos, es decir, gira una cantidad de grados determinada dependiendo de sus entradas de control.

Está basado en un estator construido por varios devanados en un material ferromagnético y un rotor que puede girar libremente en el estator. Estos diferentes devanados son alimentados uno a continuación del otro, con una secuencia determinada y originan un desplazamiento angular deseado (**ángulo de paso**), que es la principal característica de los motores paso a paso.

Dependiendo del motor, el paso puede variar desde 90° hasta incluso 1° , según la precisión requerida y la aplicación que esté realizando.

Otro parámetro importante es el **número de pasos por vuelta**, que es la cantidad de pasos necesarios para que el rotor realice un giro completo.

Para realizar el control se precisa de un circuito externo que distribuya la alimentación de forma secuencial a los arrollamientos del estator. El sistema de alimentación no será de corriente continua o alterna, sino que será una secuencia constante y uniforme de pulsos de excitación que se envían a los devanados. De esta forma la velocidad del motor dependerá de la frecuencia con la que se envíen los pulsos y del número de pares de polos del motor.

El giro del rotor se puede sincronizar con el envío de un tren de pulsos, suponiendo que no se pierdan pasos, lo que hace que el motor responda fielmente a la señal en lazo abierto.

Los motores paso a paso han surgido como alternativas rentables de los servomotores de corriente continua en aplicaciones de control de movimiento.

Los motores paso a paso se dividen en dos categorías: motor paso a paso de reluctancia variable y motor paso a paso de imanes permanentes, existiendo también el híbrido que combina las ventajas de ambos, aunque se considera de imanes permanentes.

2.1.2.2. Clasificación de los motores paso a paso según su estructura

- Motor paso a paso de imanes permanentes

El rotor está constituido por un imán permanente unido al eje. Gira cuando se aplica la tensión adecuada al bobinado, de manera que se obtenga un campo magnético que vaya girando en un determinado sentido.

El campo magnético del rotor tenderá a alinearse con el del estator, que variará según la secuencia de alimentación.

Al tener un imán permanente, cuando se deja de alimentar el bobinado la posición alcanzada del rotor tiende a mantenerse.

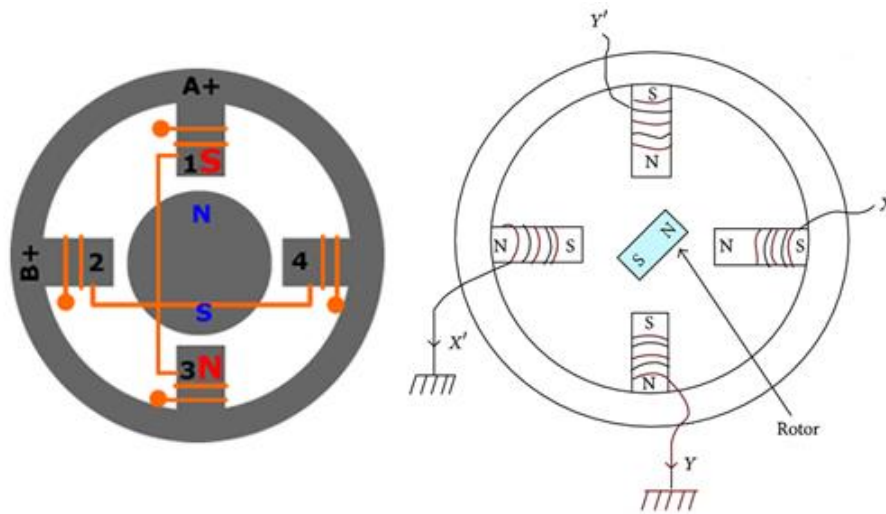


Figura 15. Motor paso a paso de imanes permanentes. Fuente: [8]

- Motor paso a paso de reluctancia variable

El rotor está constituido por un bloque de algún material magnéticamente blando con dientes salientes. Su giro se produce para facilitar un camino de menor permeabilidad magnética a los campos producidos por las bobinas estáticas, maximizando el flujo magnético.

Para producir un par significativo ambos deben tener un gran número de dientes para el tamaño que tiene el motor, con la condición de que el número de polos salientes del estator y del rotor no sean iguales para favorecer el movimiento de este. El desempeño del motor paso a paso depende de la fuerza del campo magnético. A mayor flujo, mayor par producirá.

Permite alcanzar velocidades mayores que los de imanes permanentes, aunque debido a la ausencia de campo magnético permanente del rotor, pierde la capacidad de mantener la posición al desconectar la alimentación.

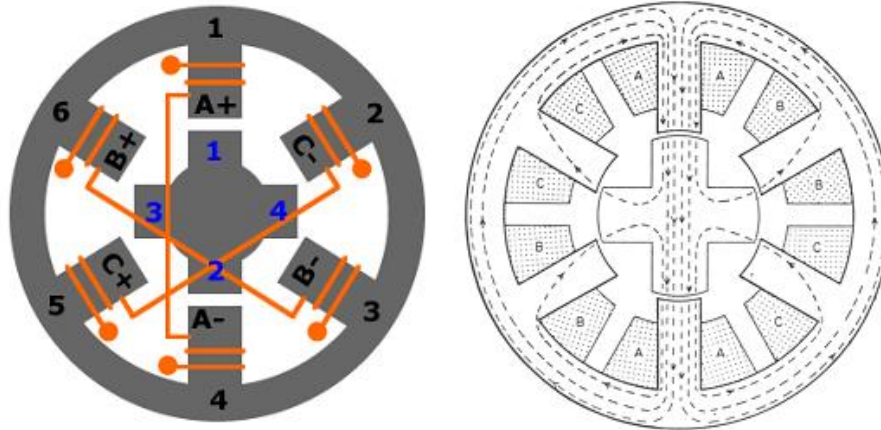


Figura 16. Motor paso a paso de reluctancia variable. Fuente: [8]

- **Motor paso a paso híbrido** (reluctancia variable + imán permanente)

Este es una variante del motor de reluctancia variable al que, con el objetivo de no requerir de ningún sistema mecánico para poder enclavarlo se le ha añadido un imán permanente en el rotor. Su ventaja es que combina los pros de los otros dos tipos, anclaje y velocidad, pero no llega a ser tan rápido como el motor de reluctancia variable.

Normalmente hay 8 polos del estator y cada polo tiene entre 2 y 6 dientes. Hay dos fases y cada fase está situada en 4 de los 8 polos del estator.

Para conseguir, por ejemplo, pasos de 1.8° , el rotor tendrá 50 dientes y el estator 48 (8 polos y 6 dientes por polo).

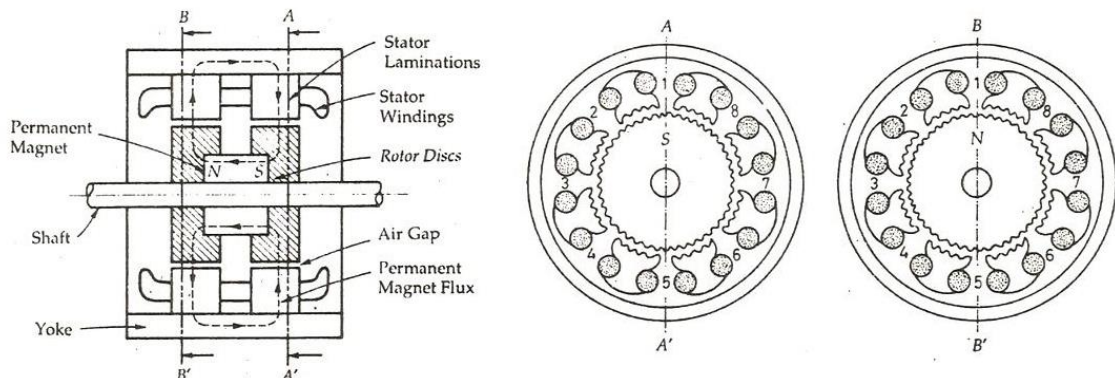


Figura 17. Motor paso a paso híbrido. Fuente: [37]

2.1.2.3. Control de motores paso a paso

Control en lazo abierto de un motor paso a paso

El giro del rotor se sincroniza con el tren de impulsos enviado.

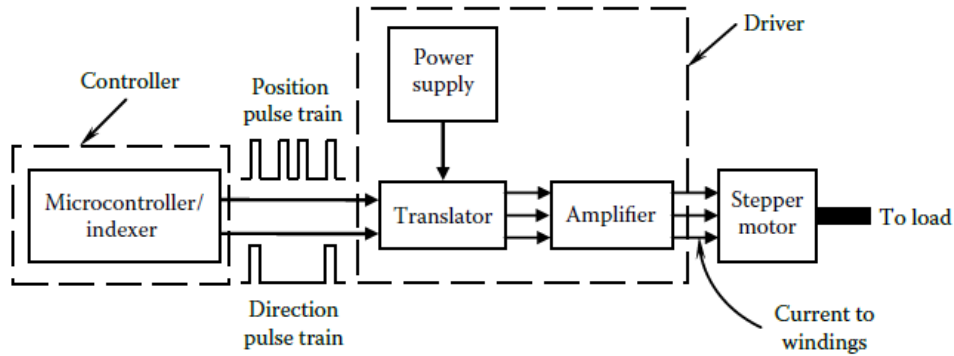


Figura 18. Esquema de control en lazo abierto. Fuente: [10]

El error de posición no es acumulativo, por lo que realizar un control en lazo abierto es buena opción. Una excepción sería cuando el motor trabajara en condiciones altamente transitorias cerca del par nominal, donde la pérdida de pulso podría ser un problema.

Componentes básicos para el control de lazo abierto de un motor paso a paso

- **Generador de pulsos:** generalmente un microcontrolador. Envía el tren de pulsos necesario para la aplicación requerida.
- **Módulo traductor:** formado por circuitos lógicos para interpretar el tren de pulsos y traducirlo a la secuencia de conmutación correspondiente.
- **Amplificadores:** las señales de control dentro del traductor son del orden de 10 mA y la excitación de las fases del motor requiere corrientes mayores, por ello se utilizan amplificadores o etapas de potencia a la hora de diseñar un control de este tipo.
- **Fuentes de alimentación:** la potencia para la excitación del motor y para que opere la electrónica proviene de una fuente de alimentación de corriente continua.

Control en lazo cerrado de un motor paso a paso

El control en **lazo abierto** es adecuado para muchas aplicaciones de motores paso a paso, particularmente a **bajas velocidades** y en operación en estado estacionario.

La principal desventaja del control en lazo abierto es que no se mide la respuesta real del motor, no se conoce si existe un error significativo debido a **pulsos perdidos**.

Principales **razones** por las que ocurre una falta de pulso:

- Si los pulsos se reciben a una **frecuencia alta** (velocidad de paso alta), el traductor podría no responder a un pulso particular debido a un mal funcionamiento y la fase correspondiente no se energizaría antes de que llegue el siguiente pulso.
- Debido al **mal funcionamiento** de una fuente es posible que no se genere un pulso, incluso cuando el motor funciona a una capacidad inferior a la nominal (en condiciones de bajo par y baja velocidad). También se pueden generar pulsos erróneos debido al mal funcionamiento de los circuitos de accionamiento.

Si se pierde un pulso, la respuesta tiene que alcanzarle de alguna forma o, de lo contrario, puede producirse un comportamiento no deseado que hará que el rotor oscile y se bloquee finalmente. En general, la pérdida de pulso puede provocar el paro del motor o una respuesta sin sincronía con el tren de pulsos.

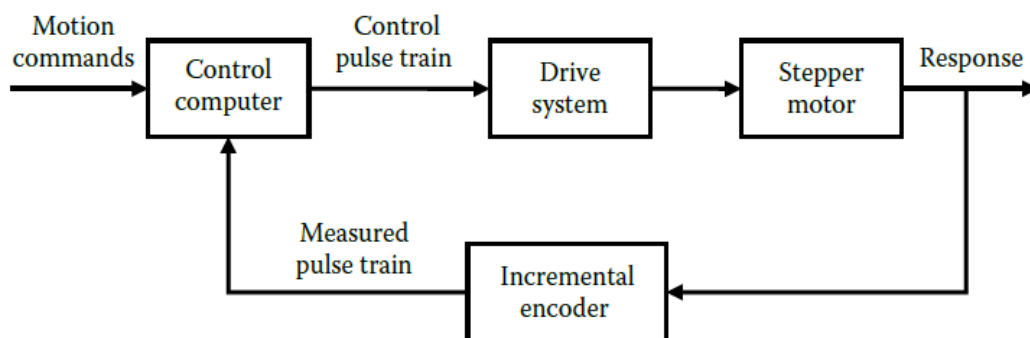


Figura 19. Esquema de control en lazo cerrado. Fuente: [10]

En la siguiente figura se observa las consecuencias de una pérdida de pulso. El rotor tendría que recibir un pulso al llegar a B, pero ha seguido desplazándose con par negativo y en B' ha recibido el siguiente pulso en vez de recibirlo en C. De B' llegará a C' y así se ha perdido el sincronismo con los pulsos.

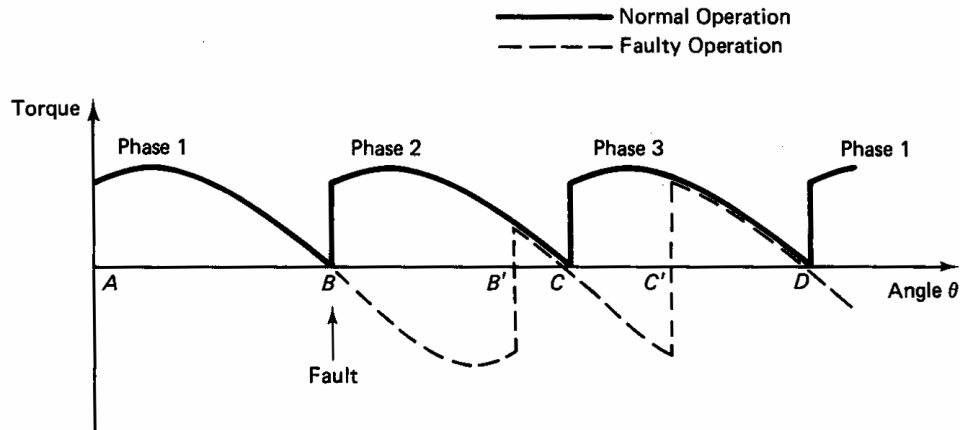


Figura 20. Efectos en el par debido a la pérdida de pulso. Fuente: [11]

La falta de pulso provocará que el motor desacelere debido al **par negativo** de la fase que no se desalimentó. Dependiendo de la sincronización de los pulsos siguientes, puede seguir existiendo un par negativo en el motor, por lo se detendrá.

Para evitar estos problemas, se debe detectar la falta de pulso mediante la **medición de la respuesta**, en este caso midiendo en tiempo real la posición del rotor con un encoder, y así modificar adecuadamente la secuencia de conmutación futura para acelerar el motor y seguir la trayectoria deseada.

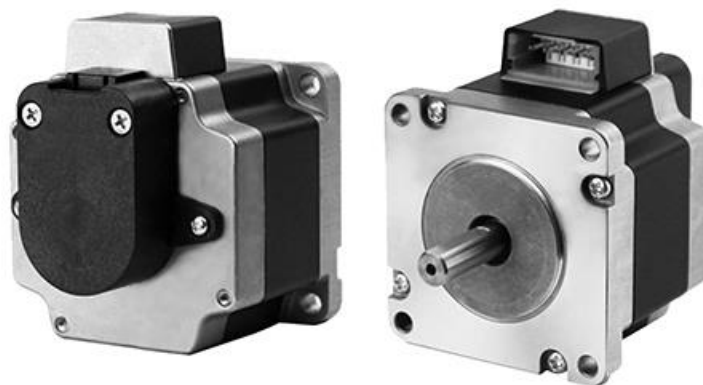


Figura 21. Motor paso a paso con encoder incorporado. Fuente: [13]

Cuando se emplea el control en lazo cerrado, se puede operar cerca de la capacidad nominal (par, velocidad, aceleración...) del motor paso a paso, incluso excediendo estos valores, sin introducir errores ni problemas de estabilidad.

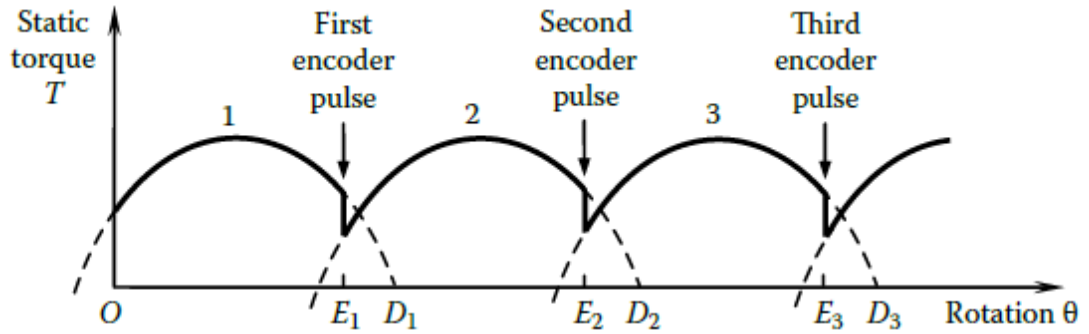


Figura 22. Operación en lazo cerrado con encoder. Fuente: [10]

En la figura 23 se muestra un ejemplo de cómo se realizaría la conmutación de forma electrónica ideal, conociendo en tiempo real todos los parámetros.

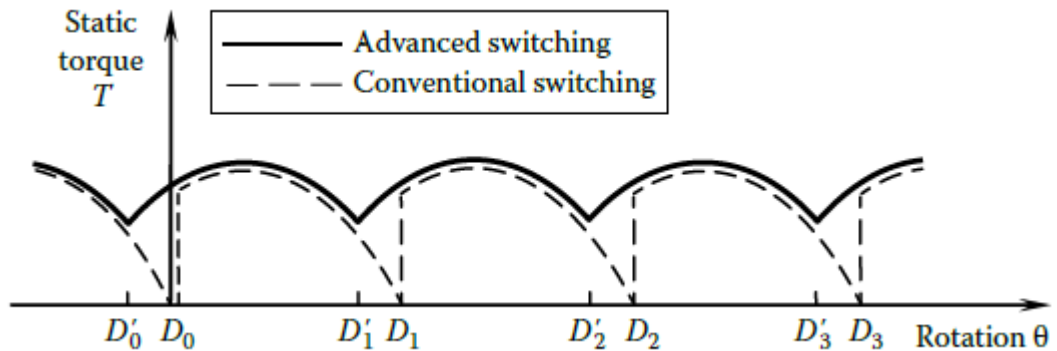


Figura 23. Conmutación avanzada vs convencional. Fuente: [10]

Existen **tres modos de realizar el control**: paso entero, medio paso y micropaso.

En el **paso entero**, el motor avanza el ángulo de paso nominal. Se puede distinguir entre paso entero con una bobina activa o con dos bobinas activas.

Con una bobina activa: por cada paso se activa una bobina.

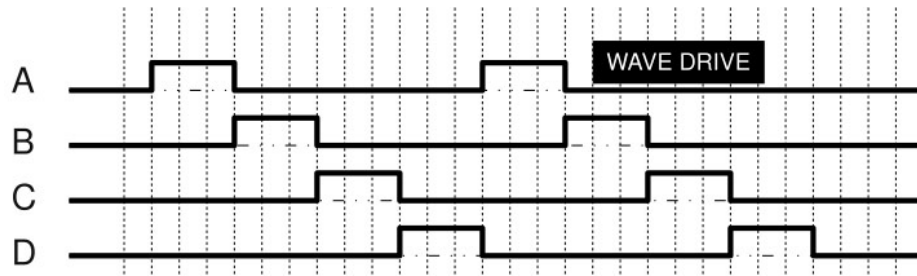


Figura 24. Paso completo con una bobina activa de un motor unipolar. Fuente: [14]

Con dos bobinas activas: por cada paso se activan dos bobinas.

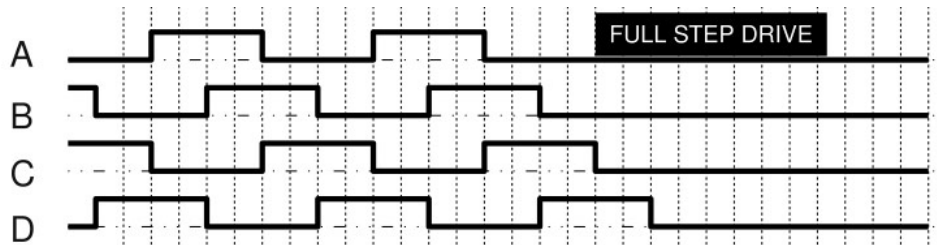


Figura 25. Paso completo con dos bobinas activas de un motor unipolar. Fuente: [14]

Ambas formas nos proporcionan un paso completo, aunque tienen algunas diferencias:

Al tener dos bobinas activas, aumenta el flujo de corriente y, por lo tanto, el par de salida del motor paso a paso es mayor que con una bobina activa. El inconveniente es que tiene mayor consumo al alimentar a dos bobinas por cada paso.

Además, la respuesta con dos bobinas activas es ligeramente mejor que con una sola, el rotor se estabiliza antes (menos oscilaciones).

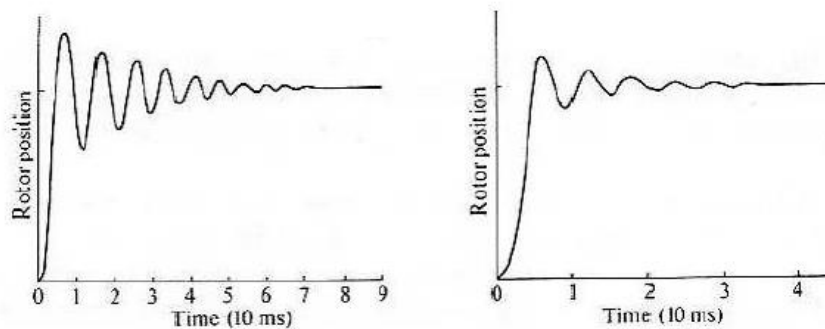


Figura 26. Diferencia en la respuesta ante un paso entre una bobina activa y dos bobinas activas. Fuente: [12]

En el **medio paso** se avanza la mitad del paso nominal, con lo que se dispone de mejor resolución. Combina la secuencia de los dos anteriores, tal y como se observa en la siguiente imagen.

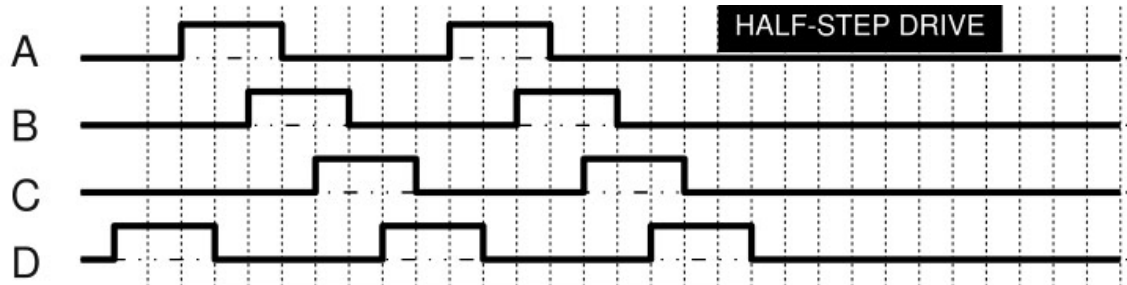


Figura 27. Medio paso en motor unipolar. Fuente: [14]

Otra forma de control es el **micropaso**. Consiste en alimentar al mismo tiempo **varias fases** a la vez con **corrientes medias distintas**, la activación de la bobina 1 iría decayendo suavemente siguiendo una curva senoidal, mientras que la activación de la bobina 2 iría aumentando de la misma forma, pero en proporciones diferentes.

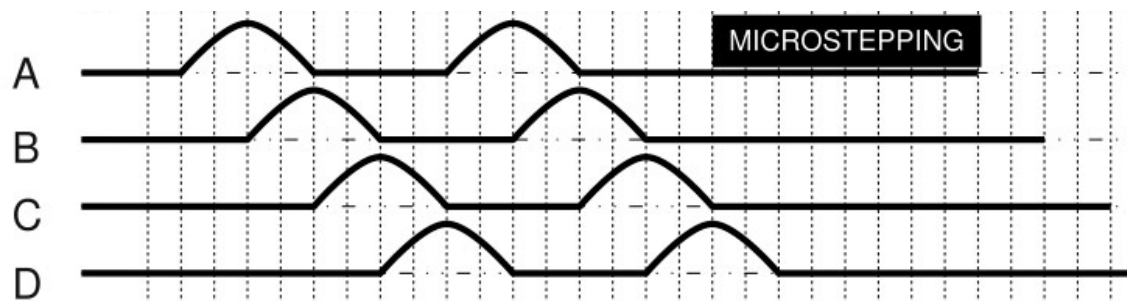


Figura 28. Micropaso en motor unipolar. Fuente: [14]

El principio detrás de este modo de control puede entenderse considerando dos polos del estator enrollados con bobinados idénticos. Cuando las corrientes a través de los devanados son idénticas en magnitud y dirección, el campo magnético resultante se ubicará de manera simétrica entre los dos polos. Si la corriente en un polo disminuye mientras la otra corriente se mantiene sin cambios, el campo magnético resultante se moverá más cerca del polo con la corriente mayor.

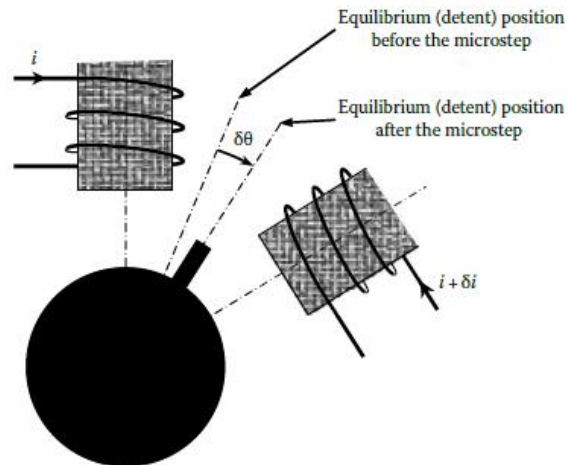


Figura 29. Principio de funcionamiento del micropaso. Fuente: [10]

Dado que la posición de equilibrio depende de la posición del campo magnético resultante, se pueden lograr ángulos de paso muy pequeños controlando las corrientes de fase.

Mediante micropasos pequeños, como por ejemplo de $1/32$ del paso completo, se puede lograr un movimiento más suave, pero la desventaja es la disminución de la precisión en cuanto a la posición del rotor. El error de posicionamiento es mínimo cuando el rotor se ubica en un polo, y máximo cuando se posiciona entre polos.

Este modo combina perfectamente con un control en lazo cerrado para disminuir los errores de posición y corregirlos en tiempo real.

El uso de micropasos se da en aplicaciones en las que se necesita alta precisión en el posicionamiento y bajas vibraciones y ruido.

Clasificación de los motores paso a paso según la disposición de sus bobinados

Unipolares (Punto neutro accesible)

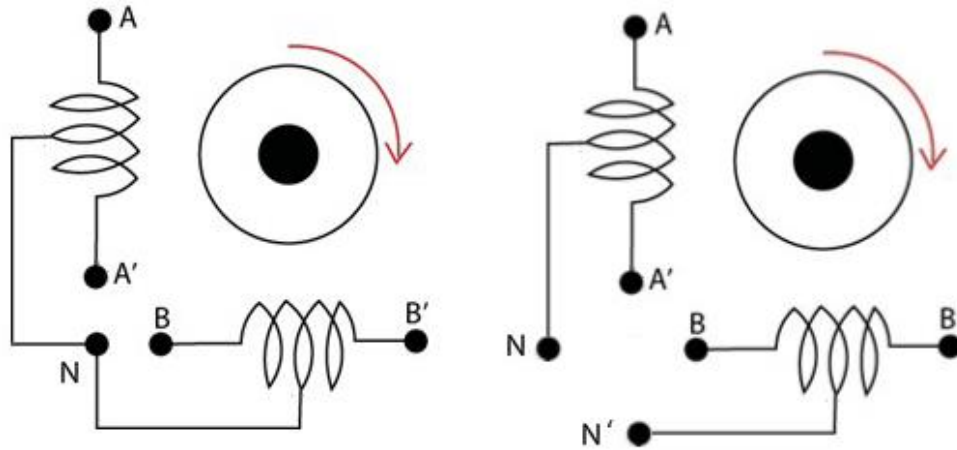


Figura 30. Motor paso a paso unipolar con 5 y 6 terminales accesibles. Fuente: [8]

Este modelo de motor paso a paso tiene accesible el punto medio de las bobinas (6 hilos disponibles), lo más habitual es que los puntos medios estén unidos (5 hilos disponibles).

La regulación es más sencilla pero tan sólo se aprovecha la mitad del bobinado.

Esquema del control:

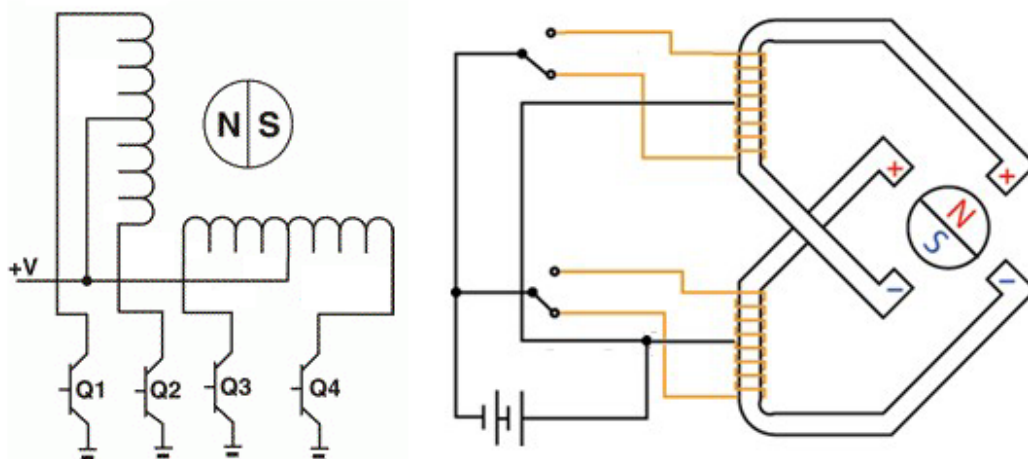


Figura 31. Control del motor paso a paso unipolar. Fuente: [15]

Q1, Q2, Q3 Y Q4 son transistores que alimentarán cada mitad de bobinado dependiendo de las señales que reciban. A continuación, se muestran las secuencias necesarias para el control con paso completo y medio paso y

sentido horario y antihorario que se realizarán en el diseño del sistema de control.

Paso Completo

- Una bobina activa

Horario

PASO	A	A'	B	B'
1	ON	OFF	OFF	OFF
2	OFF	OFF	OFF	ON
3	OFF	ON	OFF	OFF
4	OFF	OFF	ON	OFF

Tabla 1. Paso completo, una bobina activa, horario I. Fuente: Elaboración propia.

Secuencia:

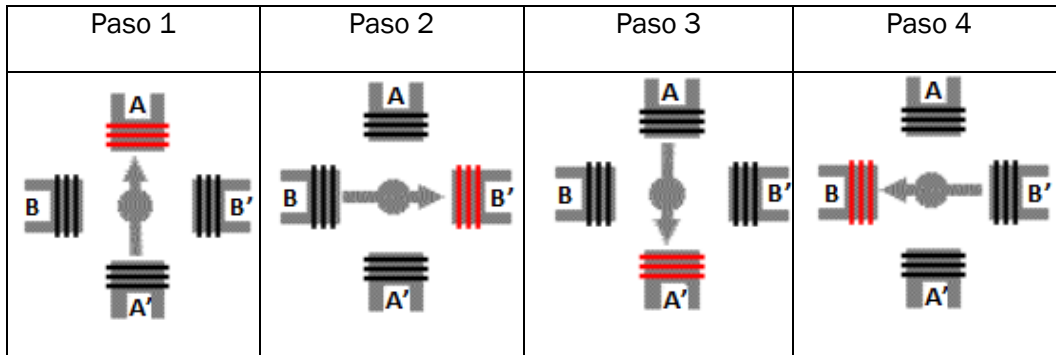


Tabla 2. Paso completo, una bobina activa, horario II. Fuente: [16]

Antihorario

PASO	A	A'	B	B'
1	ON	OFF	OFF	OFF
2	OFF	OFF	ON	OFF
3	OFF	ON	OFF	OFF
4	OFF	OFF	OFF	ON

Tabla 3. Paso completo, una bobina activa, antihorario I. Fuente: Elaboración propia.

Secuencia:

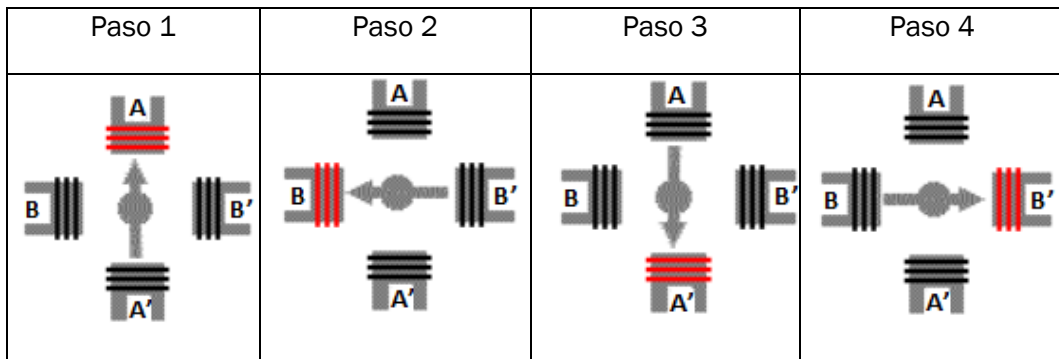


Tabla 4. Paso completo, una bobina activa, antihorario II. Fuente: [16]

· Dos bobinas activas

Horario

PASO	A	A'	B	B'
1	ON	OFF	OFF	ON
2	OFF	ON	OFF	ON
3	OFF	ON	ON	OFF
4	ON	OFF	ON	OFF

Tabla 5. Paso completo, dos bobinas activas, horario I. Fuente: Elaboración propia.

Secuencia:

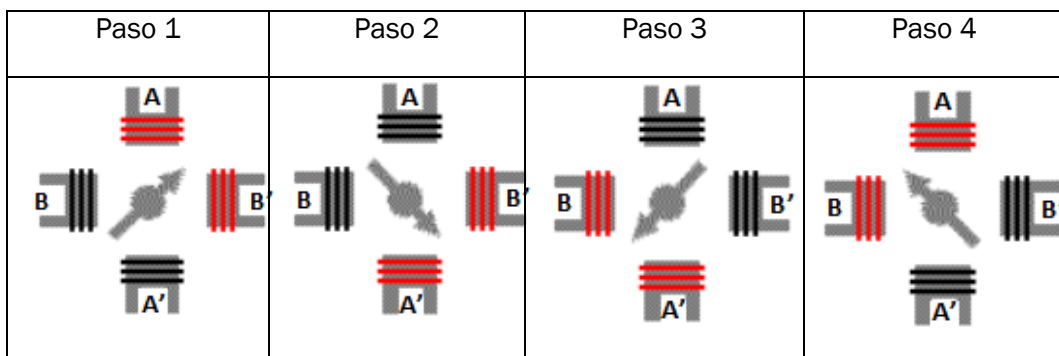


Tabla 6. Paso completo, dos bobinas activas, horario II. Fuente: [16]

Antihorario

PASO	A	A'	B	B'
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON

Tabla 7. Paso completo, dos bobinas activas, antihorario I. Fuente: Elaboración propia.

Secuencia:

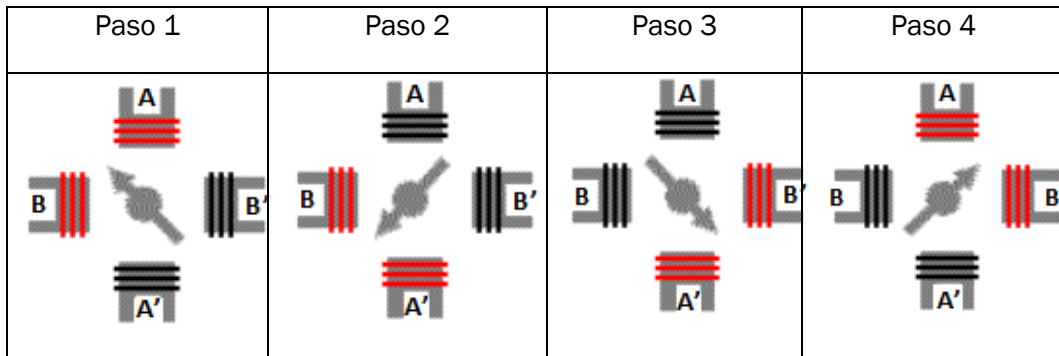


Tabla 8. Paso completo, dos bobinas activas, antihorario II. Fuente: [16]

Medio paso

Horario

PASO	A	A'	B	B'
1	ON	OFF	OFF	OFF
2	ON	OFF	OFF	ON
3	OFF	OFF	OFF	ON
4	OFF	ON	OFF	ON
5	OFF	ON	OFF	OFF
6	OFF	ON	ON	OFF
7	OFF	OFF	ON	OFF
8	ON	OFF	ON	OFF

Tabla 9. Medio paso, horario I. Fuente: Elaboración propia.

Secuencia:

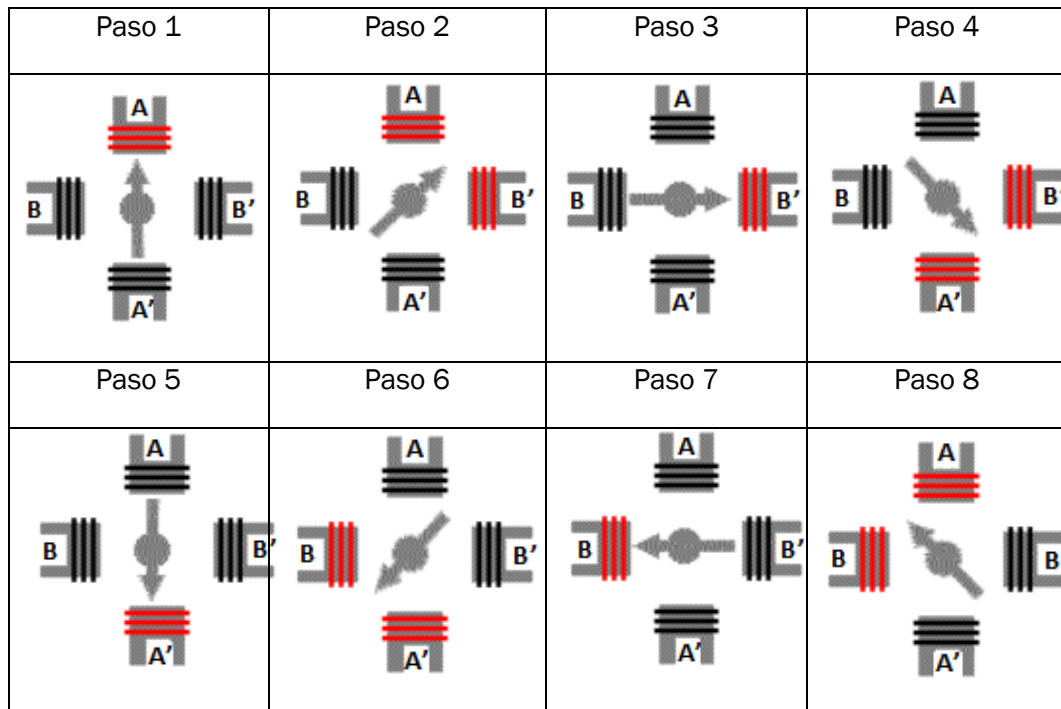


Tabla 10. Medio paso, horario II. Fuente: [16]

Antihorario

PASO	A	A'	B	B'
1	ON	OFF	OFF	OFF
2	ON	OFF	ON	OFF
3	OFF	OFF	ON	OFF
4	OFF	ON	ON	OFF
5	OFF	ON	OFF	OFF
6	OFF	ON	OFF	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

Tabla 11. Medio paso, antihorario I. Fuente: Elaboración propia.

Secuencia:

Paso 1	Paso 2	Paso 3	Paso 4
Paso 5	Paso 6	Paso 7	Paso 8

Tabla 12. Medio paso, antihorario II. Fuente: [16]

Bipolares (Bobinados independientes)

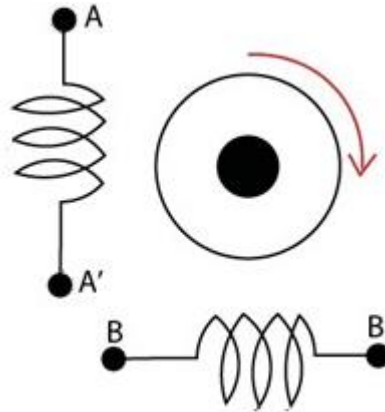


Figura 32. Motor paso a paso bipolar. Fuente: [8]

Este modelo de motor paso a paso tiene accesible tan sólo los extremos de las bobinas (4 hilos disponibles).

Se aprovecha la totalidad del bobinado, pero la regulación es más compleja.

Esquema del control:

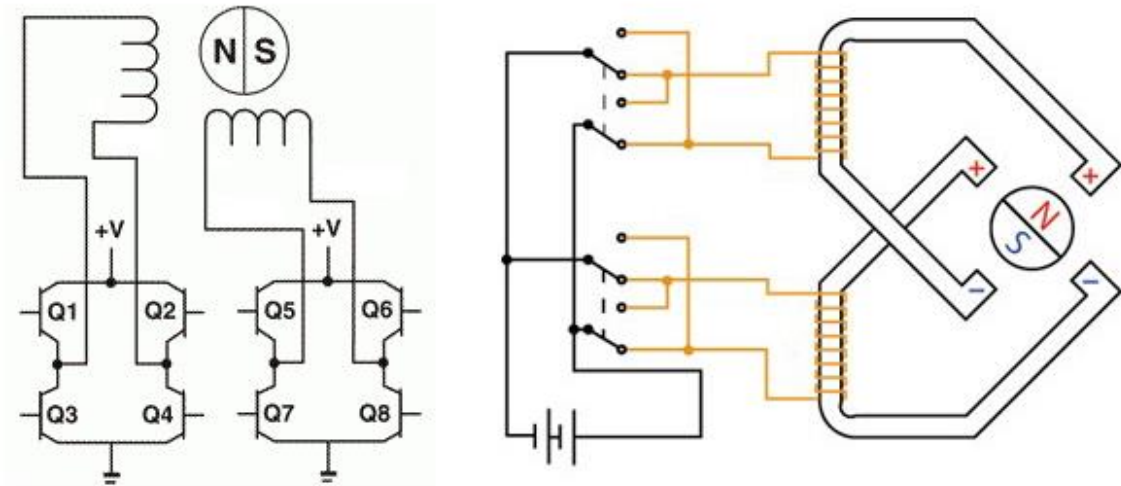


Figura 33. Control del motor paso a paso bipolar. Fuente: [15]

El control del motor paso a paso bipolar es más complejo que el unipolar, ya que es necesario incluir dos puentes H para alimentar a cada bobina. Dichos puentes están compuestos por 4 transistores.



A continuación, se muestran las secuencias necesarias para el control con paso completo y medio paso y sentido horario y antihorario que se podrían realizar en el diseño del sistema de control.

Paso Completo

- Una bobina activa

Horario

PASO	A	A'	B	B'
1	+V	GND	-	-
2	-	-	GND	+V
3	GND	+V	-	-
4	-	-	+V	GND

Tabla 13. Paso completo, una bobina activa, horario I. Fuente: Elaboración propia.

Secuencia:

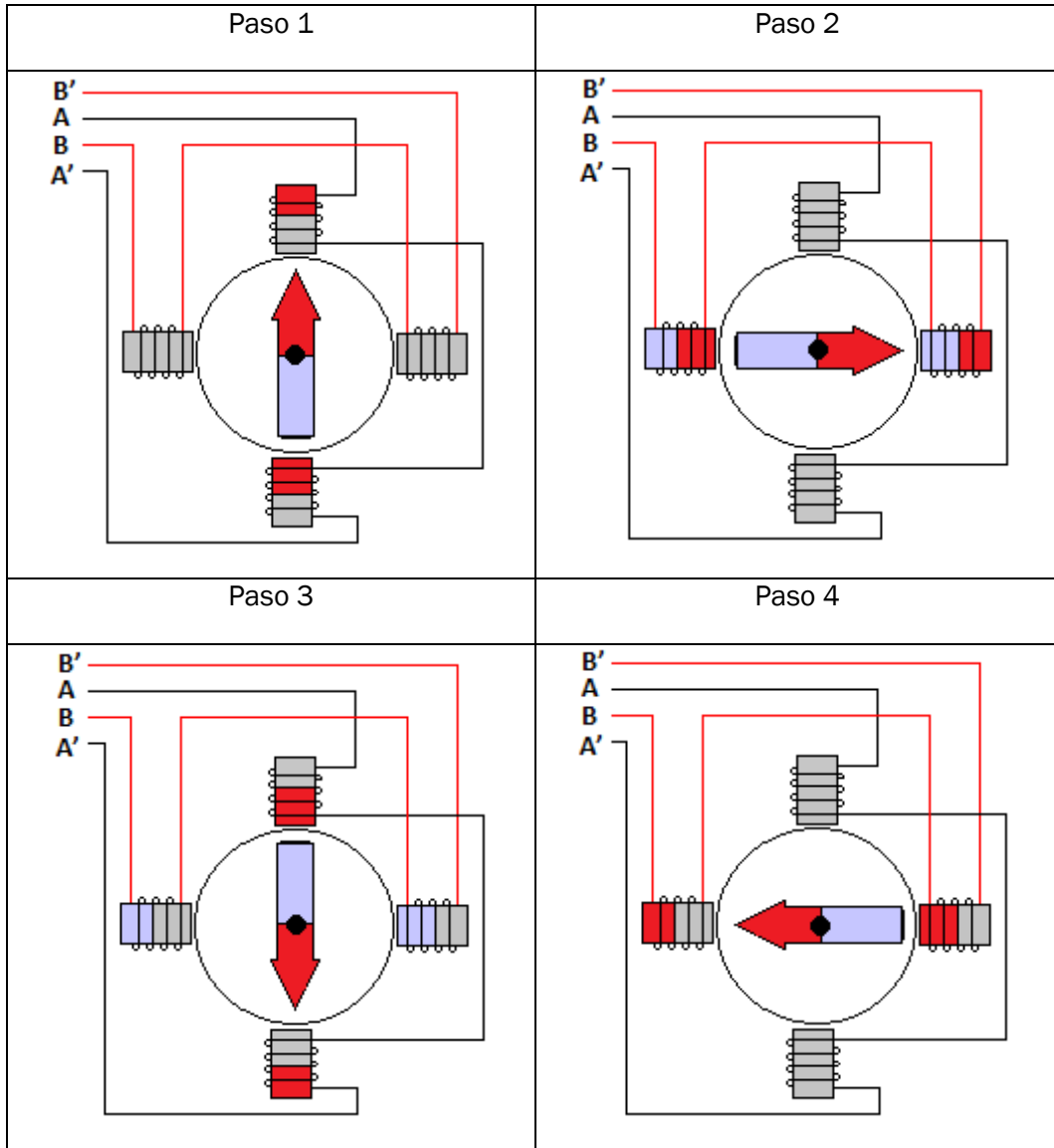


Tabla 14. Paso completo, una bobina activa, horario II. Fuente: [16]

Antihorario

PASO	A	A'	B	B'
1	+V	GND	-	-
2	-	-	+V	GND
3	GND	+V	-	-
4	-	-	GND	+V

Tabla 15. Paso completo, una bobina activa, antihorario I. Fuente: Elaboración propia.

Secuencia:

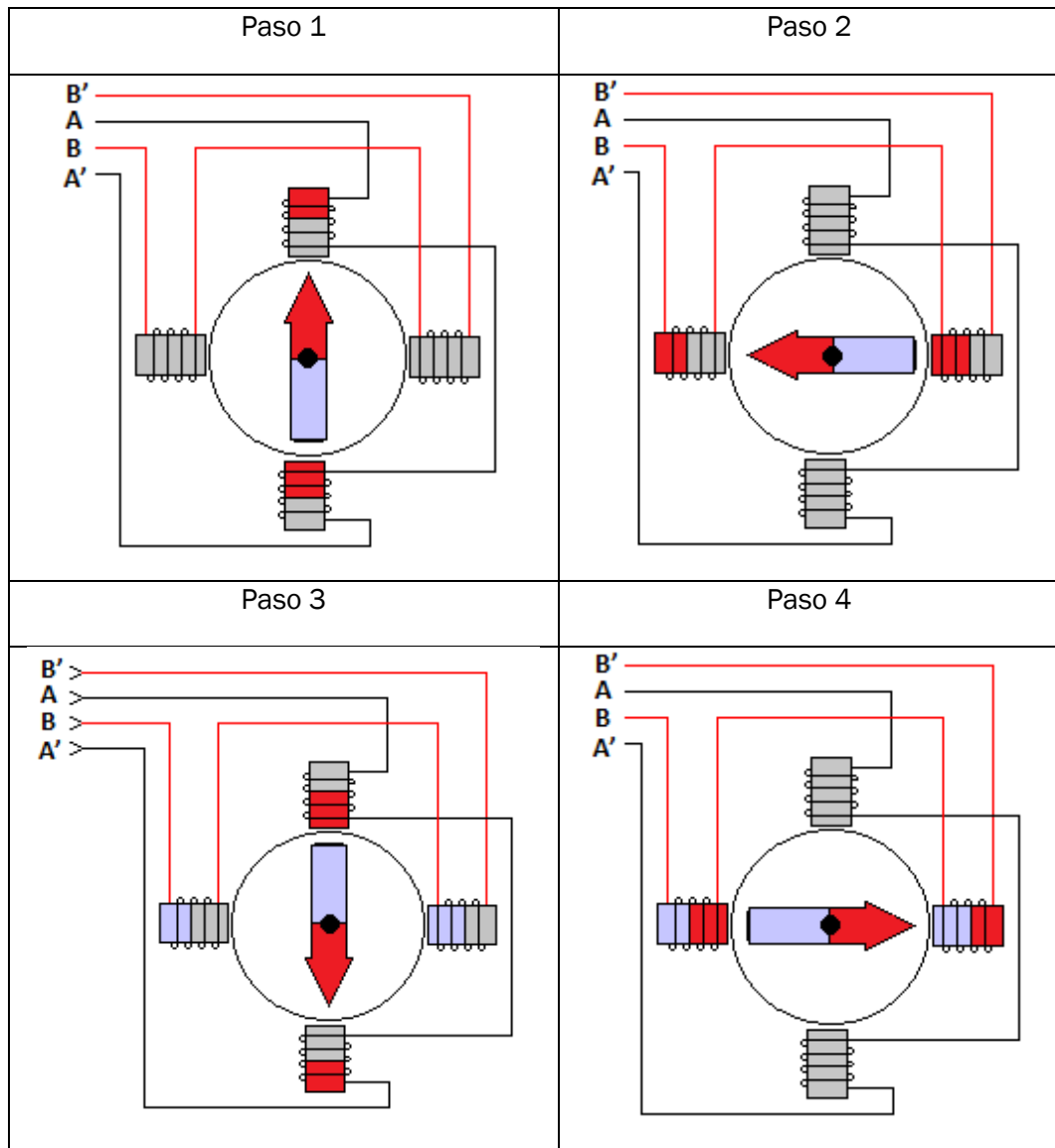


Tabla 16. Paso completo, una bobina activa, antihorario II. Fuente: [16]

· Dos bobinas activas

Horario

PASO	A	A'	B	B'
1	+V	GND	GND	+V
2	GND	+V	GND	+V
3	GND	+V	+V	GND
4	+V	GND	+V	GND

Tabla 17. Paso completo, dos bobinas activas, horario I. Fuente: Elaboración propia.

Secuencia:

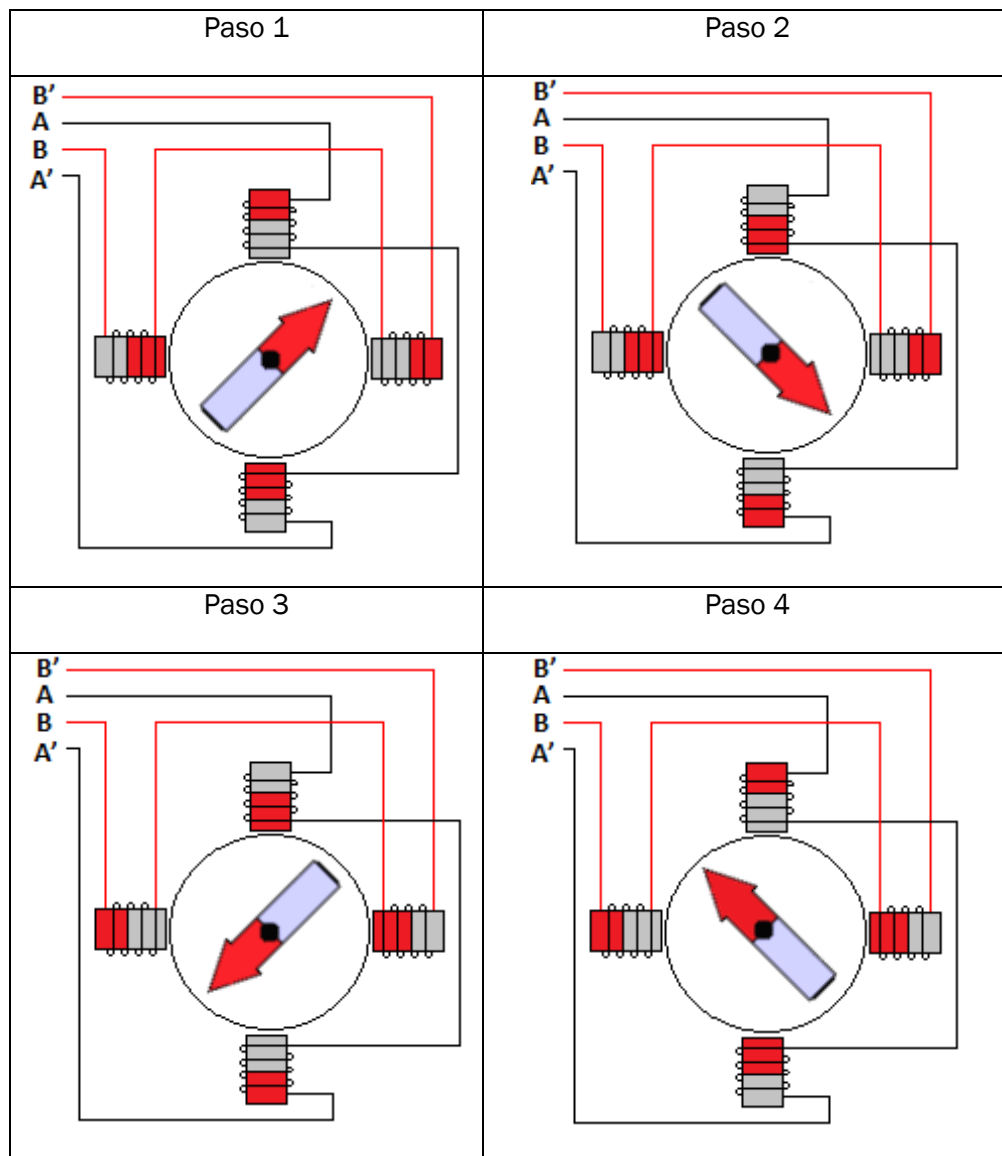


Tabla 18. Paso completo, dos bobinas activas, horario II. Fuente: [16]

Antihorario

PASO	A	A'	B	B'
1	+V	GND	+V	GND
2	GND	+V	+V	GND
3	GND	+V	GND	+V
4	+V	GND	GND	+V

Tabla 19. Paso completo, dos bobinas activas, antihorario I. Fuente: Elaboración propia.

Secuencia:

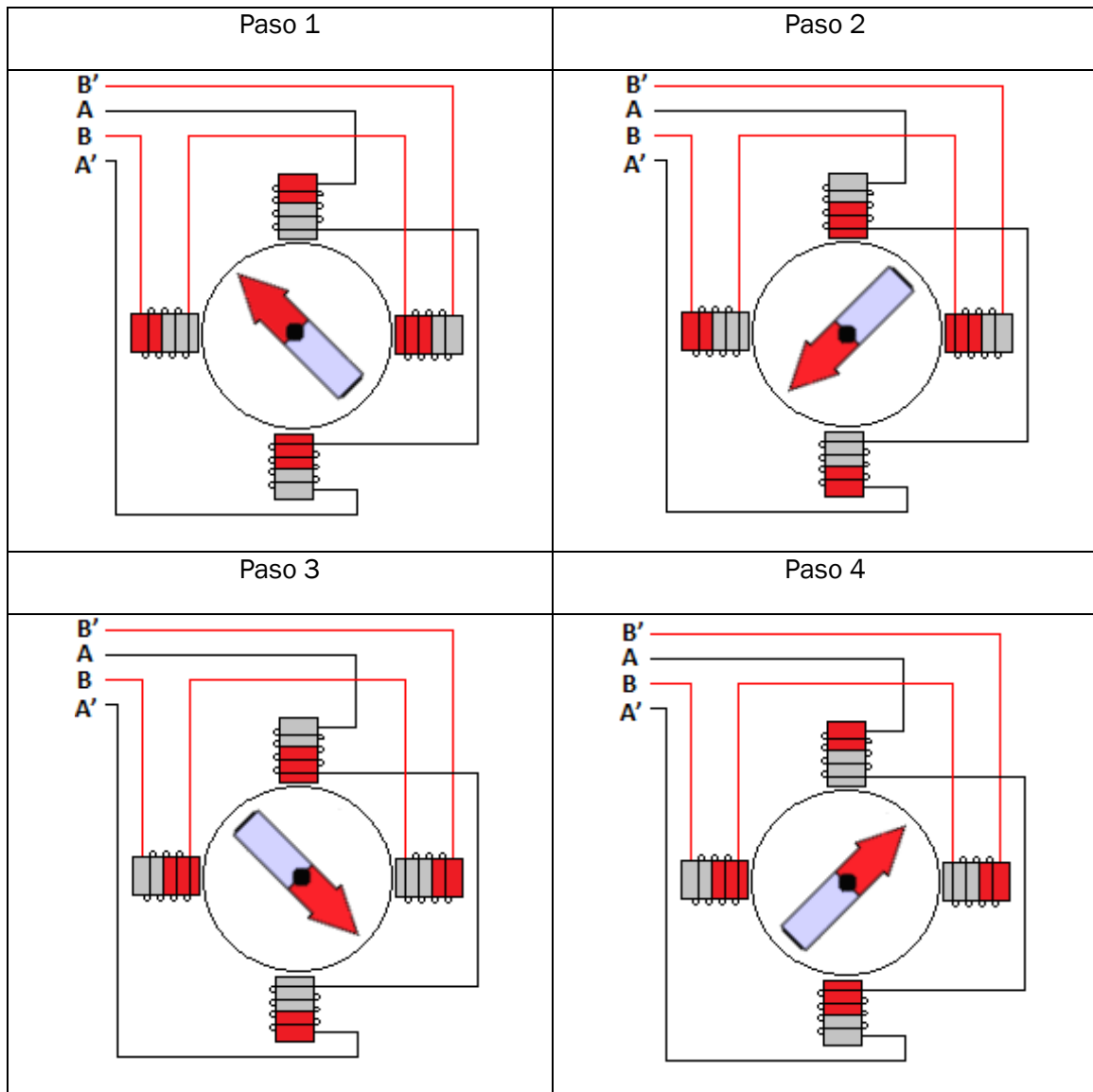


Tabla 20. Paso completo, dos bobinas activas, antihorario II. Fuente: [16].



Medio paso

Horario

PASO	A	A'	B	B'
1	+V	GND	-	-
2	+V	GND	GND	+V
3	-	-	GND	+V
4	GND	+V	GND	+V
5	GND	+V	-	-
6	GND	+V	+V	GND
7	-	-	+V	GND
8	+V	GND	+V	GND

Tabla 21. Medio paso, una bobina activa, horario I. Fuente: Elaboración propia.

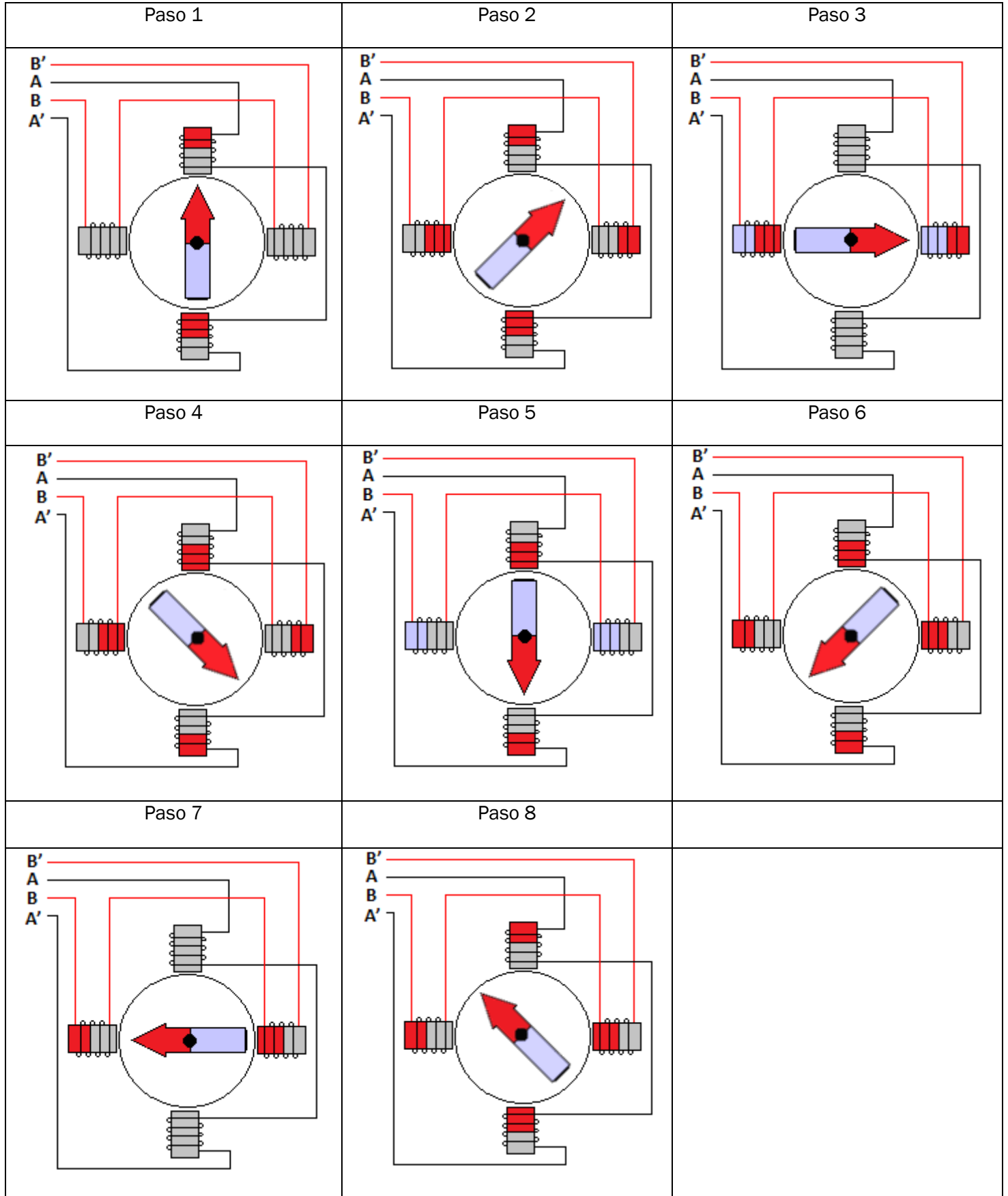


Tabla 22. Medio paso, una bobina activa, horario II. Fuente: [16]



Antihorario

PASO	A	A'	B	B'
1	+V	GND	-	-
2	+V	GND	+V	GND
3	-	-	+V	GND
4	GND	+V	+V	GND
5	GND	+V	-	-
6	GND	+V	GND	+V
7	-	-	GND	+V
8	+V	GND	GND	+V

Tabla 23. Medio paso, una bobina activa, antihorario I. Fuente: Elaboración propia.

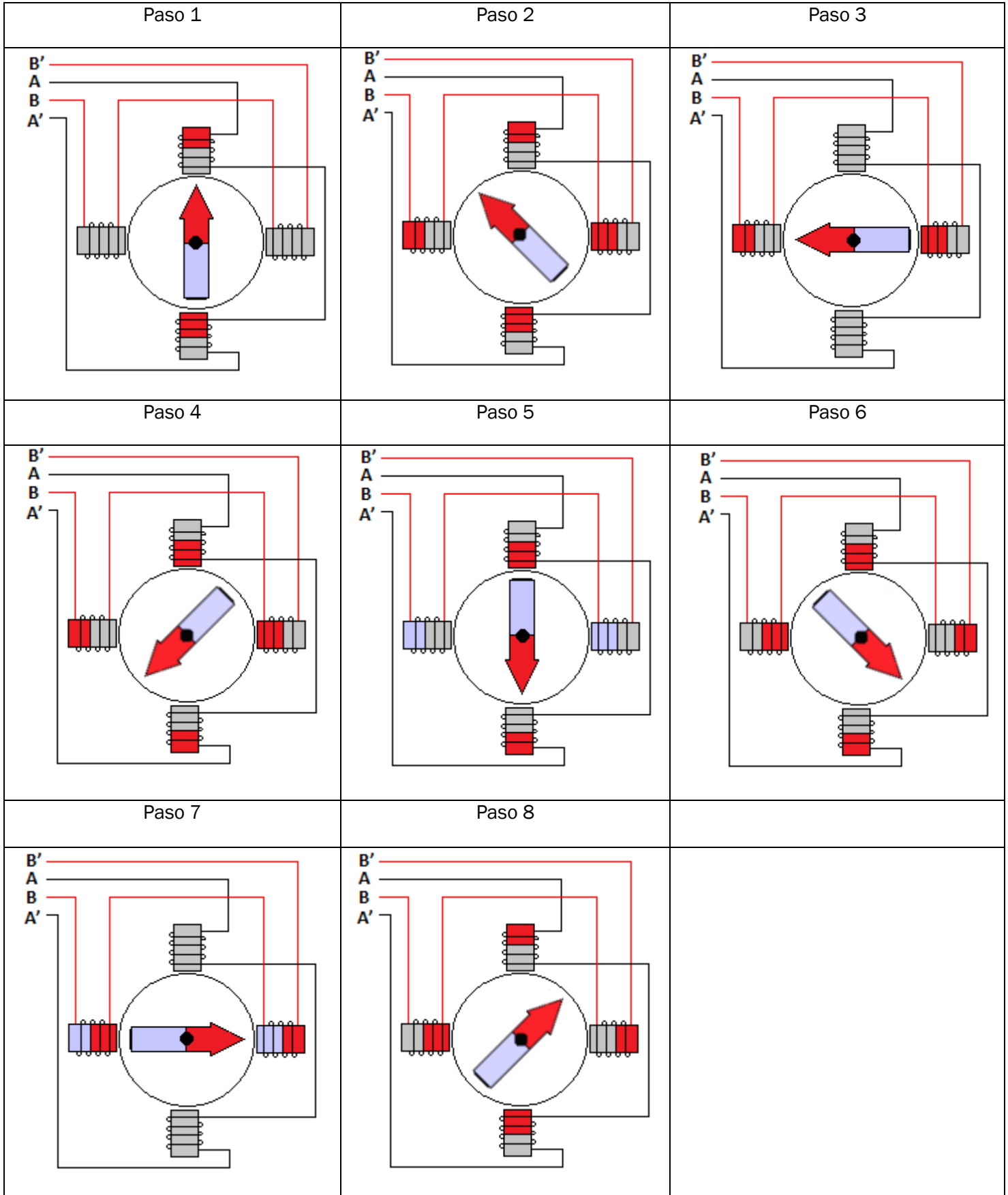


Tabla 24. Medio paso, una bobina activa, antihorario II. Fuente: [16]

2.1.2.4. Características de los motores paso a paso

Característica par velocidad

Un motor paso a paso tiene una curva par-velocidad como la de la figura 34.

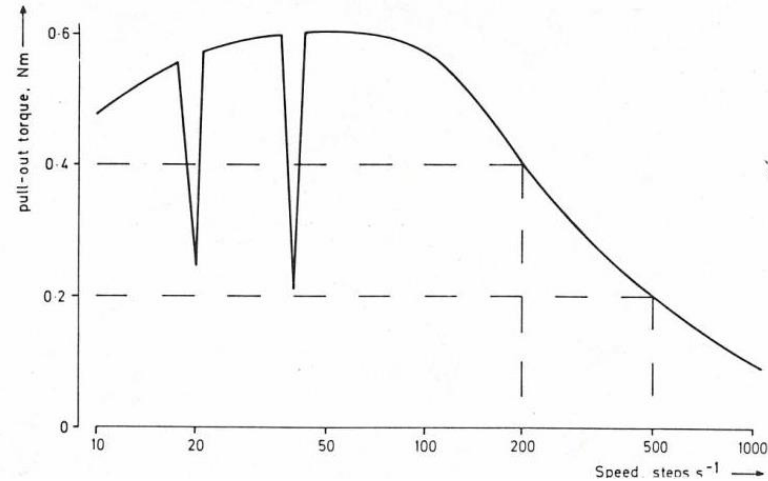


Figura 34. Curva típica par velocidad de un motor paso a paso. Fuente: [11]

Se distinguen dos zonas:

Zona de baja velocidad: cuando se activa una fase, la corriente por el devanado se estabiliza y permanece cerca de su valor nominal durante la mayor parte del tiempo durante el cual se excita la fase. En el ejemplo se observa con una velocidad menor a 100 pasos/segundo.

Zona de alta velocidad: la constante de tiempo se convierte en una variable importante. La corriente de fase no permanece en su valor nominal durante la mayor parte del tiempo, necesita un tiempo para estabilizarse y, por lo tanto, el par producido por el motor se reducirá. En el ejemplo se observa con una velocidad mayor a 100 pasos/segundo.

Más adelante se explica la importancia de la constante de tiempo eléctrica en el correcto funcionamiento de los motores paso a paso.

En realidad, de la característica par velocidad se pueden distinguir dos curvas, dependiendo del modo de funcionamiento del motor.

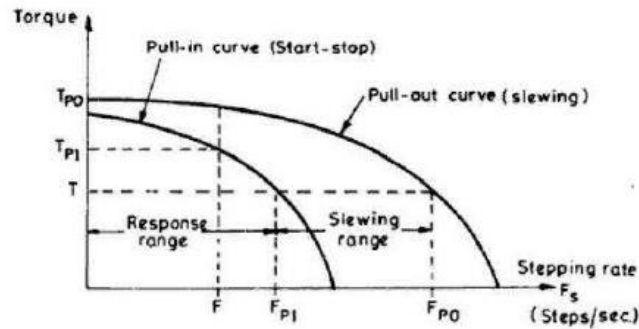


Figura 35. Curvas par velocidad del motor paso a paso. Fuente: [17]

La curva **pull-out** corresponde al **modo de funcionamiento de giro (slewing curve)**, en el cual el motor funciona en condiciones estables. En este modo, el rotor continúa moviéndose por el pulso anterior cuando recibe el siguiente pulso, por lo que gira de forma continua. Cuando el rotor recibe un pulso, este ya se encontraba en movimiento y no tiene que partir del reposo y vencer la inercia del mismo.

La curva **pull-in** corresponde al **modo de funcionamiento a pasos (start-stop curve)**. En este modo, el rotor gira desde el reposo por cada paso que se da. Al desenergizar la fase correspondiente, el rotor no girará hasta que no reciba el siguiente pulso, por lo que el motor arrancará desde el reposo y parará por cada paso que dé.

Las curvas de par-velocidad no tienen en cuenta la **inercia de la carga**. Existe una solución a este problema. Dado que la distancia vertical entre las dos curvas corresponde al par de inercia del motor, esa distancia llevarla por debajo de la curva pull-in dará una nueva curva de par, combinando la inercia del motor y de la carga, suponiéndolas iguales o proporcionales como se muestra en la figura 36.

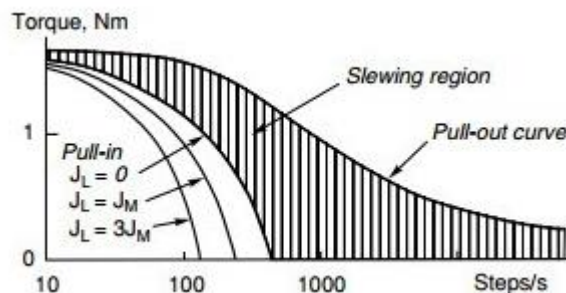


Figura 36. Curva típica par velocidad, con el efecto de la inercia de una carga. Fuente: [18]

Otros parámetros importantes son:

Par de retención (detent torque): Es el par máximo sin provocar el movimiento del eje cuando el motor se encuentra sin energizar. Aparece solamente en los motores paso a paso de imanes permanentes.

Par de anclaje (holding torque): Es el par máximo el cual el motor se encuentra en reposo cuando está energizado. Si el motor gira ciertos grados, el rotor mantiene la posición deseada.

Respuesta del motor paso a paso

Resulta útil estudiar la respuesta de un motor paso a paso ante un solo pulso. Idealmente, cuando se aplica un pulso, el rotor debería girar instantáneamente un ángulo de paso y detenerse en la posición de retención (posición de equilibrio).

Sin embargo, la respuesta real está lejos de este comportamiento ideal. El rotor oscilará durante un tiempo alrededor de la posición de retención antes de estabilizarse.

En la figura 37, suponiendo la excitación de una fase, en el punto C se aplica un pulso y se activa la fase del estator correspondiente. Esto genera un par que hace que el rotor gire hacia la posición de retención correspondiente D.

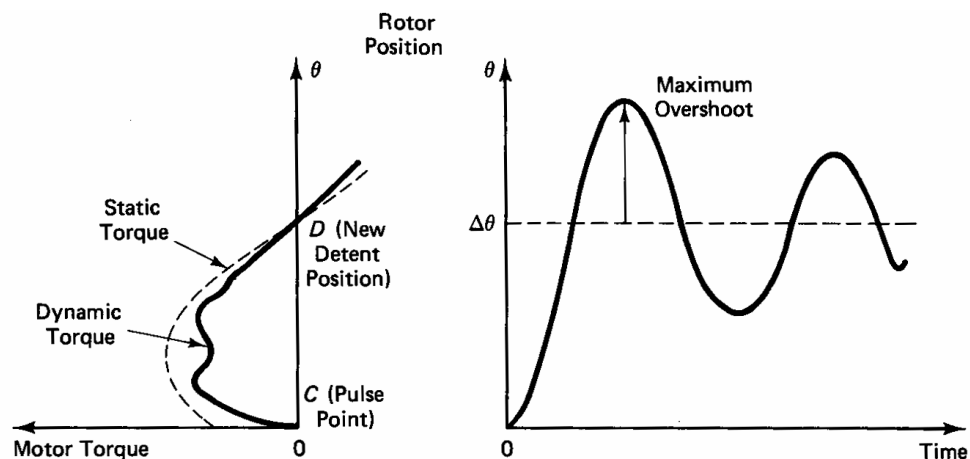


Figura 37. Respuesta de un motor paso a paso ante un pulso. Fuente: [11]

Si los pulsos están suficientemente espaciados, mayor que el tiempo de estabilización del motor, el rotor se detendrá al final de cada paso antes de comenzar el siguiente.

La respuesta del motor ante varios pulsos es equivalente a una secuencia en cascada de respuestas de un único pulso.

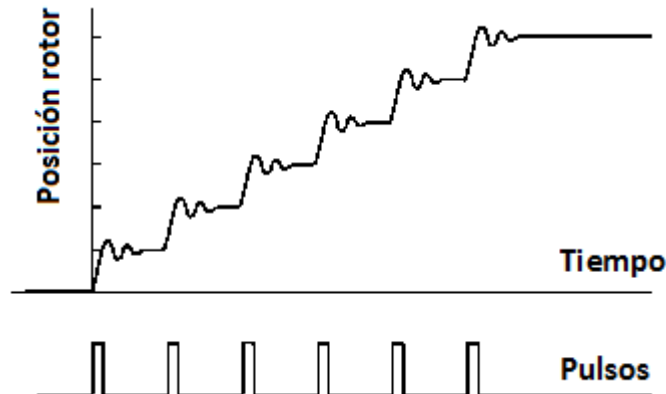


Figura 38. Respuesta del motor paso a paso ante varios pulsos. Fuente: [24]

Curva de par estático

Representa el **par** aplicado al rotor desde la fase energizada, en función de la **posición del rotor** θ , en condiciones ideales cuando se desprecian los efectos dinámicos.

En condiciones de funcionamiento normales, habrá tensiones inducidas debido a la autoinducción de las bobinas y puede haber corrientes parásitas en el rotor. Por lo tanto, se necesita un tiempo finito para que la corriente se acumule en los devanados una vez que se activan las fases.

Estos efectos hacen que el campo magnético se desvíe de las condiciones esperadas.

La **curva de par dinámico** es diferente de la curva de par estático. El problema es que es **difícil de predecir** debido a la dependencia de muchos factores que varían en el tiempo (velocidad del rotor, posición del rotor, corriente por las fases).

La curva de par estático es una aproximación adecuada para explicar las características de un motor paso a paso.

Para un análisis simplificado, el par estático se considera senoidal. En el caso considerado, con la fase 1 excitada y con las fases restantes inactivas, el par estático se puede expresar como: $T_1 = -T_{\max} \cdot \text{sen}(n_r \cdot \theta)$

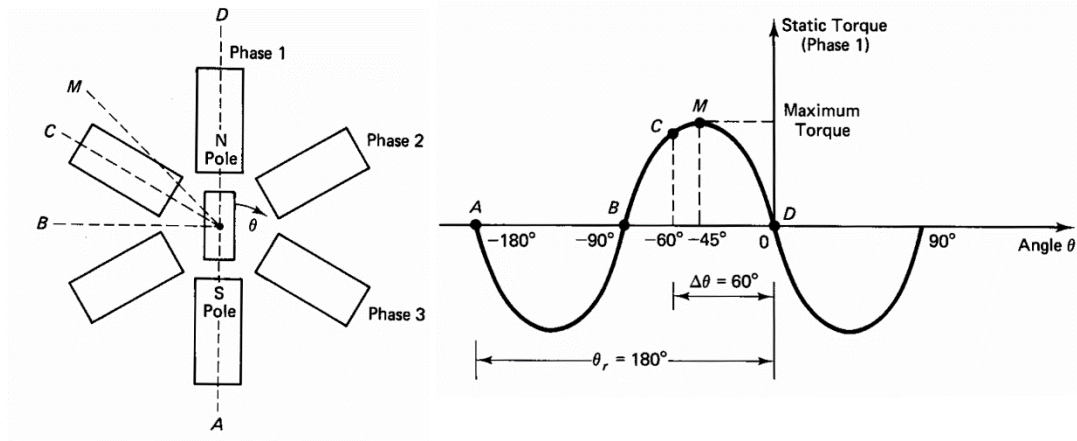


Figura 39. Distribución del par estático en un motor de reluctancia variable con una fase energizada.
Fuente: [11]

Siendo:

θ = posición angular en radianes, medida desde la posición de retención actual (con la fase 1 excitada).

n_r = número de dientes del rotor.

T_{max} = par estático máximo.

Analizando los **puntos importantes** del caso considerado:

Punto D: posición de retención actual. Posición de equilibrio.

Punto C: posición de retención anterior.

Punto B: posición de equilibrio inestable.

Punto A: posición de equilibrio estable.

Punto M: posición de par máximo.

El par estático en el punto de conmutación no es máximo.

En ocasiones se necesitan **respuestas rápidas** y **velocidades** del motor lo más **continuas** posibles.

Para lograrlo habría que **disminuir el tiempo de estabilización** del motor mediante una mayor disipación: **amortiguación mecánica y eléctrica**. Los inconvenientes de este tipo de disipación son la generación excesiva de calor, par de salida reducido y respuesta más lenta.

La **amortiguación electrónica** puede eliminar estos inconvenientes.

Durante operaciones transitorias en motores paso a paso, pueden ser necesarias secuencias de pasos no uniformes, dependiendo de la complejidad y la precisión requerida.

En la figura 40 se observa, por ejemplo, que antes de llegar a la primera posición de retención, se le ha enviado el segundo pulso al motor.

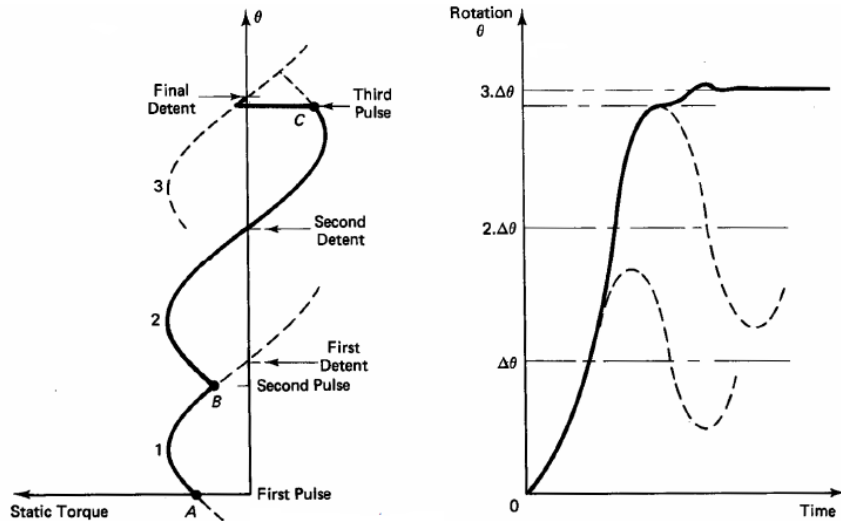


Figura 40. Respuesta del par para una secuencia de tres pasos, de secuencia de pasos no uniforme. Fuente: [11]

Constante de tiempo eléctrica τ_e

En un circuito RL la constante de tiempo es: $\tau_e = \frac{L}{R}$.

L es la inductancia de la fase.

R es la resistencia del circuito.

Como resultado de la autoinducción, la corriente en la fase energizada no se acumula instantáneamente. Cuanto mayor es la constante de tiempo eléctrica, más lenta es la acumulación de corriente.

A medida que aumenta la velocidad de paso, en el devanado se genera una fuerza contraelectromotriz que evita que la corriente alcance su valor estable, por lo que **el par disminuye con la velocidad de paso**. Cuando se estabiliza la corriente, ya se ha dado el siguiente pulso y no ha circulado la corriente esperada durante ese tiempo.

Esto dará como resultado un par más bajo al comienzo de cada paso. Además, debido a la autoinducción, la corriente no desaparece instantáneamente cuando la fase se apaga.

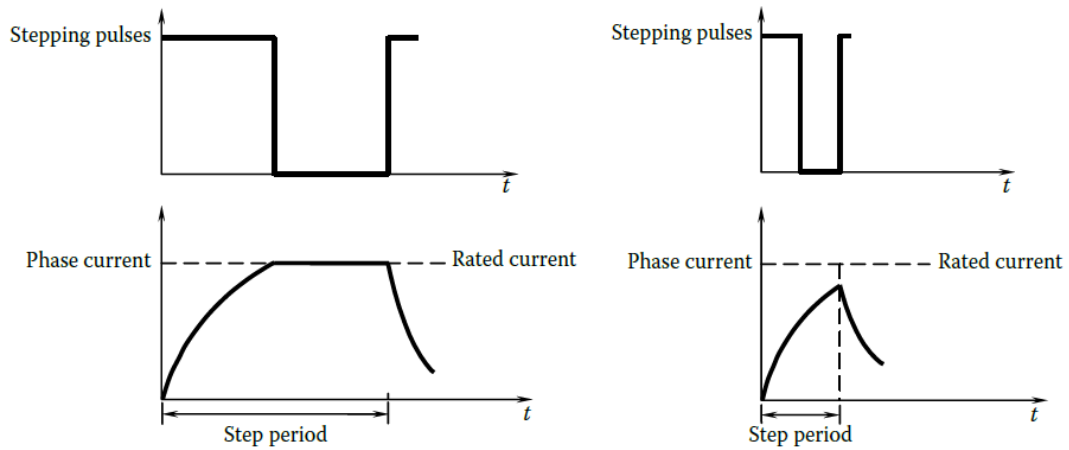


Figura 41. Efecto de la duración de los pulsos en la corriente por el devanado. Fuente: [10]

Las tensiones instantáneas causadas por la autoinducción pueden ser elevadas y dañar los circuitos. Disminuyendo la constante de tiempo eléctrica se pueden reducir estos inconvenientes, por ejemplo, aumentando la resistencia durante los períodos transitorios mediante una rama paralelo como la de la figura 42.

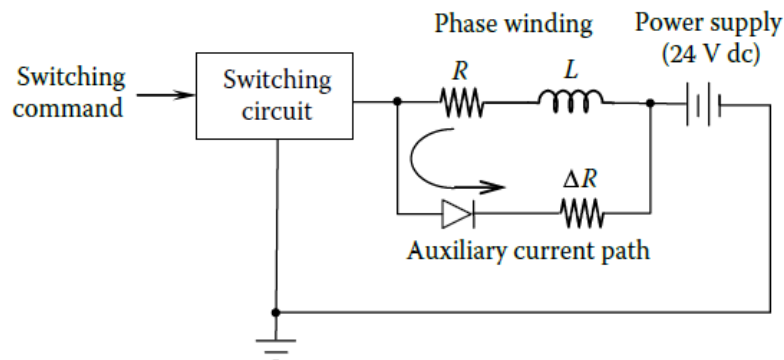


Figura 42. Circuito para disminuir la constante de tiempo eléctrica. Fuente: [10]

En condiciones estables no fluye corriente a través de ΔR .

Durante el encendido y apagado la corriente fluye a través de R y ΔR .

La resistencia total aumenta durante los tiempos de encendido y apagado y es más pequeña en los períodos estables para dar una corriente mayor, produciendo un mayor par.

$$I = \frac{V}{R} e^{1 - \frac{t}{\tau_e}}, \quad \text{siendo } \tau_e = \frac{L}{R + \Delta R}$$

2.1.2.5. Error de posición estática

Si un motor paso a paso no soporta una carga estática, la posición de equilibrio correspondería al punto de par nulo de la fase energizada.

Si hay una carga estática T_L , la posición de equilibrio se desplazaría un ángulo $-\theta_e$. El **ángulo de desplazamiento** $-\theta_e$ se llama **error de posición estática**.

Suponiendo que la curva de par estático es sinusoidal, se puede obtener una expresión para θ_e :

$$T = -T_{\max} \cdot \text{sen}(n_r \cdot \theta) = -T_{\max} \cdot \text{sen}\left(\frac{2\pi}{\theta_r} \cdot \theta\right) = -T_{\max} \cdot \text{sen}\left(\frac{2\pi}{p \cdot \Delta\theta} \cdot \theta\right)$$

n_r = número de dientes del rotor.

θ_r = ángulo de inclinación del diente del rotor.

p = número de fases en el estator.

$\Delta\theta$ = ángulo de paso.

T_{\max} = par máximo.

En condiciones de conmutación estándar se cumple que $-\Delta\theta \leq \theta \leq 0$.

$$T_L = -T_{\max} \cdot \text{sen}\left(\frac{2\pi \cdot (-\theta_e)}{p \cdot \Delta\theta}\right)$$

$$\theta_e = \frac{p \cdot \Delta\theta}{2\pi} \cdot \text{sen}^{-1}\left(\frac{T_L}{T_{\max}}\right) = \frac{p}{n} \cdot \text{sen}^{-1}\left(\frac{T_L}{T_{\max}}\right)$$

n = número de pasos por vuelta

De la expresión se deduce que el error de posición estática disminuye con el número de pasos por vuelta.

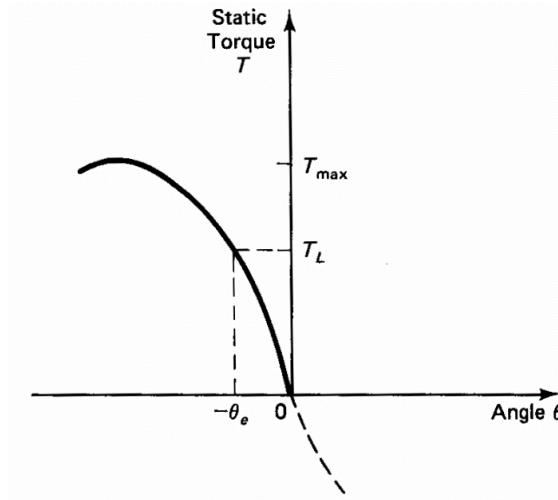


Figura 43. Error de posición estática. Fuente: [11]

2.1.2.6. Resonancia de motores paso a paso

A bajas velocidades, el motor se detiene en la posición de equilibrio después de cada cambio de excitación.

La respuesta del sistema a cada cambio de excitación ante un solo paso es bastante oscilatoria, como ya se mencionó anteriormente.

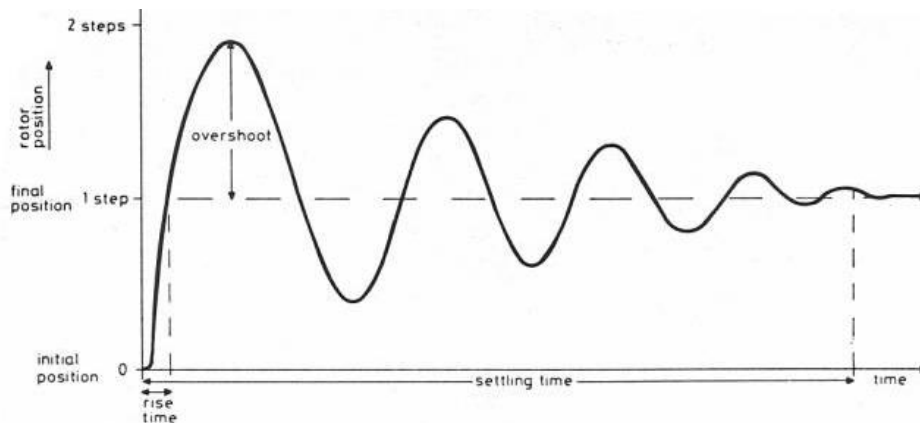


Figura 44. Respuesta típica de un motor paso a paso ante un solo pulso. Fuente: [11]

En aplicaciones que requieren un posicionamiento preciso, esta respuesta poco amortiguada puede ser un problema.

La frecuencia de oscilación para cualquier motor a partir de la característica par-posición del rotor se puede predecir. La ecuación de movimiento del sistema no amortiguado es:

$$J \cdot \frac{d^2\theta}{dt^2} + K_m \cdot \theta = 0 \quad y \quad f_n = \frac{1}{2\pi} \cdot \sqrt{\frac{K_m}{J}}$$

Siendo K_m rigidez magnética, obtenida a partir de la gráfica par estático/posición del rotor (figura 45).

J es el momento de inercia total del sistema.

f_n es la frecuencia natural de oscilación del rotor.

Las oscilaciones están ligeramente amortiguadas y el rotor finalmente se estabiliza en la posición de equilibrio.

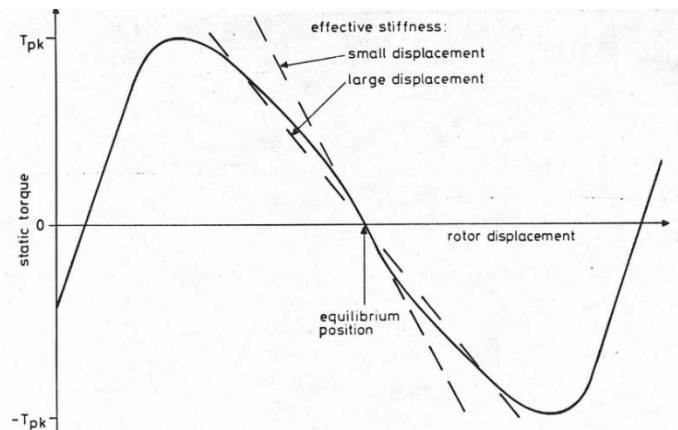


Figura 45. Obtención de K_m . Fuente: [11]

Una consecuencia de las oscilaciones en la respuesta ante un paso es la existencia de efectos de **resonancia** a frecuencias de pasos hasta la **frecuencia natural** de la oscilación del rotor.

Para el siguiente ejemplo se consideran dos casos:

- Frecuencia de paso = 60% de la frecuencia natural.
- Frecuencia de paso = frecuencia natural.

El comportamiento resonante del sistema produce una pérdida de par a velocidades que pueden predecirse si se conoce la frecuencia natural de la oscilación del rotor.

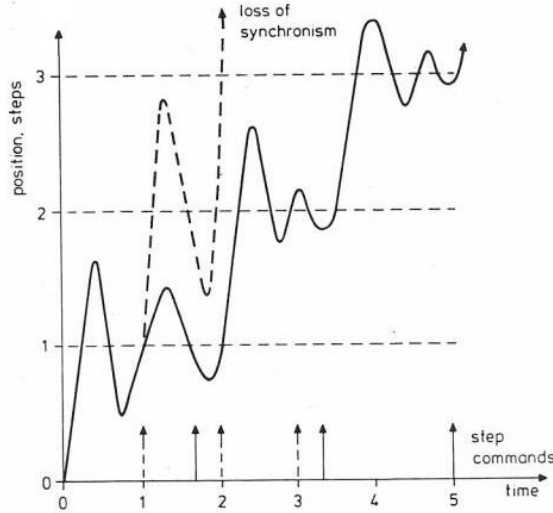


Figura 46. Respuestas ante frecuencias de paso proporcionales a la frecuencia natural. Fuente: [11]

Frecuencia de paso = frecuencia natural de oscilación

En la posición de equilibrio, el rotor tiene una velocidad positiva al final del primer paso. Como consecuencia de esta velocidad, al inicio del segundo paso, la respuesta será muy oscilatoria y podrá producir una pérdida del sincronismo con las señales de pulso enviadas al motor, perdiendo la posición de equilibrio (por cada paso, aumenta la amplitud de las oscilaciones, superando el par del motor y pudiendo provocar que el rotor se estabilice en una posición de paso diferente).

Frecuencia de paso = 0.6 · frecuencia natural de oscilación

El rotor gira a baja velocidad y no ha llegado a la posición de equilibrio cuando va a empezar el siguiente cambio de excitación. Al tener una velocidad más pequeña, el rotor tendrá una respuesta más uniforme para cada señal que se le envía, como se puede observar en la figura 46.

El rotor tiene que pasar a través de estas regiones después de tiempos que son múltiplos del período de oscilación del rotor ($1 / f_n$) y, por lo tanto:

$$\frac{f_n}{k} = \frac{1}{2\pi k} \sqrt{\frac{K_m}{J}}, k = 1,2,3, \dots \text{ submúltiplos de la frecuencia natural.}$$

Para un motor con una frecuencia natural de 100 Hz, tenga caídas en la característica par / velocidad a 100, 50, 33, 25, 20, ... pasos por segundo.

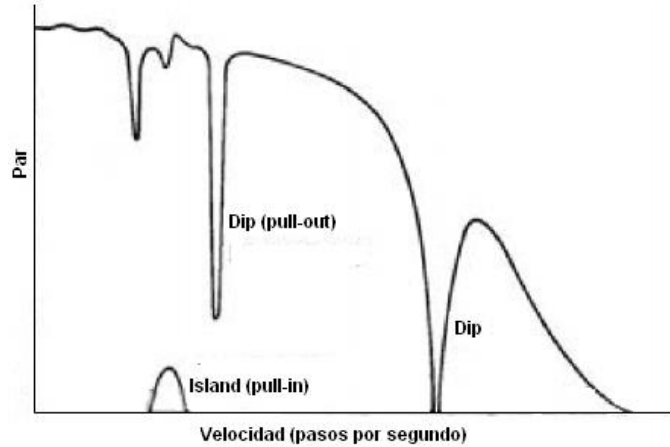


Figura 47. Caídas en la característica par velocidad. Fuente: [29]

Las oscilaciones de un motor paso a paso pueden reducirse mediante la introducción de una **mayor amortiguación** y, por lo tanto, **limitando la amplitud de la oscilación** en la respuesta ante un paso. Existen técnicas para mejorar la amortiguación, que se comentan a continuación.

2.1.2.7. Amortiguación de motores paso a paso

Las oscilaciones poco amortiguadas en los motores paso a paso no son deseables, particularmente en aplicaciones que requieren una trayectoria precisa en condiciones transitorias.

La **amortiguación** tiene las **ventajas** de suprimir los sobreimpulsos y disminuir las oscilaciones (disminuye el tiempo de estabilización).

Sin embargo, una amortiguación elevada tiene **inconvenientes**: respuesta lenta (mayor tiempo de subida, tiempo pico o retraso) y reducción del par. Aun así, las ventajas de la amortiguación superan las desventajas de manera general.

Se emplean **varias técnicas**, que utilizan la disipación de energía mecánica y eléctrica, para amortiguar la respuesta en motores paso a paso.

- La **amortiguación mecánica** generalmente es proporcionada por un amortiguador unido al eje del motor.

· Los métodos de **amortiguación eléctrica** incluyen la disipación de corriente parásita en el rotor, el uso de histéresis magnética y efectos de saturación, y una mayor disipación resistiva al añadir devanados adicionales al estator del motor.

Estas técnicas tienen sus **inconvenientes**, por ejemplo, generación excesiva de calor, reducción del par de salida del motor y disminución de la velocidad de respuesta.

Existen métodos de **amortiguación electrónica** para eliminar estos inconvenientes. Se basan en el empleo de esquemas de conmutación diseñados adecuadamente para la activación de fase y conseguir eliminar los sobrepulsos en el último paso de la respuesta.

El **inconveniente** general de la amortiguación electrónica es que las **secuencias de conmutación** son **complejas** y dependen de la naturaleza de una trayectoria de movimiento particular.

Para utilizar los métodos de amortiguación electrónica de manera efectiva, se requiere una gran cantidad de conocimiento sobre la respuesta real del motor.

A continuación, se explica más en detalle en qué consisten los tipos de amortiguación mencionados.

Amortiguación Mecánica

Consiste en colocar un elemento al eje del motor a través de un medio de disipación de energía, por ejemplo, un fluido viscoso o una superficie de fricción sólida.

Dos tipos comunes de amortiguadores de torsión son el amortiguador de Houdaille (amortiguador de torsión viscoso) y el amortiguador de Lanchester (amortiguador de fricción de Coulomb).

Se va a evaluar la efectividad de los amortiguadores mecánicos en motores paso a paso utilizando un modelo dinámico lineal para las oscilaciones de un solo paso.

Aproximación del par lineal (el par actúa como un muelle electromagnético):

$$T = -Km\theta.$$

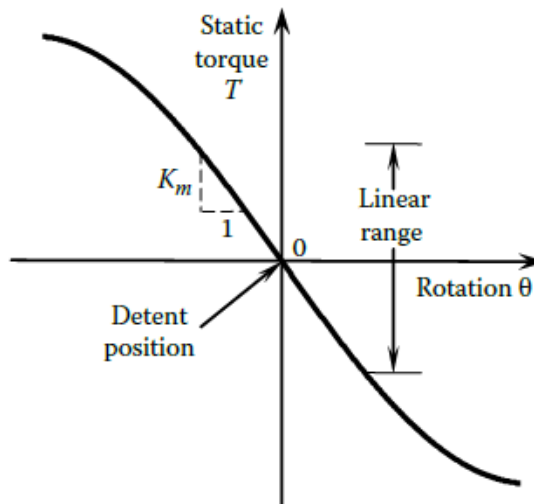


Figura 48. Aproximación de par lineal. Fuente: [10]

La siguiente figura muestra el modelo de un motor paso a paso para estudiar las oscilaciones ante un paso:

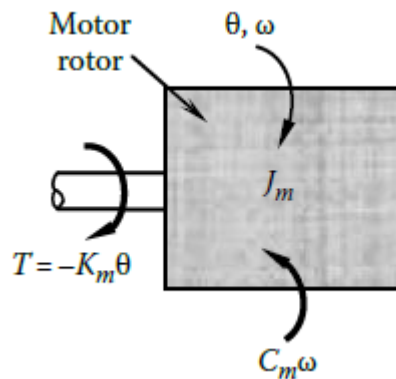


Figura 49. Modelo para oscilaciones de un motor paso a paso de un solo paso. Fuente: [10]

La ecuación de movimiento que describe el sistema anterior es:

$$J_m \cdot \frac{dw}{dt} = -C_m \cdot w - K_m \cdot \theta \rightarrow J_m \cdot \frac{d^2\theta}{dt^2} + C_m \cdot \frac{d\theta}{dt} + K_m \cdot \theta = 0$$

Siendo

K_m = constante de par del motor.

C_m = constante de amortiguación debido a los mecanismos de disipación interna (incluyendo fricción de los cojinetes, disipación resistiva en devanados, disipación de corriente parásita en el rotor e histéresis magnética).

Frecuencia natural no amortiguada w_n :

$$w_n = \sqrt{\frac{K_m}{J}} , \quad w_n = 2\pi \cdot f_n$$

Razón de amortiguamiento ζ :

$$\zeta = \frac{C_m}{2\sqrt{K_m \cdot J_m}}$$

Con un amortiguador Houdaille conectado al motor, suponiendo que el amortiguador está rígidamente unido al eje, las ecuaciones de movimiento son:

$$(J_m + J_h) \cdot \frac{d^2\theta}{dt^2} = -C_m \cdot \frac{d\theta}{dt} - K_m \cdot \theta - C_d \left(\frac{d\theta}{dt} - \frac{d\theta_d}{dt} \right)$$

$$J_d \cdot \frac{d^2\theta_d}{dt^2} = C_d \left(\frac{d\theta}{dt} - \frac{d\theta_d}{dt} \right)$$

Siendo: J_d momento de inercia del amortiguador

J_h momento de inercia de la carcasa.

C_d constante de viscosidad del amortiguador.

θ_d ángulo de rotación de la inercia del amortiguador.

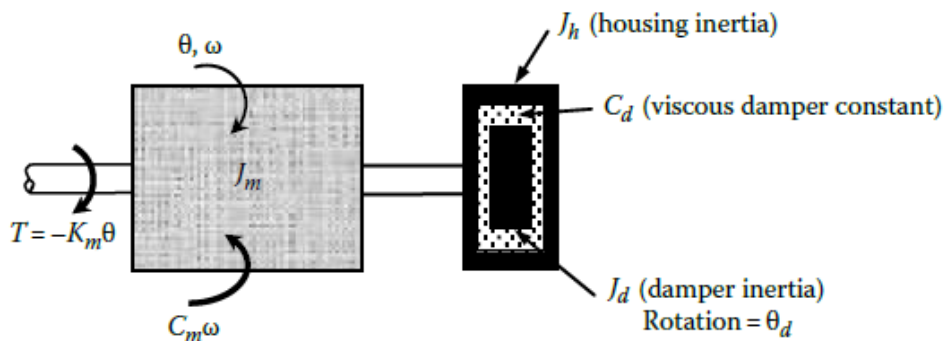


Figura 50. Motor paso a paso con un amortiguador Houdaille. Fuente: [10]

En la siguiente figura se puede observar la amortiguación de la respuesta del rotor ante un pulso. Se alcanza la posición de equilibrio sin muchas oscilaciones con el amortiguador mecánico.

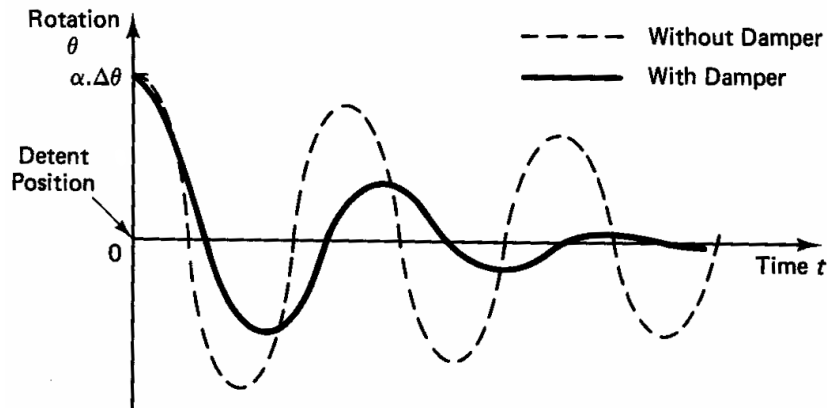


Figura 51. Respuesta típica de un solo paso de un motor paso a paso con un amortiguador Houdaille.
Fuente: [11]

Este método tiene una serie de **inconvenientes**:

Aumenta inercia al motor, lo que reduce la frecuencia natural del motor y, por lo tanto, disminuye la velocidad de respuesta.

Reducción del par y aumento de la generación de calor, lo que puede requerir un método de enfriamiento especial.

Amortiguación electrónica

La amortiguación de la respuesta de un motor paso a paso mediante el control de **conmutación electrónica** es un método más interesante de supresión de sobreimpulso por varias **razones**:

- No es un método de disipación de energía.
- Si se realiza la secuencia de conmutación correctamente, se podría obtener prácticamente una respuesta de sobreimpulso nulo.

La reducción en el par a la salida es mucho menor en comparación con las pérdidas de par en los métodos de amortiguación mecánica y eléctrica.

La mayoría de las técnicas de amortiguación electrónica siguen un **procedimiento** de dos pasos:

1. Desacelerar la respuesta del último paso del motor para evitar grandes sobreimpulsos.

2. Energizar la fase final cuando la respuesta del motor esté muy **cerca de la posición de retención final** (cuando el par sea muy pequeño).

La mayoría de los esquemas se diferencian en la forma en que se produce la desaceleración de la respuesta (paso 1).

Los **tres métodos** comunes de desaceleración de la respuesta son los siguientes:

Método de apagado de pulso: consiste en desenergizar todas las fases activas durante un período corto tiempo. Durante ese tiempo el par estático es nulo y el motor desacelerará. Se vuelve a energizar en el punto C, donde el par es positivo y de bajo valor y el rotor llegará a la posición de retención final deseada.

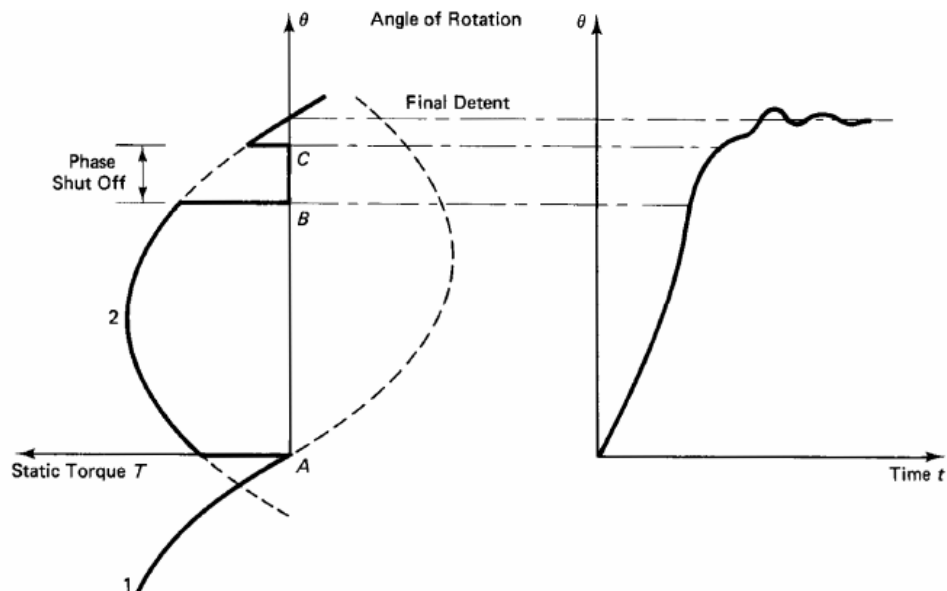


Figura 52. Método de apagado de pulso. Fuente: [11]

Método de inversión de pulso: consiste en aplicar un pulso en la dirección opuesta durante un período corto tiempo. Se energiza la fase 3 en el punto B, a menos de la mitad del ángulo de paso. El par estático es negativo para la primera mitad del ángulo de paso y positivo para la segunda, por lo que el motor desacelerará y después acelerará, como se observa en la figura. En el punto C el par estático de la fase 2 y 3 son iguales. Para evitar grandes sobreimpulsos se desenergiza la fase 3 y se energiza la fase 2. Finalmente, el rotor llegará a la posición de equilibrio.

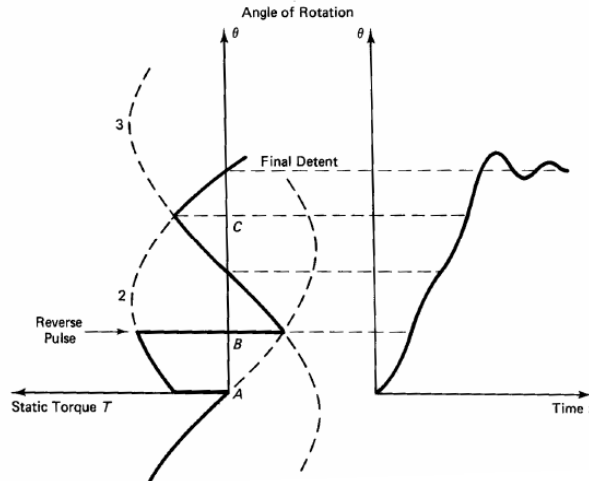


Figura 53. Método de inversión de pulso. Fuente: [11]

Método de retraso de pulso: consiste en mantener la fase actual más allá de su posición de retención durante un período corto tiempo. Para alcanzar la posición de equilibrio A se ha energizado la fase 1. Al llegar al punto A no se deja de desenergizar, el par negativo hará que el motor desacelere y al cabo de un tiempo, en el punto B, se energiza la fase 2 en el que el par estático es bajo y el rotor llegará a la posición de retención final.

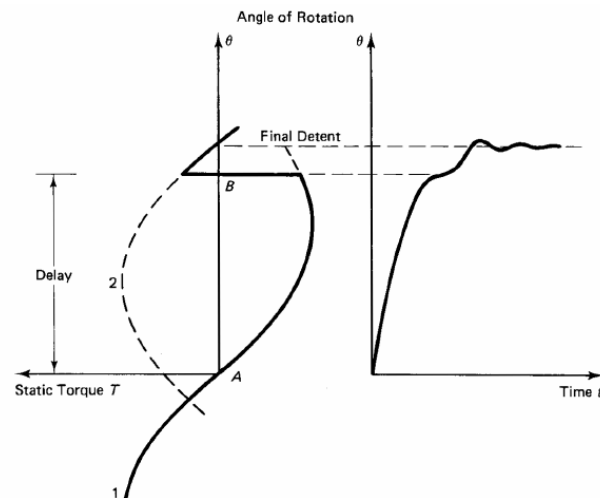


Figura 54. Método de retraso de pulso. Fuente: [11]

En estas técnicas de amortiguación electrónica, la respuesta real depende de **muchos factores**, como el comportamiento dinámico de la carga. Por lo tanto, escoger esos puntos de conmutación no pueden especificarse de manera exacta a menos que se conozca la respuesta real mediante simulaciones.

Para una conmutación precisa se requiere medir la respuesta real del motor y usar esa información en tiempo real para aplicar los pulsos necesarios.

2.1.2.8. Modelos de motores paso a paso

Para el funcionamiento estable de un motor paso a paso a bajas velocidades, generalmente no se necesita diferenciar entre motores de reluctancia variable y de imanes permanentes (el motor híbrido es un tipo especial de imanes permanentes).

Pero en condiciones transitorias, las características de par de ambos tipos de motores son bastante diferentes.

El par en un motor de imanes permanentes varía linealmente con la corriente de fase.

El par en un motor de reluctancia variable varía cuadráticamente con la corriente de fase.

Para un motor paso a paso en condiciones estables, el par se puede aproximar por una función sinusoidal: $T = -T_{\max} \cdot \text{sen}(n_r \theta)$.

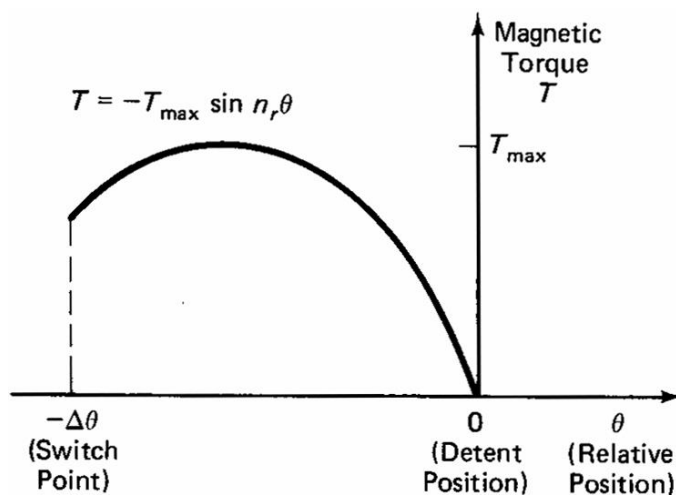


Figura 55. Par estático sinusoidal. Fuente: [11]

La ecuación mecánica del movimiento del motor viene dada por:

$$T - T_L - T_b(\theta, \dot{\theta}) = J \ddot{\theta}$$

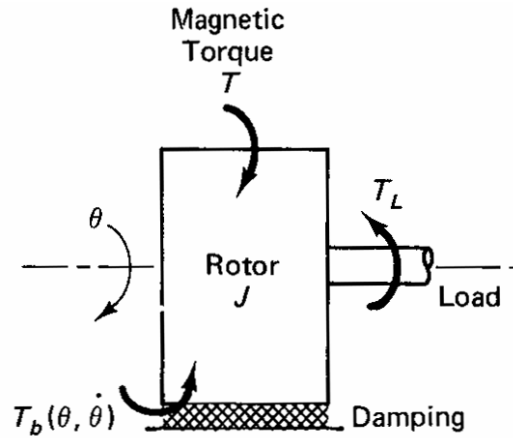


Figura 56. Modelo de un motor paso a paso. Fuente: [11]

En condiciones transitorias y de alta velocidad, muchas de las **magnitudes** que se suponían constantes **variarán con el tiempo y con la posición del rotor**.

Para una tensión dada en un devanado, la corriente de fase no será constante.

La inductancia L del circuito variará en cada fase con la posición del rotor.

Además, se inducirá una f.c.e.m. debido a la variación de flujos magnéticos resultantes de la velocidad de rotación del rotor.

Se necesitaría un **modelo dinámico** para representar el comportamiento de un motor paso a paso en condiciones transitorias y de alta velocidad.

A continuación, se muestra un circuito equivalente aproximado para una fase de un motor paso a paso.

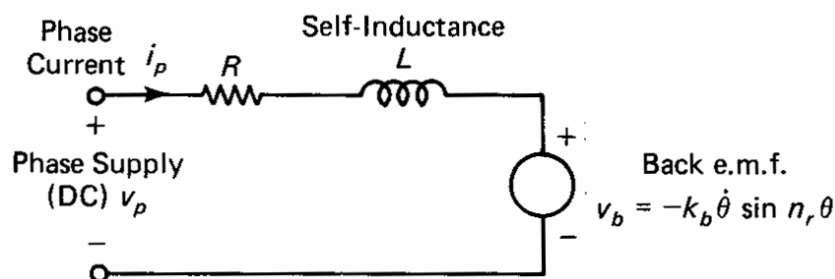


Figura 57. Circuito equivalente aproximado por fase de un motor paso a paso. Fuente: [11]

$$v_p = R \cdot i_p + L \cdot \frac{di_p}{dt} + v_b$$

Como θ es negativo en un paso convencional (de $-\Delta\theta$ a 0), se observa que v_b es positivo para una velocidad angular positiva.

La inductancia también varía con la posición del rotor θ y es periódica con el paso del diente: $L = L_0 + L_a \cdot \cos(n_r \cdot \theta)$

Este modelo es válido para ambos tipos de motores, pero la ecuación del par dependerá del tipo de motor paso a paso.

Ecuación del par para motores de imanes permanentes

En un motor paso a paso de imanes permanentes, el flujo magnético es creado tanto por la corriente de fase i_p como por el rotor magnetizado. El flujo magnético del rotor es constante, pero su enlace con los devanados de fase dependerá de la posición del rotor θ .

$$T = -k_m \cdot i_p \cdot \text{sen}(n_r \cdot \theta)$$

i_p es la corriente de fase y k_m es la constante de par para el motor de imanes permanentes.

Ecuación del par para motores de reluctancia variable

En un motor paso a paso de reluctancia variable el rotor no está magnetizado, por lo tanto, no crea flujo magnético. El flujo generado por la corriente de fase i_p está enlazado con los devanados de la fase. Sin embargo, el enlace de flujo dependerá del movimiento del motor.

$$T = -k_r \cdot i_p^2 \cdot \text{sen}(n_r \cdot \theta)$$

k_r es la constante de par del motor de reluctancia variable.

Resumen

Para calcular el par en una posición dada del rotor, hay que resolver las siguientes ecuaciones diferenciales, para valores conocidos de la posición y velocidad del rotor y para una tensión de fase v_p .

$$v_p = R \cdot i_p + L \cdot \frac{di_p}{dt} + v_b$$

$$v_b = -k_b \cdot \dot{\theta} \cdot \text{sen}(n_r \cdot \theta)$$

$$L = L_o = L_a \cdot \cos(n_r \cdot \theta)$$

Se supone que los parámetros del modelo R , L_o , L_a y k_b son conocidos (experimentalmente o de datos que proporciona el fabricante).

El par se calcula utilizando:

$T = -k_r \cdot i_p^2 \cdot \sin(n_r \cdot \theta)$ para motores paso a paso de reluctancia variable.

$T = -k_m \cdot i_p \cdot \sin(n_r \cdot \theta)$ para motores paso a paso de imanes permanentes e híbridos.

Se supone que es conocida la constante de par (k_m o k_r).

La simulación del modelo se completa utilizando el par calculado en la ecuación mecánica de movimiento para determinar la posición y la velocidad del rotor:

$$T - T_L - T_b(\theta, \dot{\theta}) = J \ddot{\theta}$$

2.1.2.9 Elección de un motor paso a paso

Diferencias entre motores de imanes permanentes y de reluctancia variable

Motores de imanes permanentes (incluidos los híbridos)

Pasos pequeños (normalmente en torno a 0.9° y 1.8°), por lo tanto, mayor resolución.

Mayor capacidad de producir de par para un volumen de motor dado.

Cuando se desenergizan todos los devanados, el rotor mantiene la posición debido al imán permanente. Tiene un **par de retención**.

Motores de reluctancia variable

Los pasos son mayores (en torno a 15°) que en los de imanes permanentes, por lo que se requieren menos pasos para moverse una distancia determinada. Tienen menor precisión.

Menor inercia mecánica del rotor debido a la ausencia del imán permanente y, por lo tanto, es posible una aceleración más rápida.

Cuando se desenergizan los devanados del motor, al no tener un imán permanente, no hay un par de retención y el rotor perderá la posición.



Selección de un motor paso a paso

- 1.- Enumerar los requisitos principales para la **aplicación en particular**, incluidas las velocidades, las aceleraciones, la precisión y resolución requeridas, y las **características de la carga** como el tamaño, la inercia, las frecuencias naturales fundamentales y los pares resistentes.
- 2.- Con los datos proporcionados, obtener los requisitos de par y velocidad para la aplicación particular.
- 3.- Utilizar las curvas de par-velocidad de motores paso a paso comerciales y seleccionar el más adecuado. Para ello, utilizar los requisitos de par y velocidad obtenidos en el paso 2 y los requisitos de precisión y resolución en el paso 1.
- 4.- Si el motor que cumple con los requisitos no está disponible, se pueden modificar los requisitos de velocidad y par mediante dispositivos como sistemas de engranajes o amplificadores.

Ventajas y desventajas de los motores paso a paso

Proporcionan un par elevado a bajas velocidades, aunque comparado con un motor de corriente continua convencional el par es bastante menor.

El control de estos motores es sencillo. Funcionan perfectamente en lazo abierto, sin realimentación, pudiendo ahorrarse un sensor de posición del rotor y el programa a realizar será menos complejo.

Los motores paso a paso, al ser controlados por impulsos con niveles de tensión constantes, se adaptan fácilmente a aplicaciones de control digital. Se utilizan microcontroladores en los que el hardware de interconexión es mínimo.

Son más económicos que otros motores de corriente continua, aunque depende de la precisión requerida.

Tienen mejor precisión con ángulos de giro pequeños.

Rangos de velocidad variables, ya que depende proporcionalmente de la frecuencia de los pulsos de entrada.

Carecen de escobillas, por lo que el motor tendrá menos pérdidas que uno convencional y tendrá mayor vida útil.

Altos niveles de vibración debido al movimiento gradual de estos motores.

Pueden producirse grandes errores y oscilaciones cuando pierde un pulso cuando se utiliza el control en lazo abierto.

Los problemas térmicos pueden ser importantes cuando se opera a altas velocidades.

Aplicaciones

Las aplicaciones de estos motores se encuentran en periféricos de ordenador como impresoras, scanners o controladores de discos, cámaras, telescopios y sistemas de posicionamiento de satélites, en aplicaciones de electromedicina, movimiento de brazos robóticos, posicionamiento de válvulas en controles industriales y en control numérico.

Para movimientos cortos y repetitivos a velocidades inferiores a 2000 rpm los motores paso a paso pueden ser una buena opción.

A continuación, se muestran imágenes de alguna de las aplicaciones de estos motores.

En impresoras 3D.



Figura 58. Aplicaciones de motores paso a paso en impresoras 3D. Fuente: [19]

En telescopios.



Figura 59. Aplicaciones de motores paso a paso en telescopios. Fuente: [20]

Para una controladora de discos.

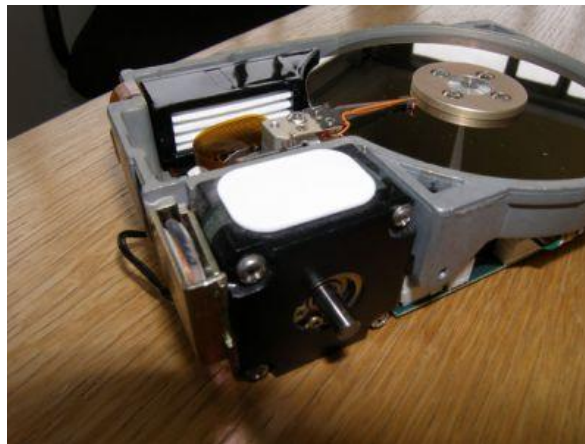


Figura 60. Aplicaciones de motores paso a paso en controladora de discos. Fuente: [23]

En una mesa de posicionamiento lineal motorizada en dos ejes, x-y, para una fresadora CNC.

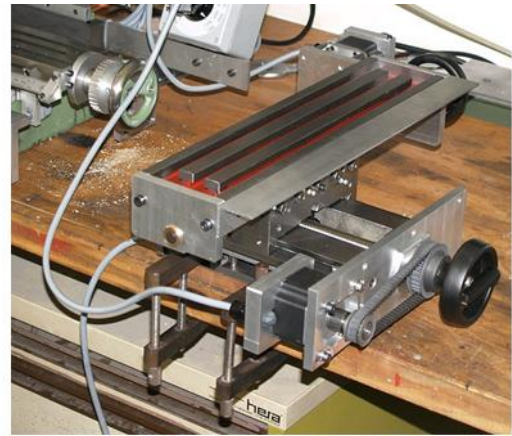
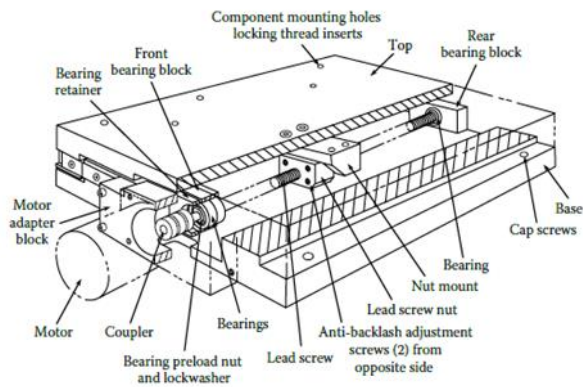


Figura 61. Aplicaciones de motores paso a paso en mesa de posicionamiento lineal para fresadora CNC. Fuentes: [10] y [21]

En una fábrica, para una aplicación específica.

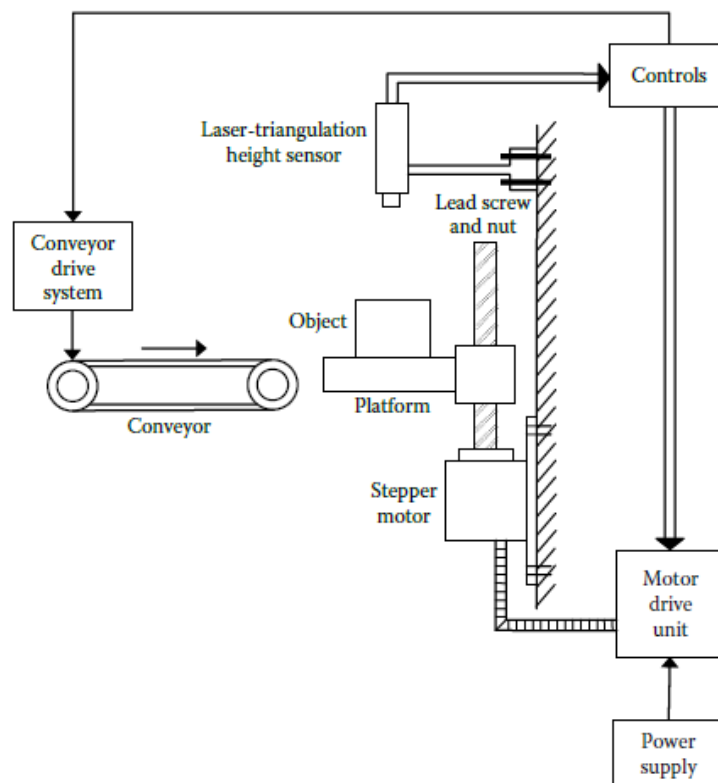


Figura 62. Aplicaciones de motores paso a paso en una fábrica. Fuente: [10]

En un brazo robótico, tres motores para cada eje x y z.



Figura 63. Aplicaciones de motores paso a paso en brazos robóticos. Fuente: [22]

3. DESARROLLO DEL TRABAJO

La información principal de este apartado se ha extraído de [30], [33] y [38] para la descripción general de LabVIEW, Raspberry Pi y el motor y driver utilizados, y de [47] y [50] para el desarrollo del software empleado para el control del motor paso a paso.

3.1. Introducción teórica

3.1.1. Raspberry Pi

Una Raspberry Pi es un ordenador de placa simple (Single Board Computer) de bajo coste y bajo consumo, creada con el objetivo de estimular la enseñanza de informática en las escuelas.

El diseño se centra en un microprocesador Broadcom con memoria RAM, GPU, puertos USB, HDMI, Ethernet, pines GPIO y conector para cámara.



Figura 64. Raspberry Pi. Fuente: [25]

Usa principalmente sistemas operativos GNU/Linux. Raspbian, una distribución derivada de Debian optimizada para el hardware de Raspberry Pi, es la distribución recomendada por la fundación para iniciarse.

También permite usar otros sistemas operativos, incluido una versión de Windows 10.

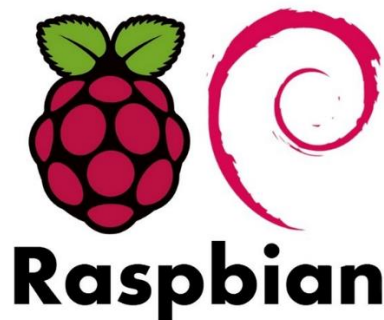


Figura 65. Raspbian, Raspberry Pi + Debian. Fuente: [26]

Esta arquitectura no se suele utilizar como ordenadores personales, sino que se usan en entornos industriales o en **sistemas embebidos**.

Un sistema embebido es un sistema de computación diseñado para realizar una o pocas funciones de procesamiento en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general, que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir **necesidades específicas**.

Se pueden programar directamente en el lenguaje ensamblador del microcontrolador o utilizando compiladores específicos como C++, Python, LabVIEW...

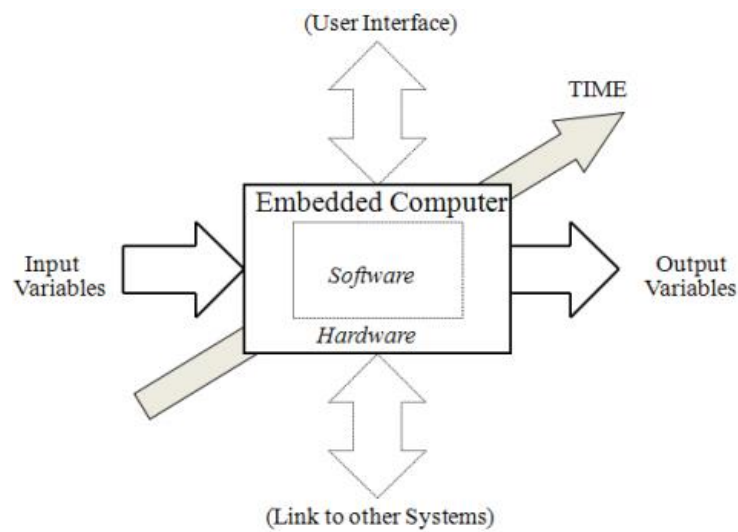


Figura 66. Sistema embebido. Fuente: [27]

Los principales **componentes** de un sistema embebido son:

- El **microcontrolador**, que aporta capacidad de cómputo al sistema, pudiendo incluir memoria interna o externa con arquitectura específica según los requisitos.
- La **comunicación** adquiere gran importancia en los sistemas embebidos. Lo normal es que el sistema pueda comunicarse mediante interfaces estándar inalámbricas o de cable.
- El subsistema de presentación tipo suele ser una **pantalla** gráfica, táctil, LCD...
- Se denominan **actuadores** a los elementos electrónicos que el sistema se encarga de controlar. Puede ser un motor eléctrico, un conmutador tipo relé, etc. El más habitual es generar una señal PWM para el control de velocidad de motores de corriente continua.

- El módulo de **E/S analógicas y digitales** suele emplearse para digitalizar señales analógicas procedentes de sensores, activar diodos ledes, reconocer el estado abierto cerrado de un conmutador o pulsador, etc.
- La **fuentes de energía** se encarga de generar las diferentes tensiones y corrientes necesarias para alimentar los diferentes circuitos.

3.1.2. Lenguaje de programación LabVIEW

LabVIEW es un **lenguaje de programación gráfico** creado por National Instruments que facilita tareas de ingeniería como la adquisición de datos, análisis de medidas, presentación de datos, creación sistemas de control embebidos y aplicaciones de gran flexibilidad, que además proporciona visualización y manejo de gráficas con datos dinámicos y procesamiento digital de señales en tiempo real.

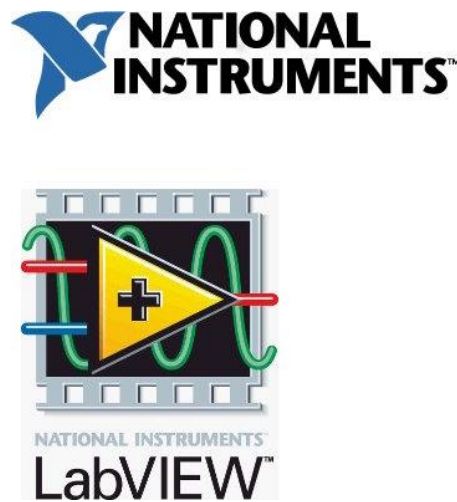


Figura 67. Logo de LabVIEW y National Instruments. Fuente: [28]

Sus principales **características** son:

- La facilidad de uso, tanto para programadores profesionales como para personas con pocos conocimientos en programación que pueden realizar programas relativamente complejos.
- Gran variedad de interfaces de comunicación (puerto serie y paralelo, Bluetooth, USB, OPC...).



- Capacidad de interactuar con otros lenguajes y aplicaciones (ActiveX, Multisim, Matlab/Simulink, AutoCAD, SolidWorks, etc).
- Reducción del tiempo de desarrollo de aplicaciones que permite la entrada a la informática a profesionales de cualquier otro campo.

Conceptos clave de LabVIEW

Dentro de LabVIEW hay varios elementos y conceptos que son clave para su utilización.

- **Entorno de LabVIEW:** El entorno de LabVIEW consiste en el administrador de LabVIEW VI (explorador de proyectos), las herramientas de programación, las características de depuración, las plantillas y los ejemplos de muestra preparados y una interfaz fácil para los controladores de hardware.
- **LabVIEW VI:** LabVIEW VI es un "instrumento virtual" que permite construir una interfaz de usuario y contiene el código de programación. Su origen provenía del control de instrumentos, aunque hoy en día se ha expandido, no sólo al control de todo tipo de electrónica, sino también a su programación embebida.
- **Programación G de LabVIEW:** Este es el lenguaje de programación gráfico donde los algoritmos funcionales se construyen usando bloques y cableando entradas y salidas.
- **Flujo de datos de LabVIEW:** Este es el concepto central que determina el orden de ejecución del programa.

Ventajas / desventajas de LabVIEW

Ventajas de LabVIEW

- La interfaz gráfica es fácil de usar.
- Proporciona una plataforma universal para aplicaciones en diversos campos.
- Puede usarse con hardware de terceros: puede interactuar con C / C ++, VB, Fortran, etc.
- Fácil de interactuar con muchos elementos de hardware, por ejemplo, para la adquisición de datos.



Desventajas de LabVIEW

- LabVIEW es de fuente única y algunas compañías pueden no querer usar un producto no estandarizado por la industria.
- La programación gráfica puede llevar un tiempo de familiarización si se está más acostumbrado a la programación de texto.

LabVIEW Community edition

Este 2020 salió el paquete LabVIEW Community edition para fines no comerciales y es el que se utilizará para implementar el control del motor paso a paso.

Incluye:

Acceso a LabVIEW 2020 Community Edition y LabVIEW NXG Community edition.

El toolkit LINX para usar la Raspberry Pi, BeagleBoard y Arduino con LabVIEW.

Acceso al Módulo de Web de LabVIEW NXG para crear aplicaciones basadas en web.

Uso del programa

LabVIEW es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener prediseñados una gran cantidad de bloques, se facilita al usuario la creación del proyecto, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final.

Cada VI contiene **dos partes** principales:

- El **panel frontal**: Cómo el usuario interacciona con el VI.
- El **diagrama de bloques**: El código del programa.

Panel Frontal

Esta pantalla se utiliza para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real (como fluyen los datos, un ejemplo sería un interruptor, el cual, si se cierra en el propio panel, se observa la salida que se enciende el led correspondiente).

En esta interfaz se definen los controles (las entradas que pueden ser botones, marcadores...) e indicadores (las salidas que pueden ser gráficas, leds...).

Cada control o indicador del panel frontal tiene un terminal correspondiente en el diagrama de bloques.

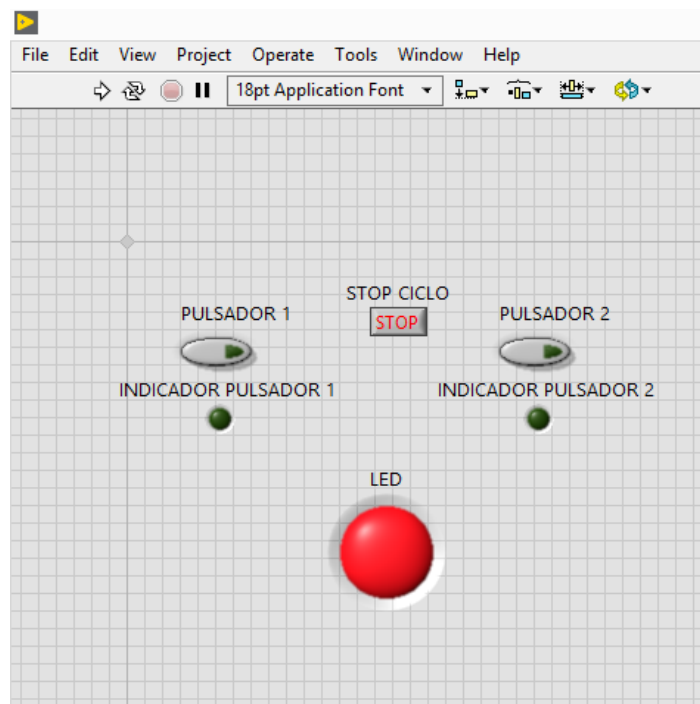


Figura 68. Panel Frontal. Fuente: Elaboración propia.

Diagrama de Bloques:

Es el programa propiamente dicho, donde se colocan los bloques que realizan una determinada función y se interconectan mediante cables, como se observa en el ejemplo.

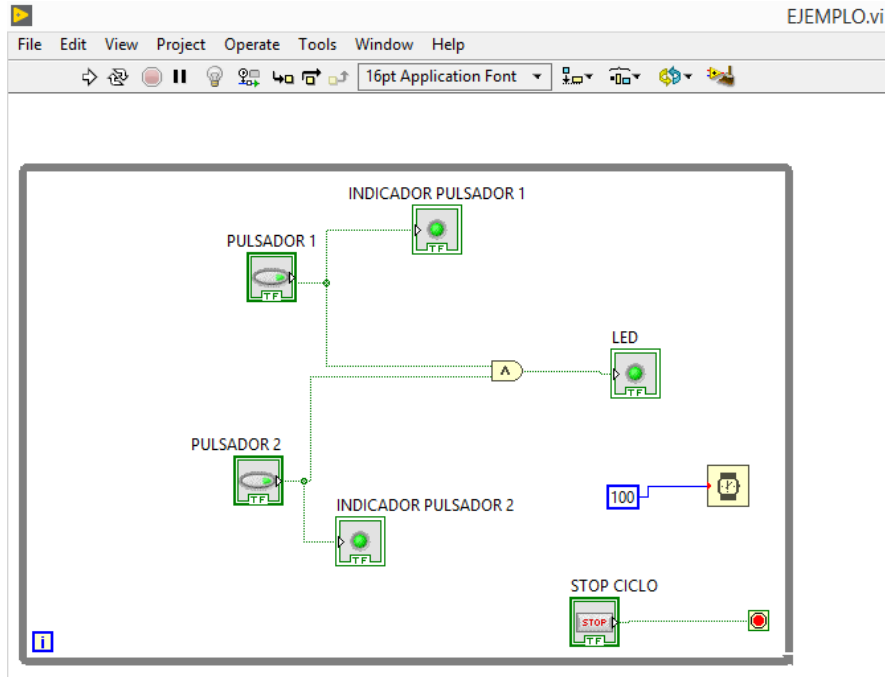


Figura 69. Diagrama de Bloques. Fuente: Elaboración propia.

En la ventana del panel frontal se encuentran las opciones para visualizar y controlar el sistema realizado.

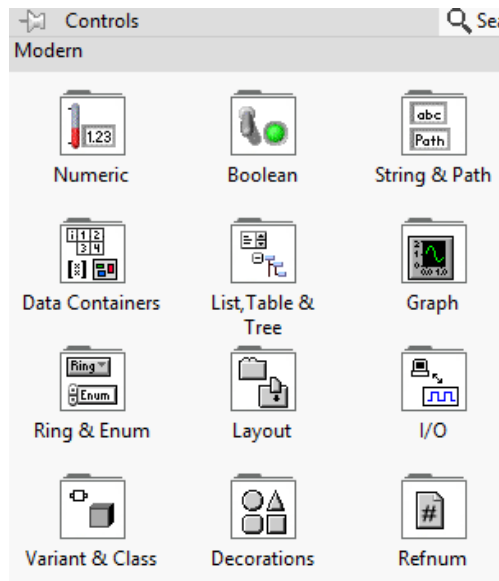


Figura 70. Paleta de controles. Fuente: Elaboración propia.

En la opción boolean se encuentran los siguientes indicadores, tales como pulsadores, interruptores y leds.

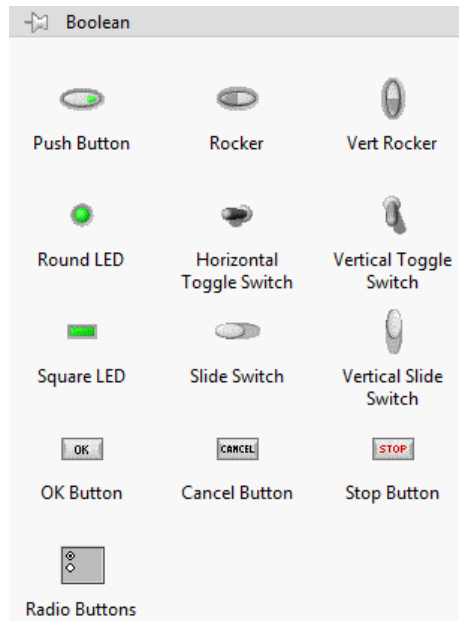


Figura 71. Ejemplo de entradas y salidas en el panel frontal. Fuente: Elaboración propia.

En la opción numeric se encuentran bloques para controlar, como por ejemplo el bloque 'gauge', que mide velocidad, 'numeric indicator' para conocer el valor en tiempo real de un parámetro...

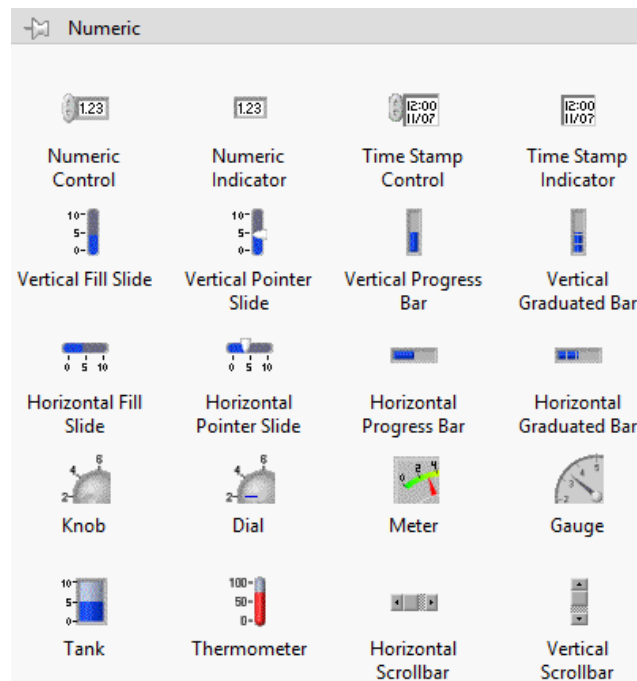


Figura 72. Ejemplo de visualización y control en el panel frontal. Fuente: Elaboración propia.

En la ventana del diagrama de bloques se encuentran las opciones para realizar el sistema con los terminales del panel frontal, mediante bucles, operaciones...

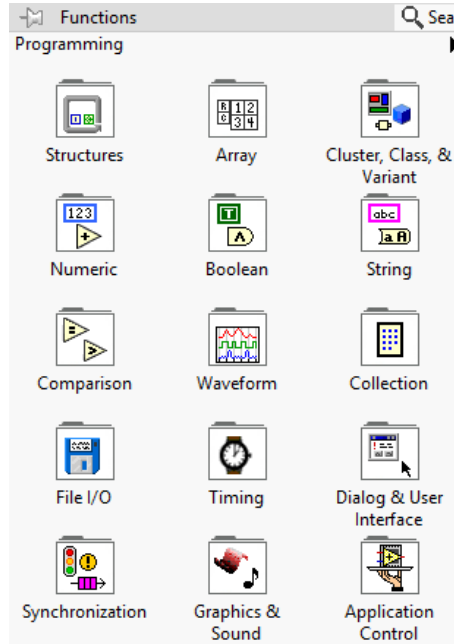


Figura 73. Paleta de funciones. Fuente: Elaboración propia.

En la opción structures se encuentran los bucles while, for, if...

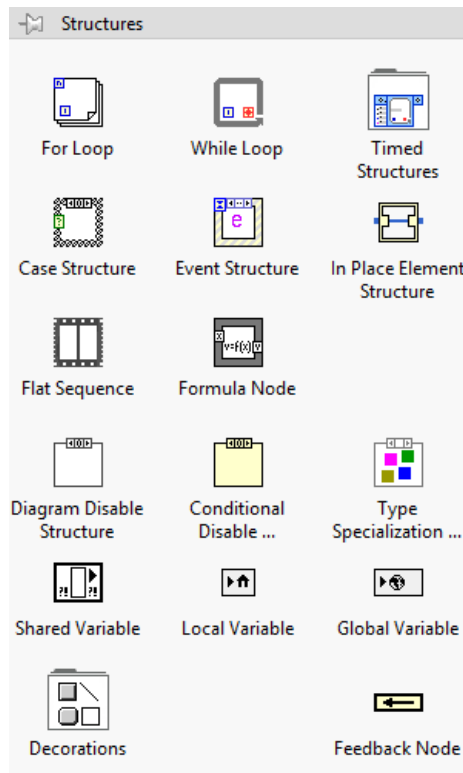


Figura 74. Ejemplo de bucles disponibles en el diagrama de bloques. Fuente: Elaboración propia.

En la opción boolean se encuentran las operaciones booleanas como And, Or, Not y constantes como True y False, para cambiar el estado de algún parámetro booleano.

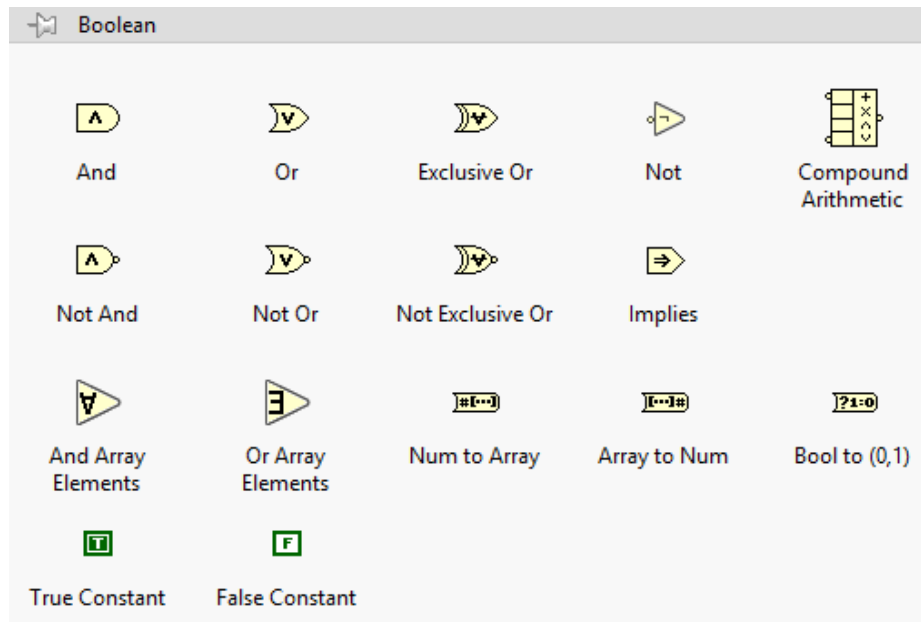


Figura 75. Ejemplo de operaciones booleanas. Fuente: Elaboración propia.

En ambas pantallas se encuentra la barra de herramientas de estado.

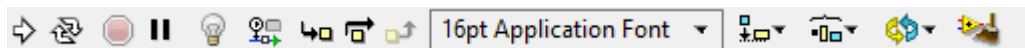


Figura 76. Barra de herramientas. Fuente: Elaboración propia.

Los elementos más importantes son:

Botón de ejecución.

Cancelación de ejecución.

Botón de pausa/continuación.

Configuración del texto.

Alineamiento de objetos.

Distribución de objetos.

Reordenamiento.

En la siguiente imagen se observa el programa en ejecución. Al pulsar el pulsador 1, se enciende el indicador correspondiente. Al pulsar el otro pulsador se enciende su indicador y, al estar ambos activos simultáneamente, se enciende el LED.



Figura 77. Panel frontal con el programa ejecutándose. Fuente: Elaboración propia.

3.2. Hardware empleado

3.2.1 Raspberry Pi

La tarjeta escogida para realizar el trabajo es la Raspberry Pi modelo 3 B+.

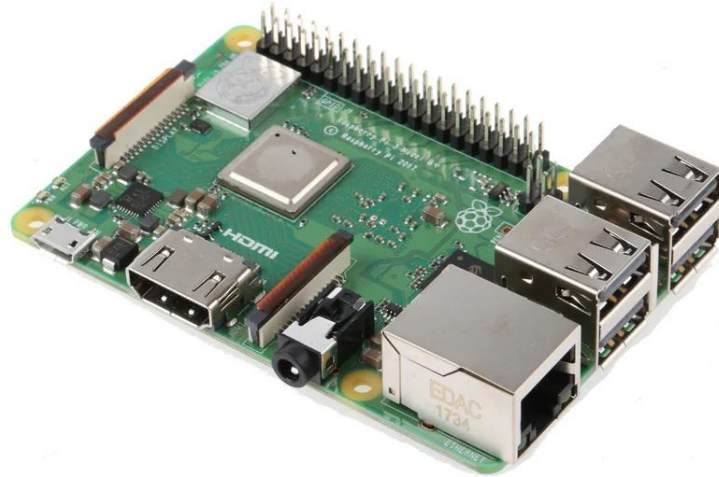


Figura 78. Raspberry Pi modelo 3B+. Fuente: [30]

Las **características** principales de la tarjeta elegida son:

Procesador ARMv8 de 64 bits con 4 núcleos a 1.4 GHz modelo BCM2837B0.

GPU VideoCore IV a 400 MHz.

Memoria de 1GB LPDDR2 SDRAM.

Conectividad inalámbrica: 2.4GHz / 5GHz.

Conexión Wifi 802.11.n.

Bluetooth 4.2, BLE.

Puertos:

- GPIO 40 pines (pines de entrada o salida con fines genéricos).
- HDMI.
- 4 x USB 2.0.
- CSI para cámara.
- DSI para pantalla táctil.

- Puerto Jack 3.5mm para audio o salida de vídeo.
- Almacenamiento interno a través de Micro SD.
- Micro USB para la alimentación.
- Puerto Ethernet.

El puerto HDMI se utilizará para conectarle una pantalla y visualizar la interfaz del sistema operativo.

Dos puertos USB para teclado y ratón, para configurar la Raspberry por primera vez.

MicroSD Sandisk Ultra de 16GB.

Fuente de alimentación de 5.1V, con una salida, 2.5A, 13W.

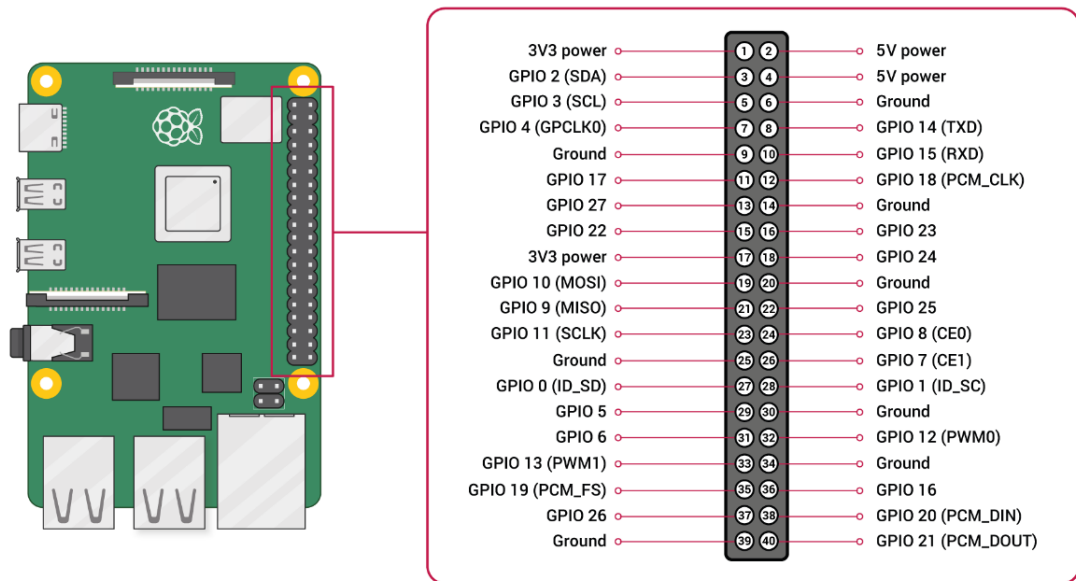


Figura 79. Pines de la Raspberry Pi. Fuente: [31]

3.2.2. Motor paso a paso y driver

El **motor** escogido es el 28BYJ-48, la versión de 5V. Es un motor de precio y características muy reducidas, pero útil para mostrar el funcionamiento de los motores paso a paso, sin necesidad de soportar grandes corrientes ni mover grandes cargas.

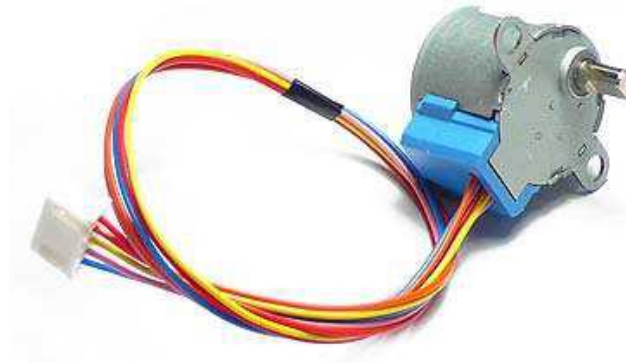


Figura 80. Motor 28BYJ-48 5V. Fuente: [32]

Sus **características principales** son:

Tensión nominal: 5V.

Motor paso a paso unipolar de 5 cables.

Ángulo de paso: 5.625° . Este dato es el mínimo ángulo de paso que gira el motor, es decir, en funcionamiento a medios pasos.

Reductora 1/64 (realmente 1/63,6875).

Frecuencia nominal: 100Hz.

Resistencia: $50\Omega \pm 7\%$ (25°C).

El resto de parámetros se encuentran en la hoja de características, incluida en los anexos.

El **driver** empleado es el ULN2003AN, diseñado específicamente para el motor 28BYJ-48.

El driver se utiliza para aumentar la corriente que circula por el motor, ya que el microcontrolador maneja unas intensidades del orden de los pocos miliamperios y tensiones pequeñas. En este caso el motor es de 5V, pero podría usarse la versión de 10V sin problema.

Internamente está formado por 7 transistores NPN Darlington con diodos de protección para cargas inductivas y evitar picos de tensión. Cada transistor es capaz de manejar hasta 500mA. También incluye 4 diodos LED para identificar que bobina está siendo alimentada en ese momento.

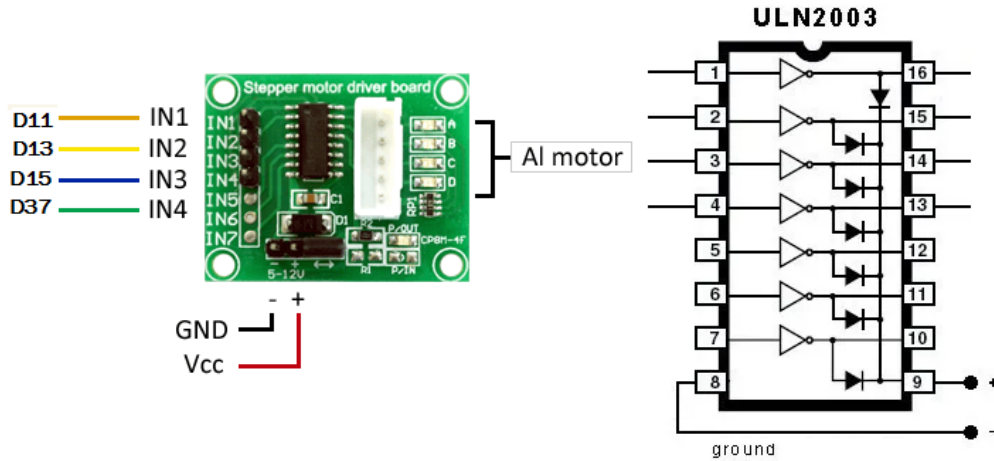


Figura 81. Driver ULN2003AN. Fuentes: [33] y [34]

El transistor Darlington es un dispositivo semiconductor que combina dos transistores bipolares en uno. Se utiliza para amplificar con mayor ganancia las corrientes con las que se trabaja. Es ideal para controlar con corrientes muy pequeñas cargas que consumen grandes corrientes (por ejemplo, para el control de motores de corriente continua).

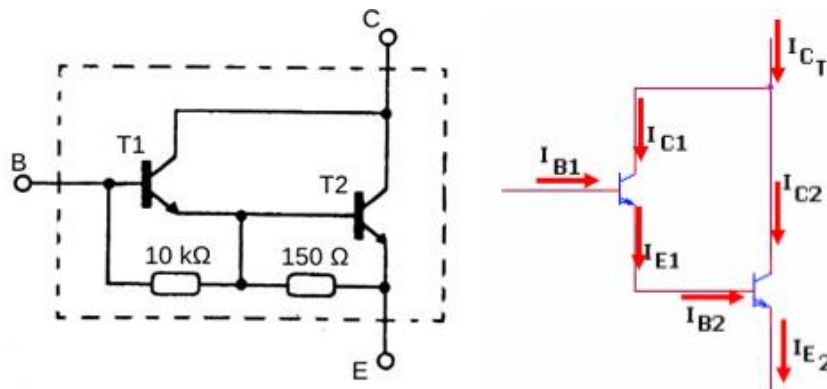


Figura 82. Transistor Darlington. Fuente: [38]

El principio de funcionamiento se comenta a continuación con las expresiones de los transistores.

El transistor 1 tiene una Base, Emisor, Colector y β 1.

El transistor 2 tiene una Base, Emisor, Colector y β 2.

Ecuaciones del transistor 1:

$$I_{C1} = \beta_1 \cdot I_{B1}$$

$$I_{E1} = I_{B1} + I_{C1} = (\beta_1 + 1) \cdot I_{B1}$$

Ecuaciones del transistor 2:

$$I_{C2} = \beta_2 \cdot I_{B2}$$

$$I_{E2} = I_{B2} + I_{C2} = (\beta_2 + 1) \cdot I_{B2}$$

Condiciones de la configuración Darlington:

$$I_{Ctot} = I_{C1} + I_{C2}$$

$$I_{B2} = I_{E1}$$

Suponiendo β_1 y β_2 lo suficientemente grandes para que $\beta_1 + 1 \approx \beta_1$ y $\beta_2 + 1 \approx \beta_2$, las ecuaciones quedarían de la siguiente forma:

$$I_{E2} = (\beta_2 + 1)I_{B2} = (\beta_2 + 1) \cdot (\beta_1 + 1) \cdot I_{B1} \approx \beta_2 \cdot \beta_1 \cdot I_{C1} = \beta_{tot} \cdot I_{B1}$$

$$I_{Ctot} = (\beta_1 + 1) \cdot I_{B1} + (\beta_2 + 1) \cdot I_{B2} \approx \beta_1 \cdot I_{B1} + \beta_2 \cdot I_{B2} \rightarrow$$

$$I_{Ctot} = \beta_1 \cdot I_{B1} + \beta_2 \cdot (\beta_1 + 1) \cdot I_{B1} \approx \beta_1 \cdot I_{B1} + \beta_2 \cdot \beta_1 \cdot I_{B1} =$$

$$(\beta_2 + 1) \cdot \beta_1 \cdot I_{B1} \approx \beta_2 \cdot \beta_1 \cdot I_{B1} = \beta_{tot} \cdot I_{B1}$$

La corriente de emisor del transistor Darlington se amplifica aproximadamente $\beta_2 \cdot \beta_1$ veces la corriente de base.

Los **inconvenientes** de esta configuración son:

La tensión de polarización, para que conduzca y amplifique, ha de ser el doble que la convencional.

Al tener que activar las dos uniones B-E, el transistor Darlington es más lento que uno normal, y también es más lento en dejar de conducir.

Se suelen colocar **resistencias** como en la figura 82, ya que al trabajar en conmutación necesitan un tiempo finito y se produce una capacidad parásita entre Base-Emisor 1 y Base-Emisor 2. Esas capacidades se descargan a través de las resistencias. La resistencia entre Base-Emisor 1 suele ser del orden de los $k\Omega$ y a otra más pequeña.

3.2.3. Otros componentes

Para facilitar las conexiones se utilizará una protoboard o placa de pruebas, cables para la unión entre la Raspberry y el driver mediante la protoboard, diodos led, resistencias y pulsadores para comprobar el funcionamiento antes de utilizar el motor real.

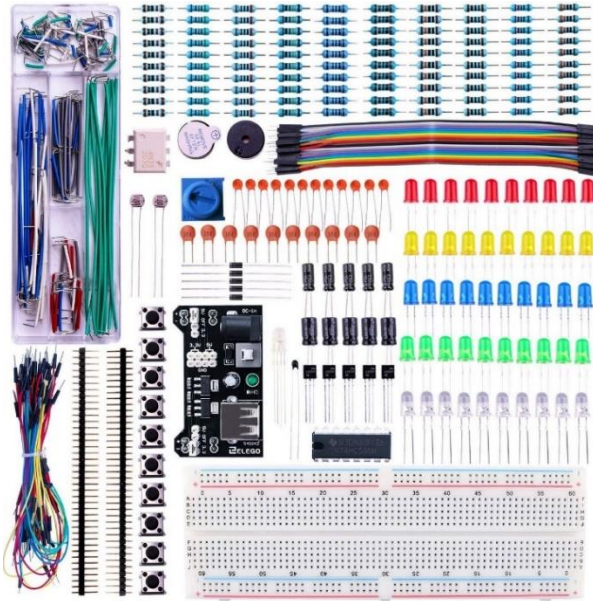


Figura 83. Protoboard y componentes. Fuente: [35]

La estructura interna de la protoboard se indica en la figura 84, simplemente hay puntos unidos entre sí para facilitar la conexión serie y paralelo entre elementos (los puntos unidos se encuentran al mismo potencial).

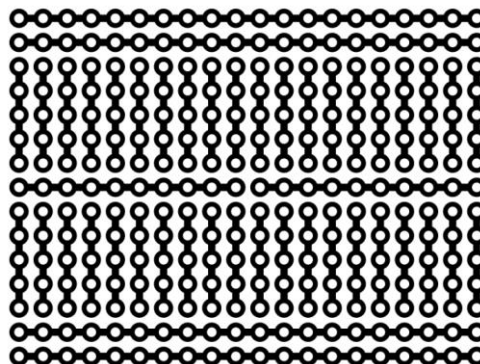


Figura 84. Estructura interna de la protoboard. Fuente: [36]

3.3. Software desarrollado

En este apartado se mostrará paso por paso como se ha realizado el programa, describiendo el panel frontal y el diagrama de bloques realizado.

También se explica cómo conectar la Raspberry con LabVIEW y cómo instalar el programa en la tarjeta.

Finalmente se describen los bloques utilizados para habilitar la interconexión con la Raspberry.

El panel frontal tiene el siguiente aspecto, bastante visual y es el lugar donde se introducirán las entradas (grados a girar, sentido horario o antihorario, pasos completos o medios pasos y la velocidad de giro deseada) y se visualizarán las salidas (velocidad de giro, grados girados actualmente y leds indicadores de la secuencia de activación de IN1, IN2, IN3 e IN4)

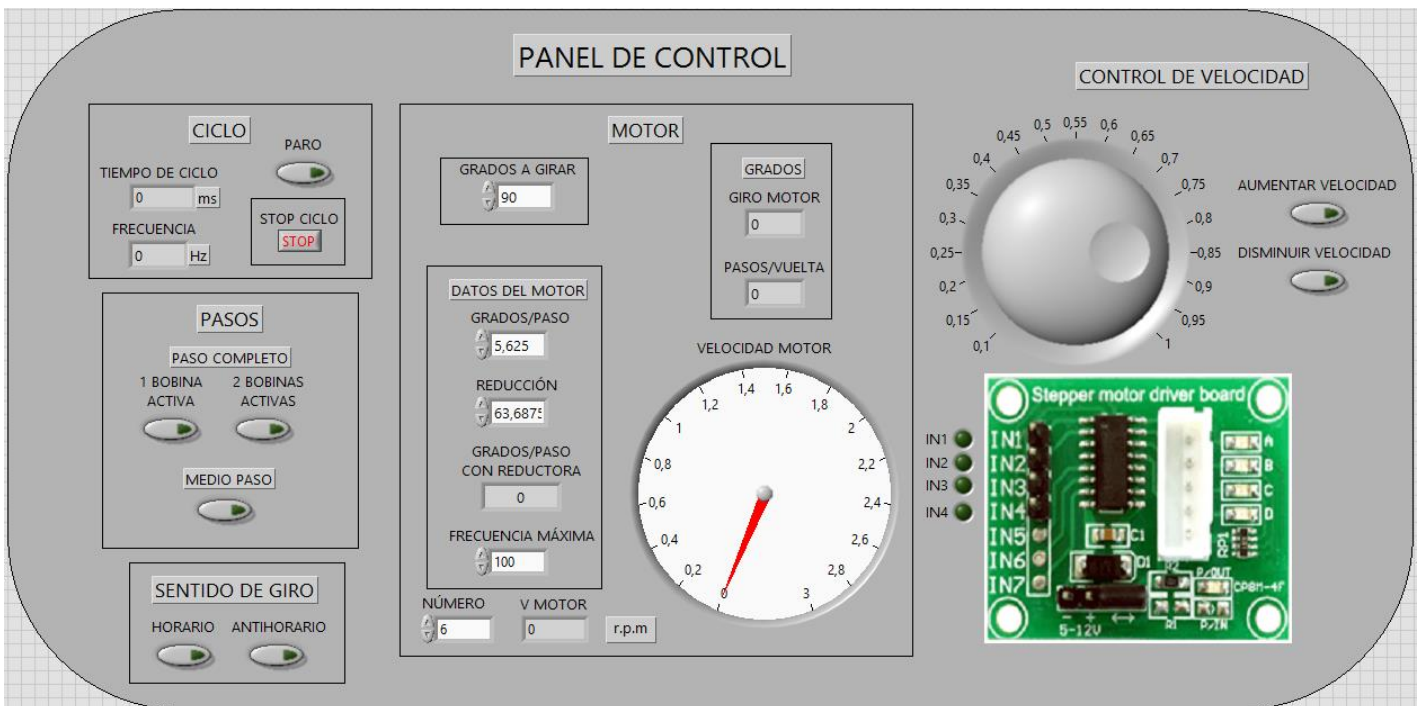


Figura 85. Aspecto del panel frontal. Fuente: Elaboración propia.

Para el control del motor paso a paso unipolar se podrá elegir el sentido de giro y la forma en que se realiza, si es modo paso completo o medio paso



Figura 86. Sentido de giro y modo de control del panel frontal. Fuente: Elaboración propia.

La velocidad de giro del motor dependerá de la frecuencia con la que se envíen los pulsos al motor, que dependerá del tiempo de ejecución del programa, con el control de velocidad se consigue variar entre 0 y 100Hz la velocidad del motor, lo que equivale a 10ms de duración de cada ciclo.

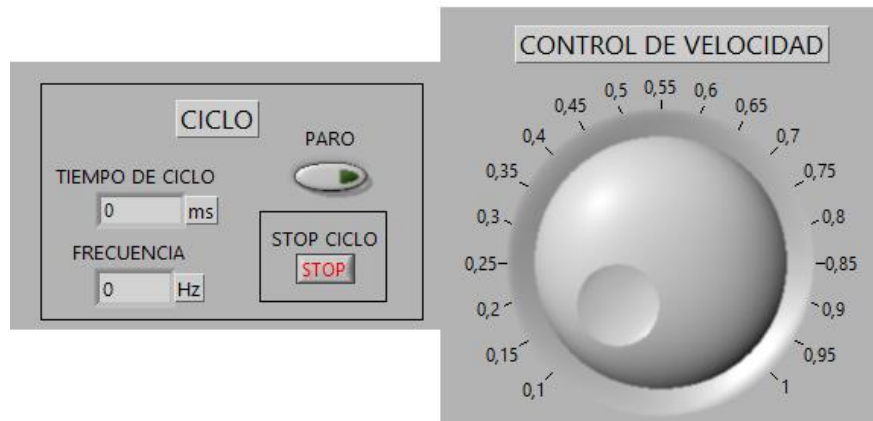


Figura 87. Variables del ciclo y control de velocidad en panel frontal. Fuente: Elaboración propia.

Se actualizan por defecto al ejecutar el programa y utilizando el control de velocidad. El botón de paro desmarca las opciones de giro del motor y el botón de STOP finaliza el ciclo.

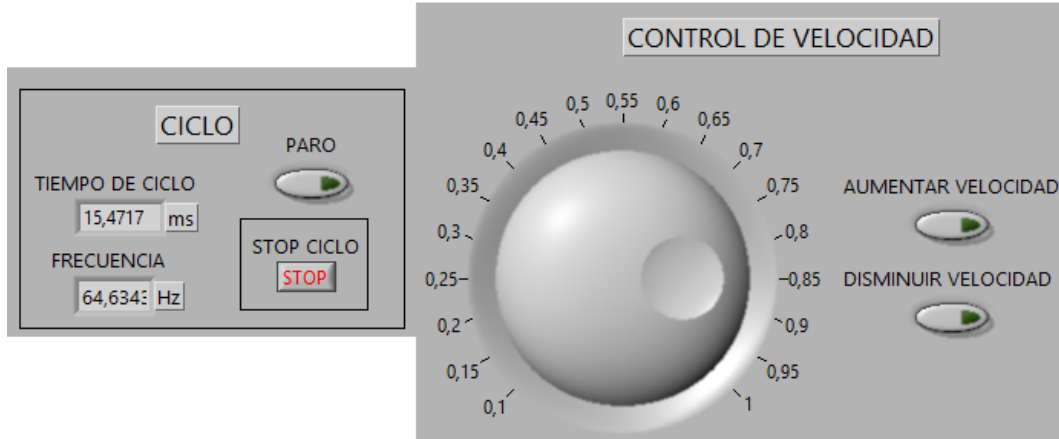


Figura 88. Variables del ciclo y control de velocidad en panel frontal en ejecución. Fuente: Elaboración propia.

Al modificar el control de velocidad, se actualiza el velocímetro y también se muestra la velocidad exacta del motor en revoluciones por minuto.

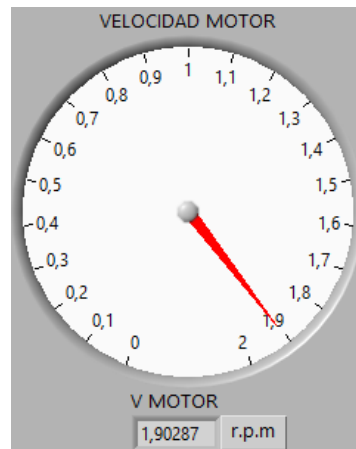


Figura 89. Velocímetro e indicador de velocidad en el panel frontal ejecutándose el programa. Fuente: Elaboración propia.

Indicador de qué bobina o pares de bobinas están siendo alimentadas y de la secuencia de activación, con la imagen del driver utilizado.



Figura 90. Indicadores y driver en el panel frontal mientras se ejecuta el programa. Fuente: Elaboración propia.

También se añaden los datos relacionados con el motor. Para ejecutar el programa, primero se debe seleccionar los grados que se desean girar y, posteriormente, la forma en que se hace, si es en sentido horario o antihorario y si es a pasos completos o medios pasos.

Al ejecutar el programa, se actualizan los datos del motor que ofrece el fabricante. Lo que más interesa para la realización del programa son los grados que gira por cada paso y la reducción con la reductora que lleva, de relación 1/64 en este caso. Si no tuviera reductora, habría que introducir un 1. Una vez elegido si se realiza a pasos completos o medios pasos, se indica el número de pasos por vuelta que debe dar el motor y se actualiza cada tiempo de ciclo los grados que lleva girados el motor.

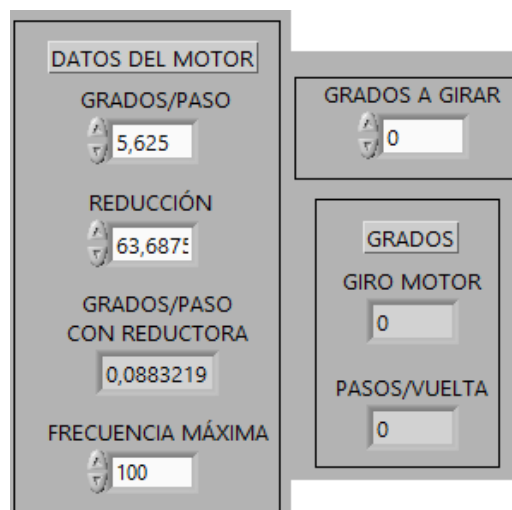


Figura 91. Datos del motor y grados en el panel frontal mientras se ejecuta el programa. Fuente: Elaboración propia.

Para que todo esto ocurra, se debe explicar la pantalla del diagrama de bloques

Primero se crea un bucle **while** en el que irá dentro lo que se comenta a continuación. Esto hará que se repita cíclicamente el programa y termine cuando se pulse el botón de STOP.

Aquí se muestran las opciones del control:

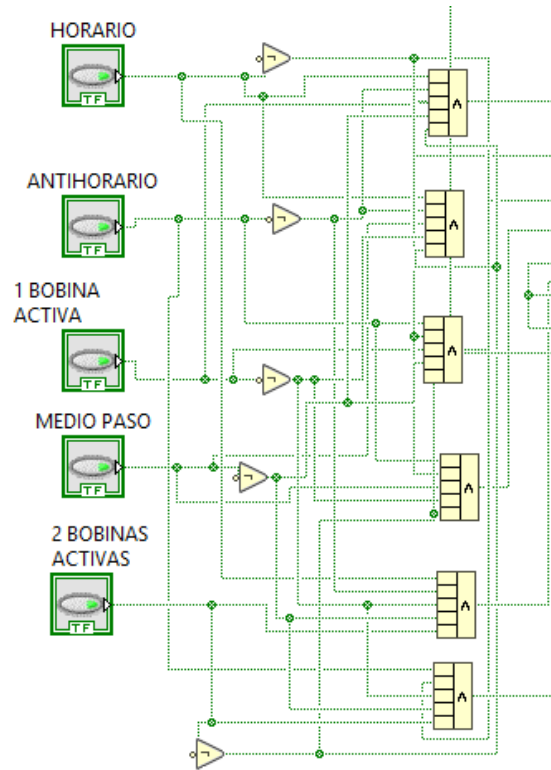


Figura 92. Sentido de giro y modo de control. Fuente: Elaboración propia.

Si se elige sentido horario y paso completo con una bobina activa, la primera operación AND devolverá un 1 y el resto 0.

Si se escoge sentido antihorario y medio paso, el quinto AND devolverá un 1 y el resto 0.

Con el operando NOT en serie con la opción elegida, devolverá un cero al resto de operaciones AND.

Después se utilizará otra operación AND para cada respuesta. Devolverá un 1 si los grados que lleva girados el motor son menores que los grados a girar deseados.

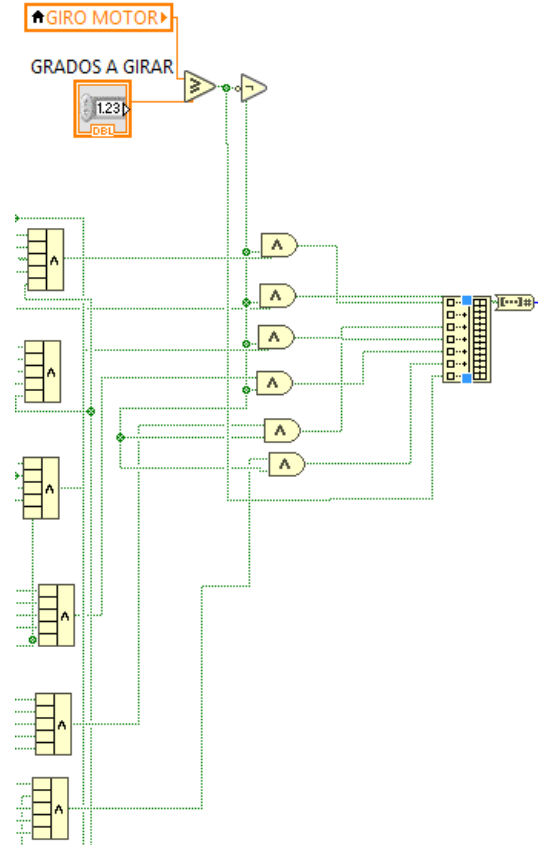


Figura 93. Condición de grados a girar \geq grados girados. Fuente: Elaboración propia.

Después, todas las respuestas y la condición de grados girados \geq grados a girar se llevan a un **Build Array** y, seguidamente, a un **Boolean Array To Number**.

El bloque Build Array creará una matriz de una sola columna con las 7 entradas que le llegan. Estas entradas, por la forma que se ha realizado el programa, enviarán o 7 ceros, o un uno y el resto ceros, es decir, valores booleanos.

El bloque Boolean Array To Number convierte esa matriz en un número.

A continuación, se muestra una tabla con las posibilidades:

HORARIO	ANTIHORARIO	PASO COMPLETO UNA BOBINA ACTIVA	PASO COMPLETO DOS BOBINAS ACTIVAS	MEDIO PASO	NÚMERO RESULTANTE
1	0	1	0	0	1
1	0	0	1	0	4
1	0	0	0	1	2
0	1	1	0	0	8
0	1	0	1	0	32
0	1	0	0	1	16

Tabla 25. Número resultante según la secuencia escogida. Fuente: Elaboración propia.

Si devuelve un 0 significa que no se ha escogido opción de sentido ni forma de giro.

Si devuelve un 64, se ha alcanzado la condición de grados girados \geq grados a girar y el motor dejará de girar.

Las variables conocidas del motor se indican a continuación.

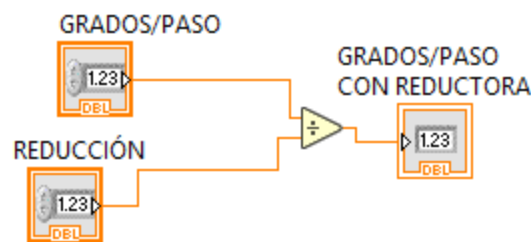


Figura 94. Características del motor. Fuente: Elaboración propia.

Hay que tener en cuenta a la hora de realizar el programa que los grados/paso que ofrece el fabricante son los grados mínimos que puede girar el motor, es decir, son los grados que gira en modo medio paso, por lo que en modo paso completo girará el doble.

A continuación, se muestra una serie de operaciones clave para la ejecución del programa:

El control de velocidad se puede modificar mientras se ejecuta el programa. Varía entre 0 y 1, por eso se multiplica por la frecuencia máxima, para que no exceda ese valor.

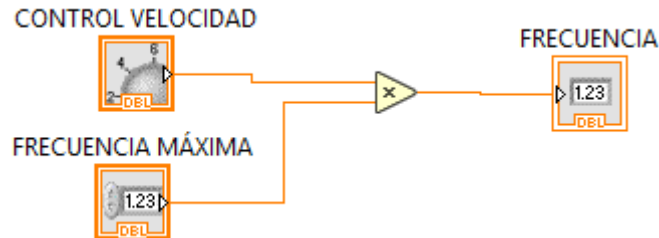


Figura 95. Obtención de la frecuencia. Fuente: Elaboración propia.

También se podrá variar la velocidad mediante dos pulsadores, imponiendo las condiciones de no exceder los 100Hz ni de ser inferior a 0Hz.

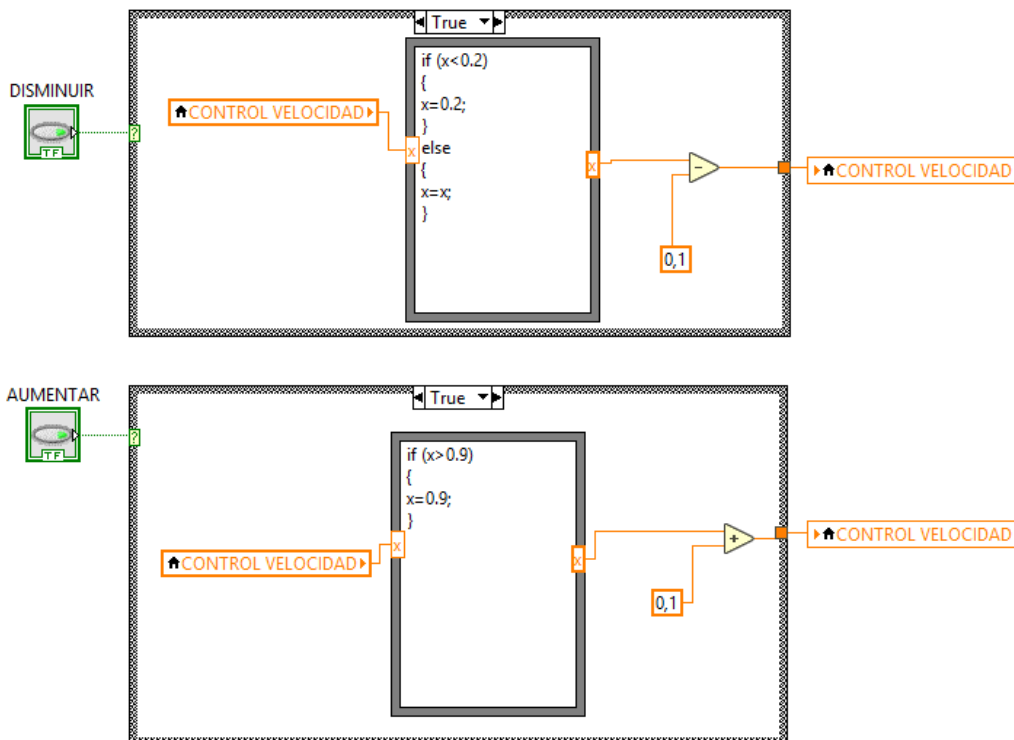


Figura 96. Control de velocidad mediante dos pulsadores. Fuente: Elaboración propia.

Una vez colocado el bloque de la frecuencia se calcula el tiempo de ejecución de cada ciclo de la siguiente forma.

Para hallar los pasos por vuelta que necesita dar el motor en caso de paso completo se calculan de la manera que aparece en la figura 99.

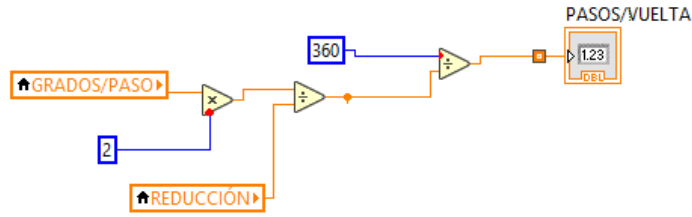


Figura 99. Obtención de pasos/vuelta necesarios para paso completo. Fuente: Elaboración propia.

Con este valor se hallará la velocidad a la que gira el motor y se podrá indicar en el panel frontal.

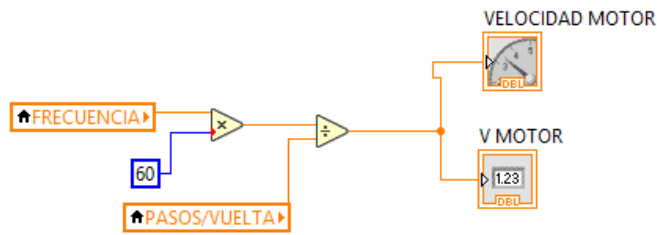


Figura 100. Obtención de la velocidad de giro. Fuente: Elaboración propia.

Se crea una condición con un IF, la cual le llega la cuenta de ciclos que se están realizando y ese número se lleva a otro CASE STRUCTURE con cuatro posibilidades.

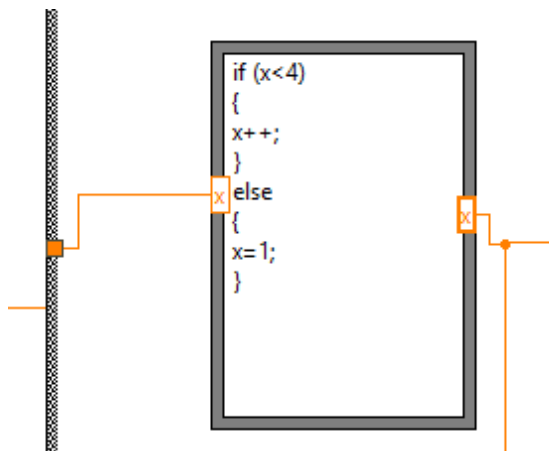


Figura 101. Condición IF para la secuencia de activación. Fuente: Elaboración propia.

Las flechas indican como se conectan los cables a los extremos del bucle while, que indican el número de ciclos completados.

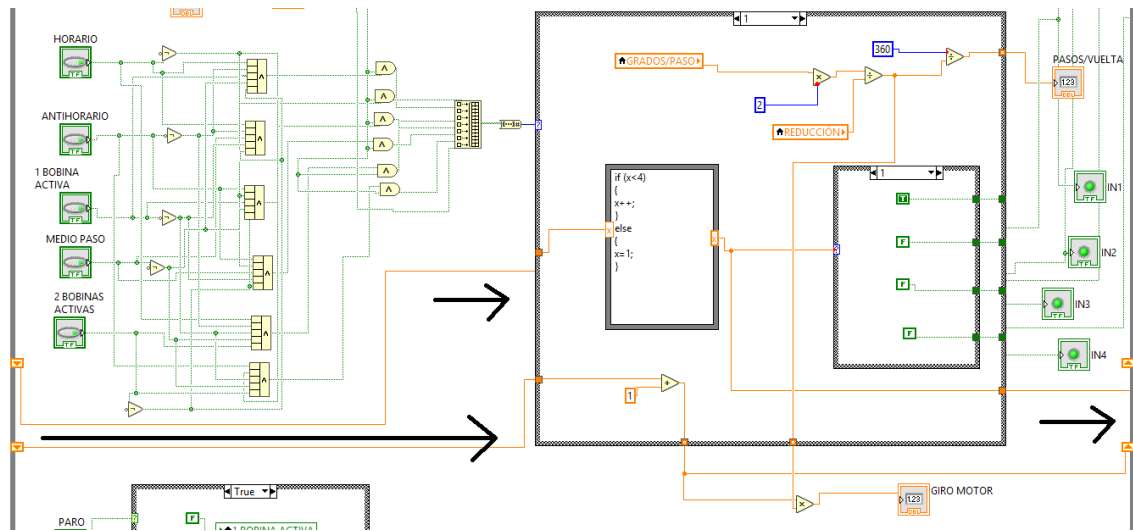


Figura 102. Conexión en los extremos del bucle while. Fuente: Elaboración propia.

Cada ciclo corresponderá a un paso y el IF hace que cuando supere el 4º ciclo, vuelva a empezar a 1 para crear la secuencia de activación de los pines IN necesaria.

Se observa que en el caso 1 se pone a TRUE el pin IN1, en el caso 2 se pone a TRUE el pin IN2 y así se haría la secuencia de paso completo con una bobina activa y sentido horario.

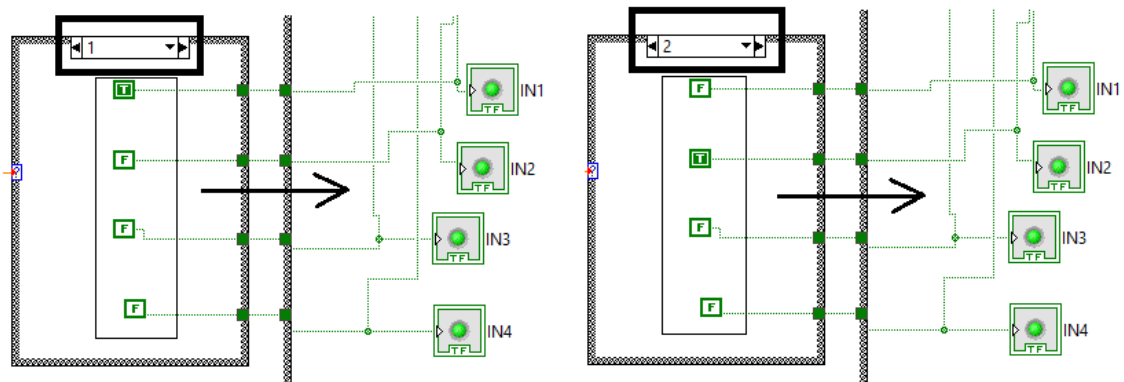


Figura 103. Activación de los pines correspondientes a sentido horario y paso completo con una bobina activa. Fuente: Elaboración propia.

Otra operación a tener en cuenta es la de calcular los grados que lleva girados el motor.

Para ello se conectan cables a los extremos del bucle while. El bloque suma se añade un 1 cada ciclo para indicar en el que se encuentra y este valor se lleva una salida al extremo derecho del bucle while y otro al bloque de multiplicación.

Al bloque de multiplicación le llegan los grados/paso a girar y así se calcularía los grados que el rotor ha girado en cada ciclo. En este caso cada ciclo avanzaría $\frac{5,625 \cdot 2}{63,6875} = 0,1637$ grados por paso.

En el segundo ciclo habrá avanzado $0,1637 \cdot (1 + 1) = 0,3274$ grados, así sucesivamente hasta llegar a los grados deseados.

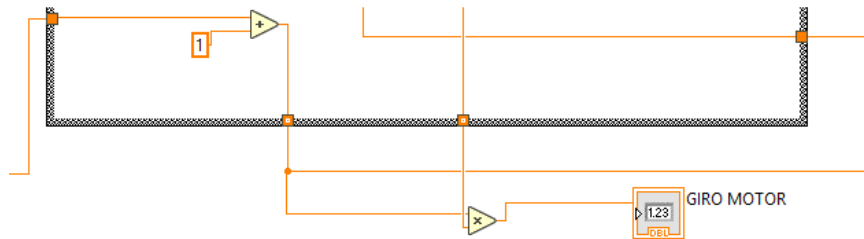


Figura 104. Obtención de los grados girados por el motor en un instante de tiempo. Fuente: Elaboración propia.

Para realizar el control con paso completo con una bobina activa en sentido antihorario, simplemente habría que cambiar la secuencia del case structure que se encuentra dentro del **caso 8**, como se indicó en la tabla 25.

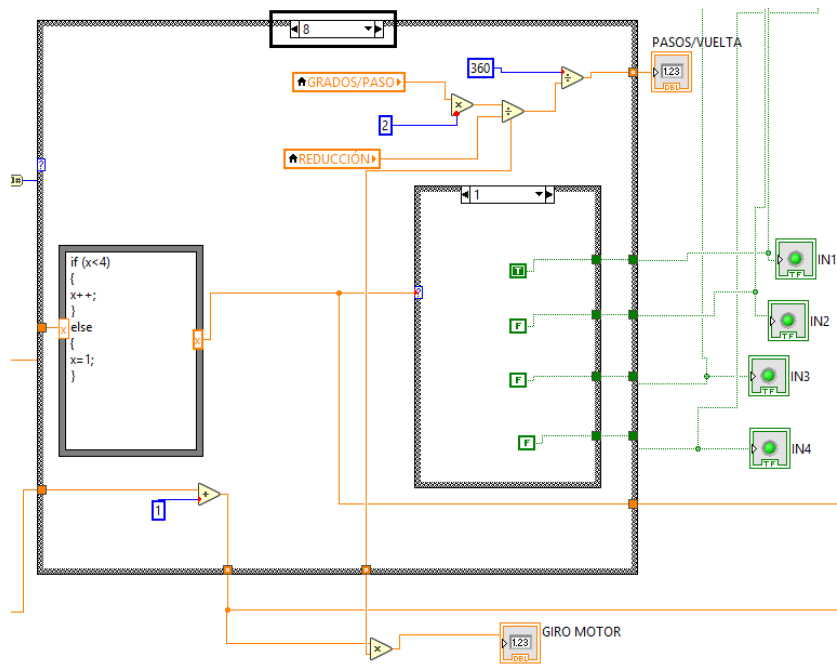


Figura 105. Case structure para sentido antihorario y paso completo una bobina activa. Fuente: Elaboración propia.

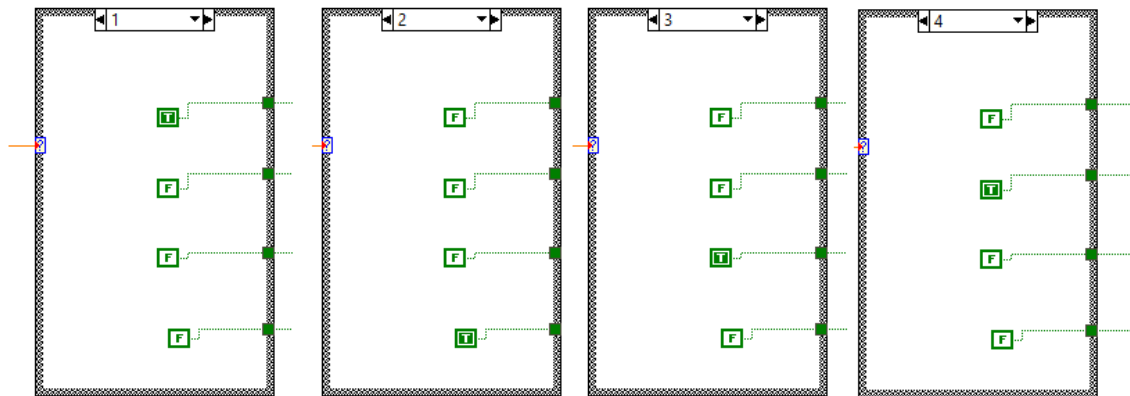


Figura 106. Secuencia de activación para sentido antihorario y paso completo con una bobina activa. Fuente: Elaboración propia.

Para el caso de paso completo y dos bobinas activas se realizaría de la misma manera cambiando únicamente la secuencia de activación en el case structure, dentro del **caso 4** y **32**, para sentido horario y antihorario, respectivamente.

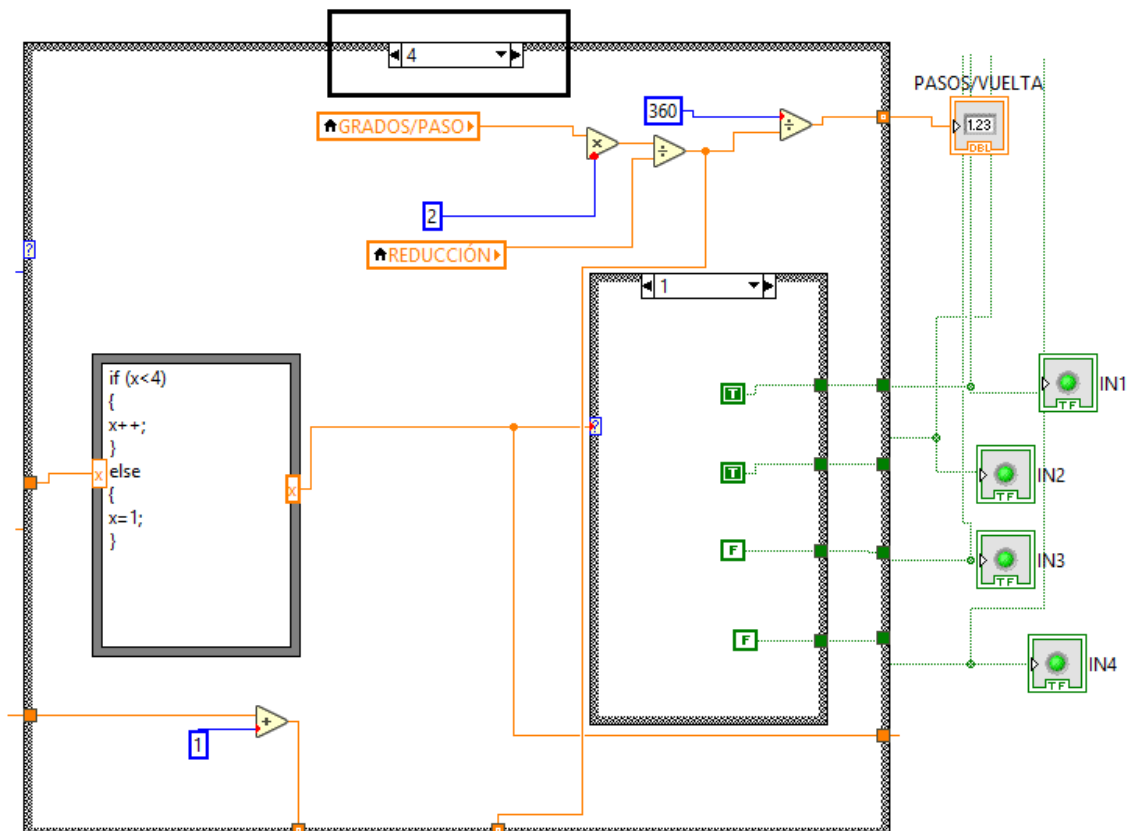


Figura 107. Case structure para sentido horario y paso completo con dos bobinas activas. Fuente: Elaboración propia.

Secuencia para sentido horario y paso completo con dos bobinas activas:

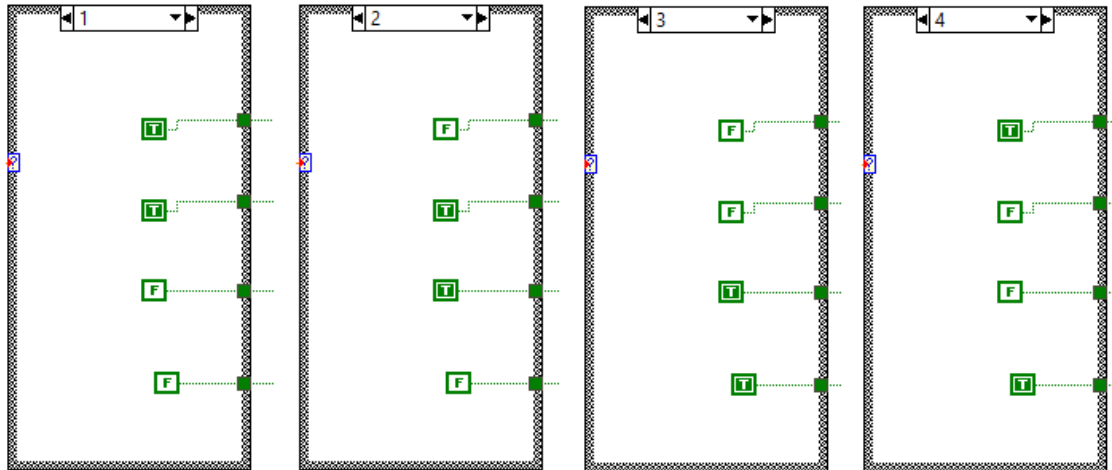


Figura 108. Secuencia para sentido horario y paso completo con dos bobinas activas. Fuente: Elaboración propia.

Para el sentido antihorario será el caso 32, igual que el 4 pero con la siguiente secuencia de activación:

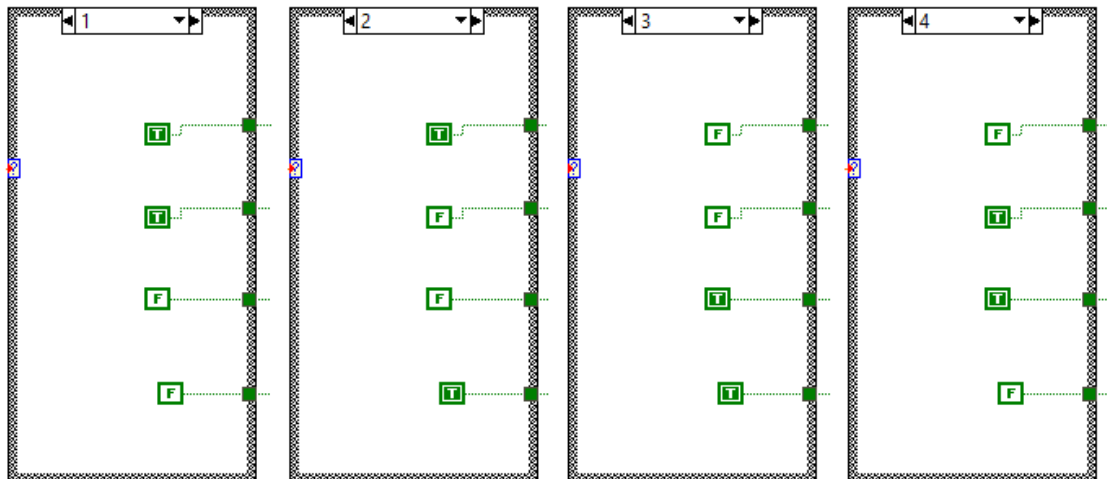


Figura 109. Secuencia para sentido antihorario y paso completo con dos bobinas activas. Fuente: Elaboración propia.

Para hacer el control con medios pasos es necesario ajustar los bucles del programa, ya que la secuencia son **ocho casos** (combinando la secuencia de paso completo con una bobina y con dos bobinas activas), por lo que el número de pasos/vuelta cambia y la velocidad del motor se reduce.

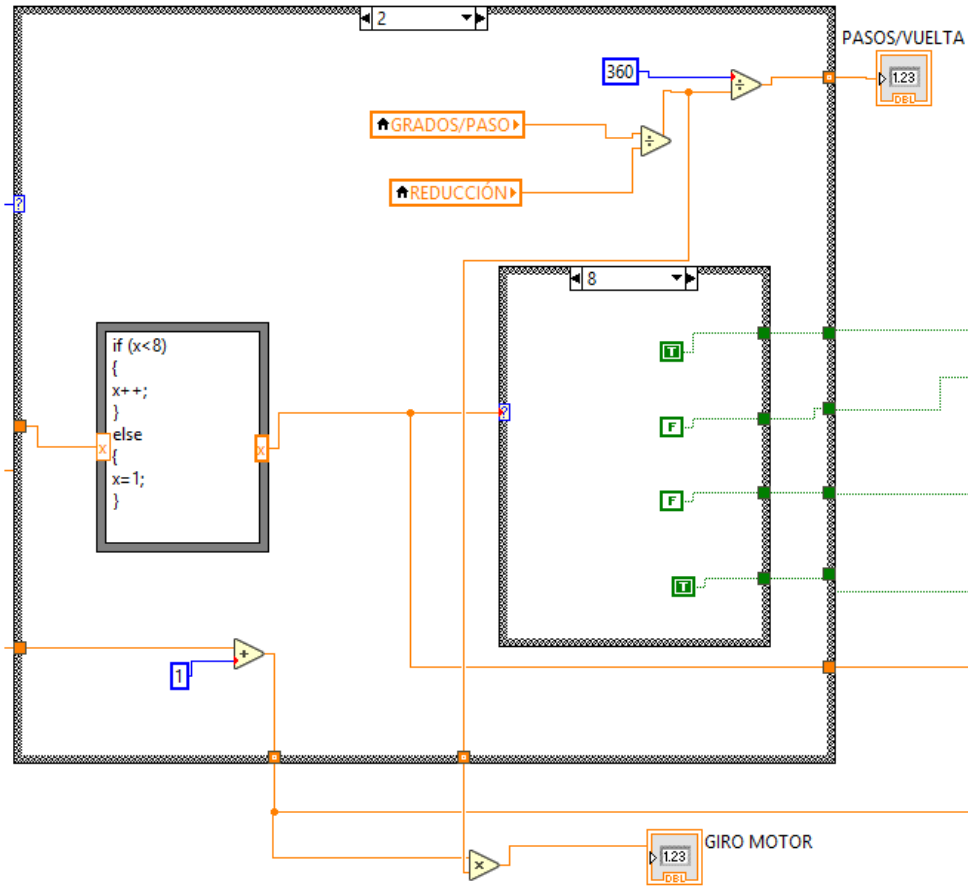


Figura 110. Esquema general de medio paso con sentido horario. Fuente: Elaboración propia.

La condición IF cambia para que al case structure le lleguen valores del 1 al 8.

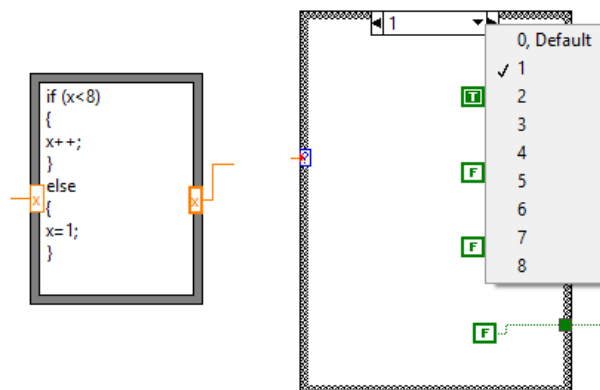


Figura 111. Estructura del IF para el control con medios pasos. Fuente: Elaboración propia.

La secuencia para que gire horario o antihorario cambiará, en la figura 112 se muestra cómo sería la secuencia de sentido horario:

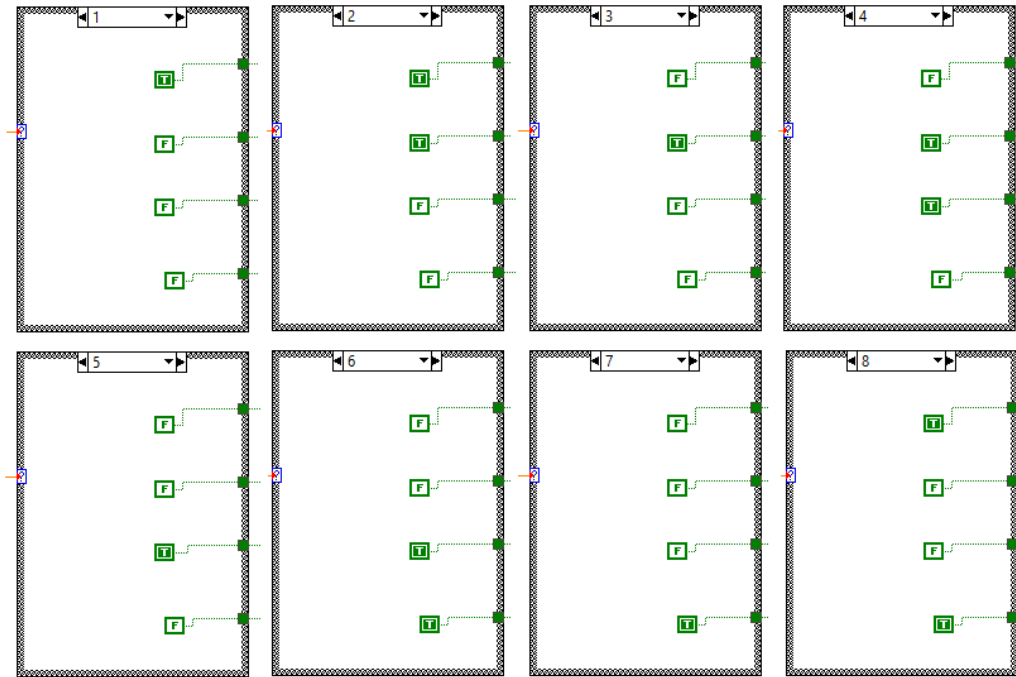


Figura 112. Secuencia para control con medios pasos sentido horario. Fuente: Elaboración propia.

En la figura 113 se muestra cómo sería la secuencia para el sentido antihorario:

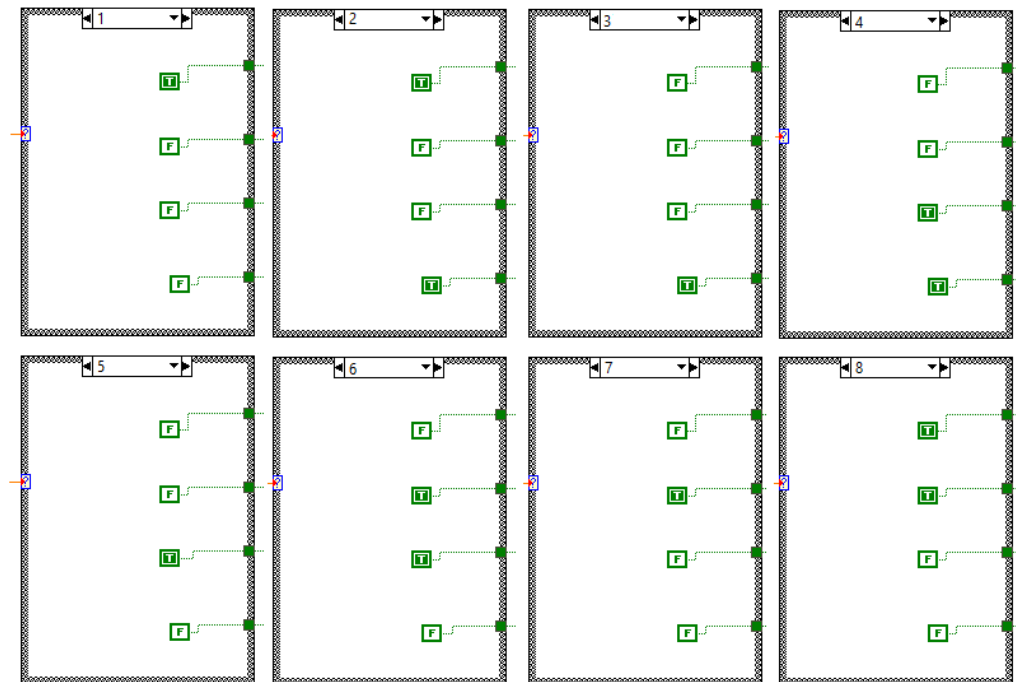


Figura 113. Secuencia para control con medios pasos sentido antihorario. Fuente: Elaboración propia.

El número de pasos/vuelta también será distinto, 4076 para medios pasos y 2038 para pasos completos en el caso del motor utilizado en este trabajo. Se observa claramente que a medios pasos necesita el doble de pasos/vuelta que a pasos completos

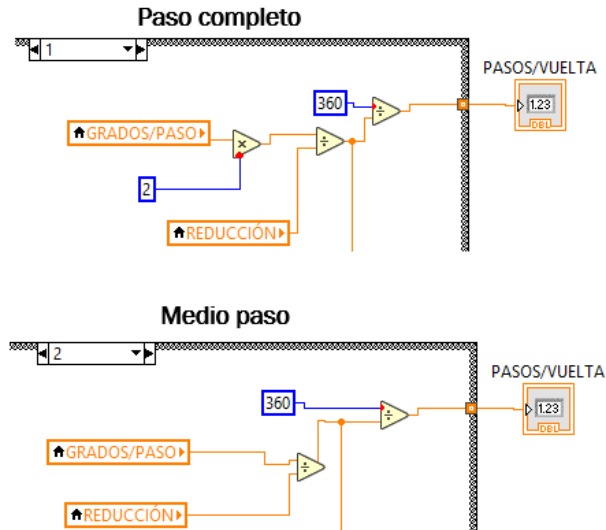


Figura 114. Comparación en la obtención de los pasos/vuelta. Fuente: Elaboración propia.

En el **caso 0**, todas las variables se encontrarán a 0 o FALSE.

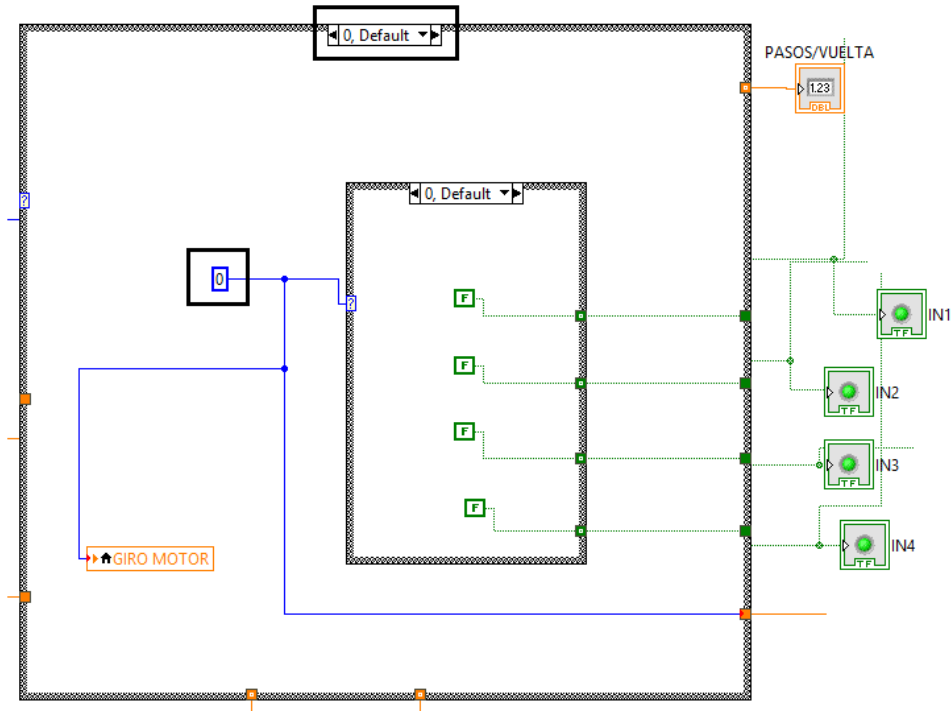


Figura 115. Caso 0 del case structure principal. Fuente: Elaboración propia.

En el **caso 64**, el motor habrá realizado los grados que se le han indicado girar, las variables se pondrán a 0 o FALSE y las opciones marcadas se desmarcarán automáticamente.

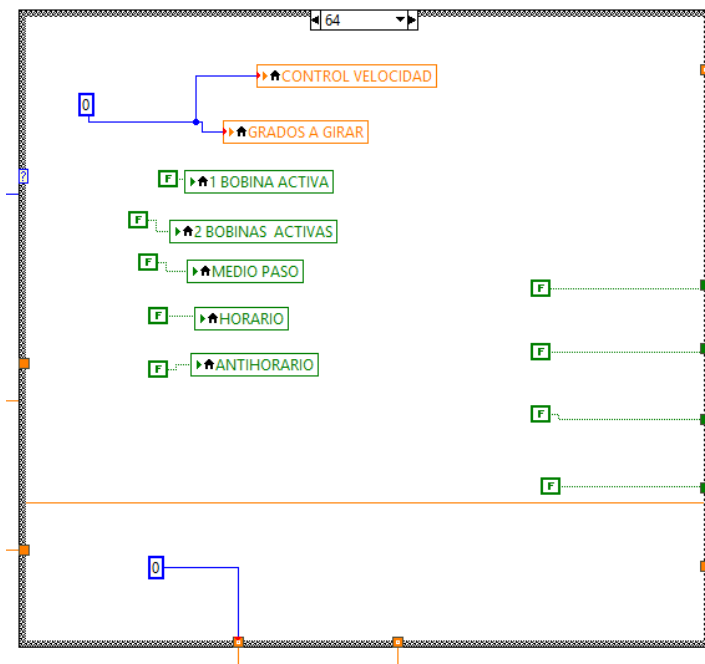


Figura 116. Caso 64 del case structure principal. Fuente: Elaboración propia.

También se tiene el **botón de paro**, que cambiará el estado de las opciones marcadas a 0 o FALSE, por lo que forzará al programa a situarse en el caso 0 del structure case principal.

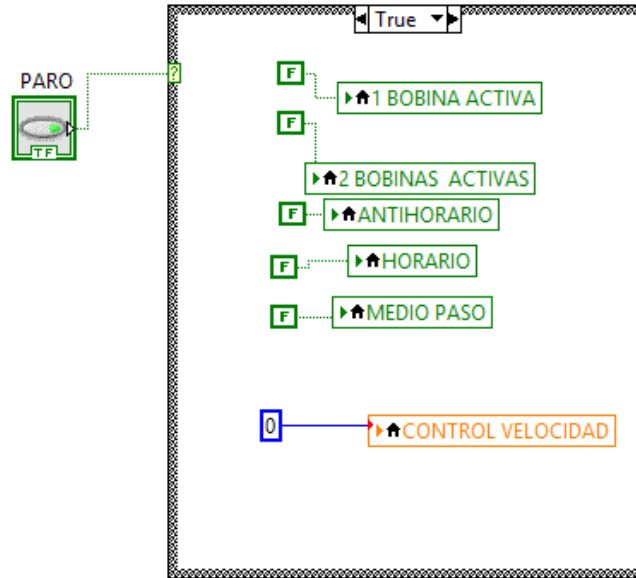


Figura 117. Función del botón de paro. Fuente: Elaboración propia.

Es obligatorio incluir una manera de detener el ciclo while, en este caso se utiliza un botón llamado **STOP CICLO**.

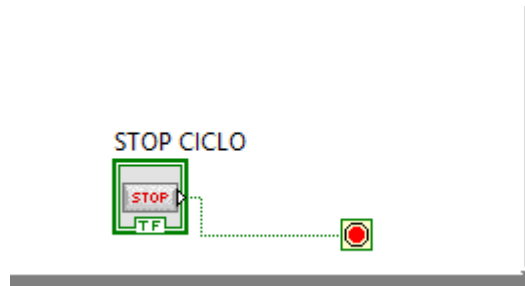


Figura 118. Botón STOP CICLO. Fuente: Elaboración propia.

Una vez realizado el programa es necesario configurar la tarjeta Raspberry para instalarle el programa de LabVIEW que se ha realizado.

Para ello hay que crear un nuevo proyecto:

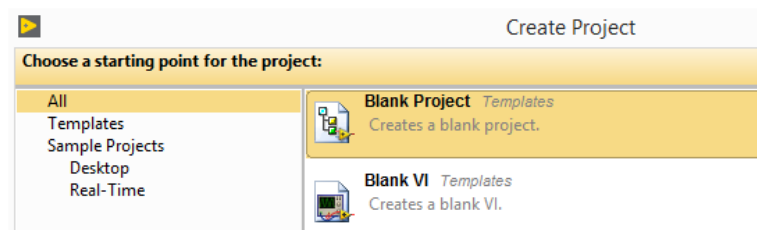
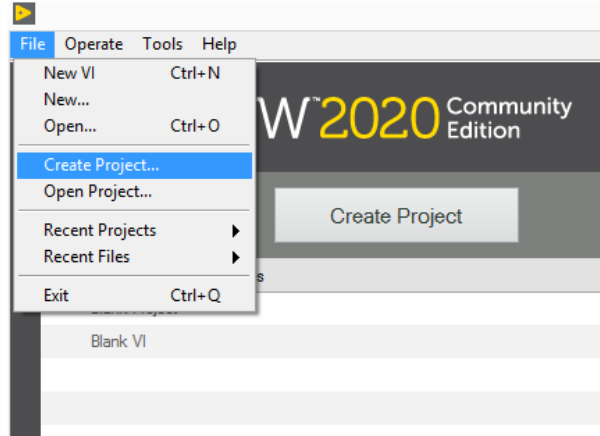


Figura 119. Creación de un nuevo proyecto en LabVIEW. Fuente: Elaboración propia.

Tendrá el aspecto de la figura 120. Botón derecho en el proyecto y se crea una nueva tarjeta o dispositivo.

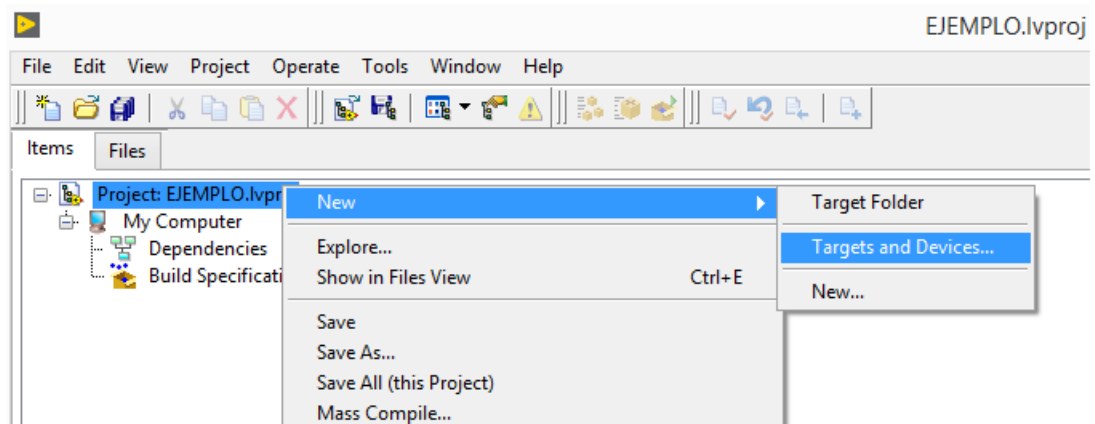


Figura 120. Creación de un nuevo dispositivo. Fuente: Elaboración propia.

Nueva tarjeta → LINUX → Raspberry Pi 2 B. Es válido para Raspberry Pi 3B+, que es el modelo a utilizar.

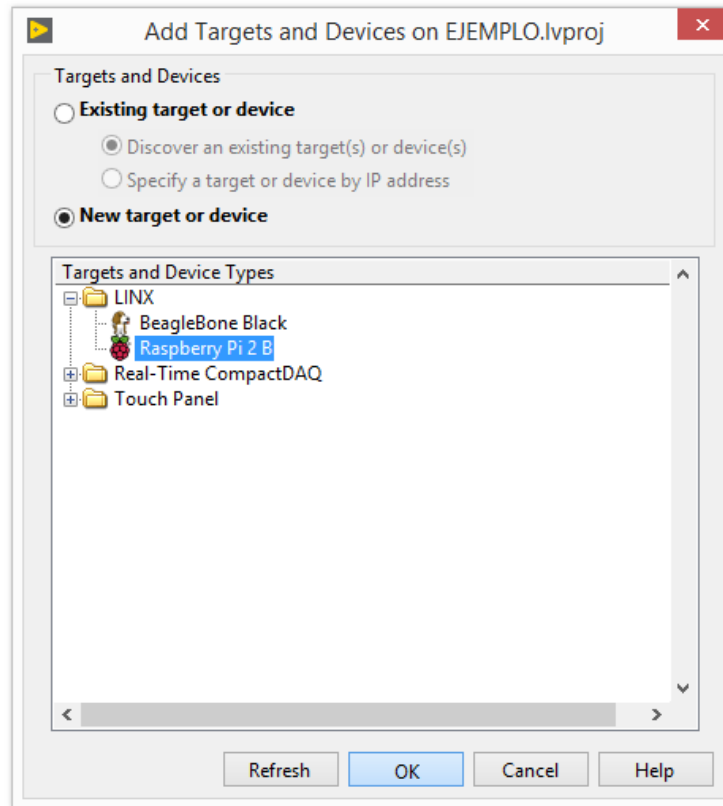


Figura 121. Elección de la tarjeta Raspberry Pi. Fuente: Elaboración propia.

Se ha creado en el proyecto un apartado nuevo, Raspberry Pi 2 B, al cual se arrastrará el VI realizado con el programa de control del motor paso a paso.

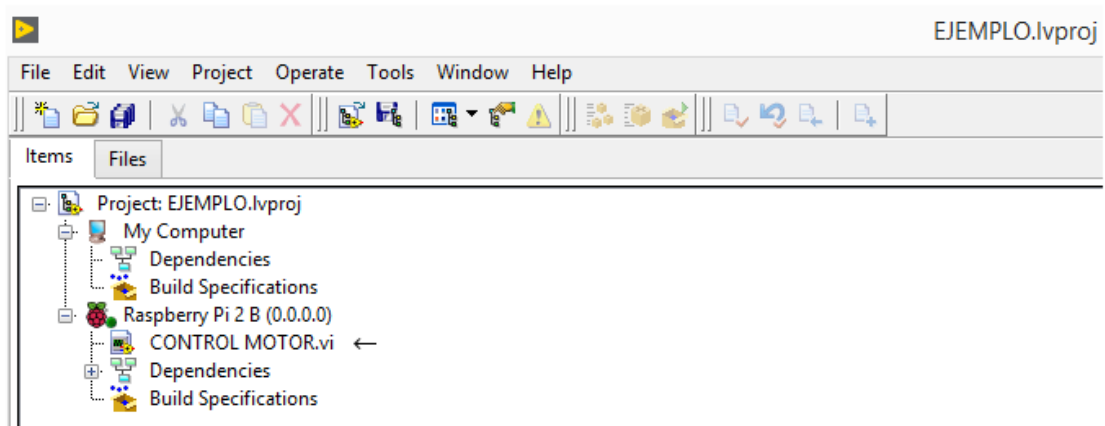


Figura 122. Colocación del VI dentro del apartado Raspberry Pi. Fuente: Elaboración propia.

Hecho esto, es necesario que la Raspberry esté conectada a la red eléctrica para poder configurarla en LabVIEW.

En Tools→ MakerHub→ LINX→ LINX Target Configuration aparecerá una pantalla para configurar la Raspberry.

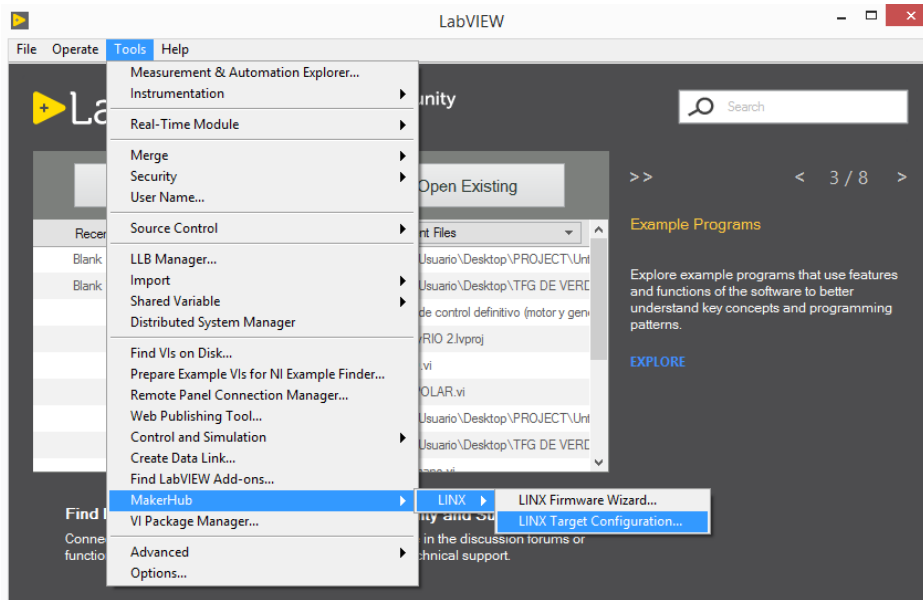


Figura 123. Opción Tools para la configuración de la Raspberry Pi. Fuente: Elaboración propia.

En esa pantalla será necesario escribir la IP de la tarjeta, el usuario y la contraseña. Por defecto, el usuario es "pi" y la contraseña "raspberrypi". La IP seleccionada por el router se encuentra, una vez configurada la Raspberry, escribiendo en la línea de comandos ifconfig, en este caso la IP es 168.192.1.245.

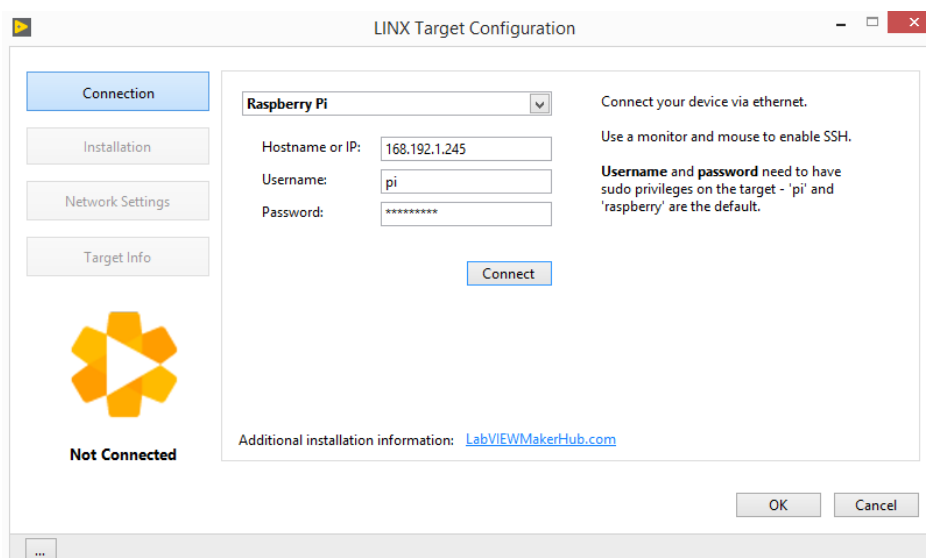


Figura 124. Configuración de la Raspberry Pi. Fuente: Elaboración propia.

Una vez introducidos esos datos se pulsa en Connect. Si se han escrito correctamente, se habilitan más opciones y aparece el logo de Raspberry.

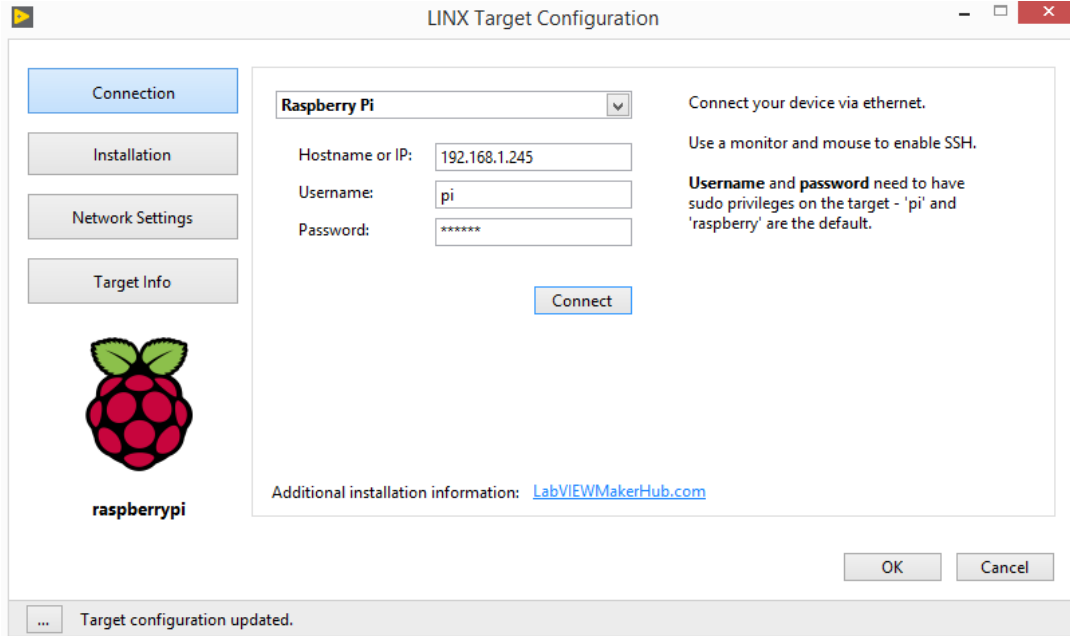


Figura 125. Configuración de la Raspberry Pi finalizada. Fuente: Elaboración propia.

En el proyecto, botón derecho en el apartado de Raspberry → properties.

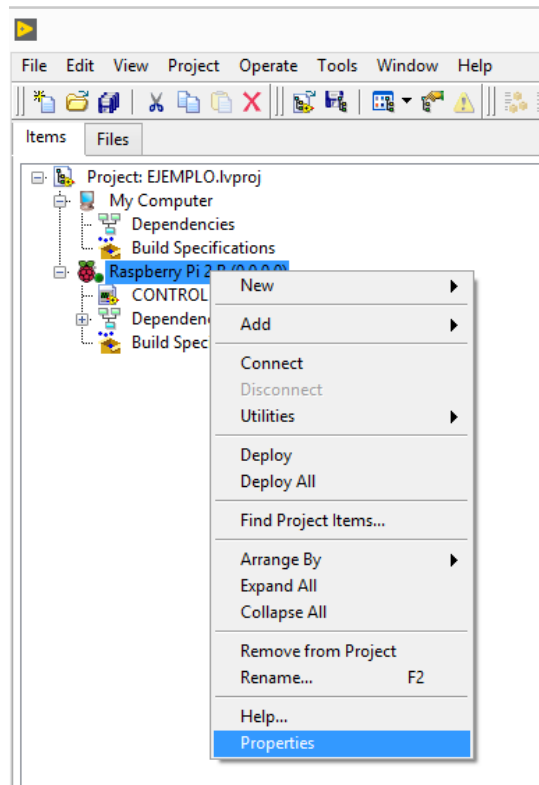


Figura 126. Propiedades del apartado Raspberry en el proyecto. Fuente: Elaboración propia.

Se abrirá una ventana en la que habrá que escribir la IP de la tarjeta de nuevo.

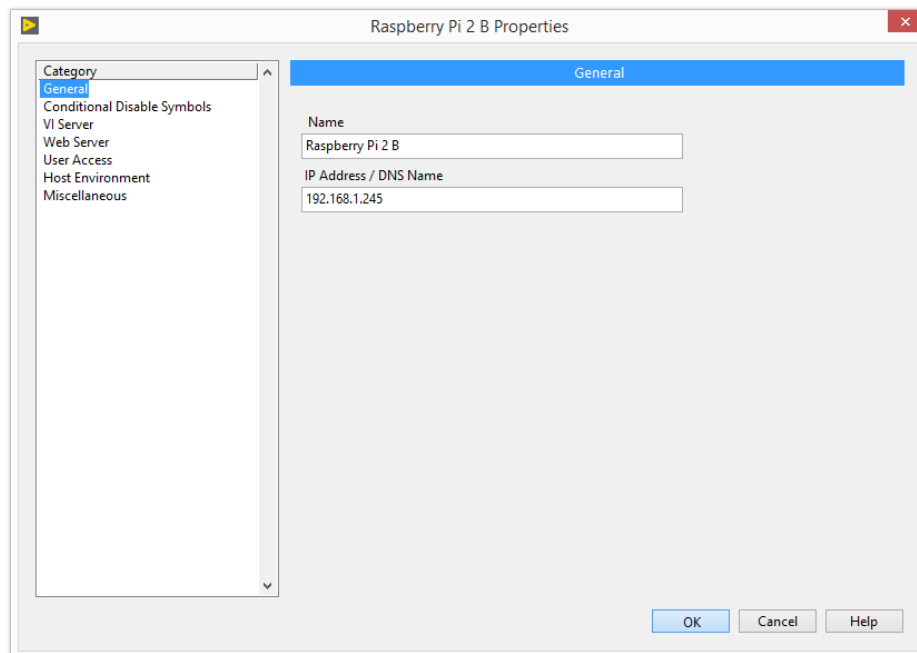


Figura 127. Propiedades generales del apartado Raspberry en el proyecto. Fuente: Elaboración propia.

Hecho esto, en la pantalla del proyecto el apartado de Raspberry pulsar Connect.

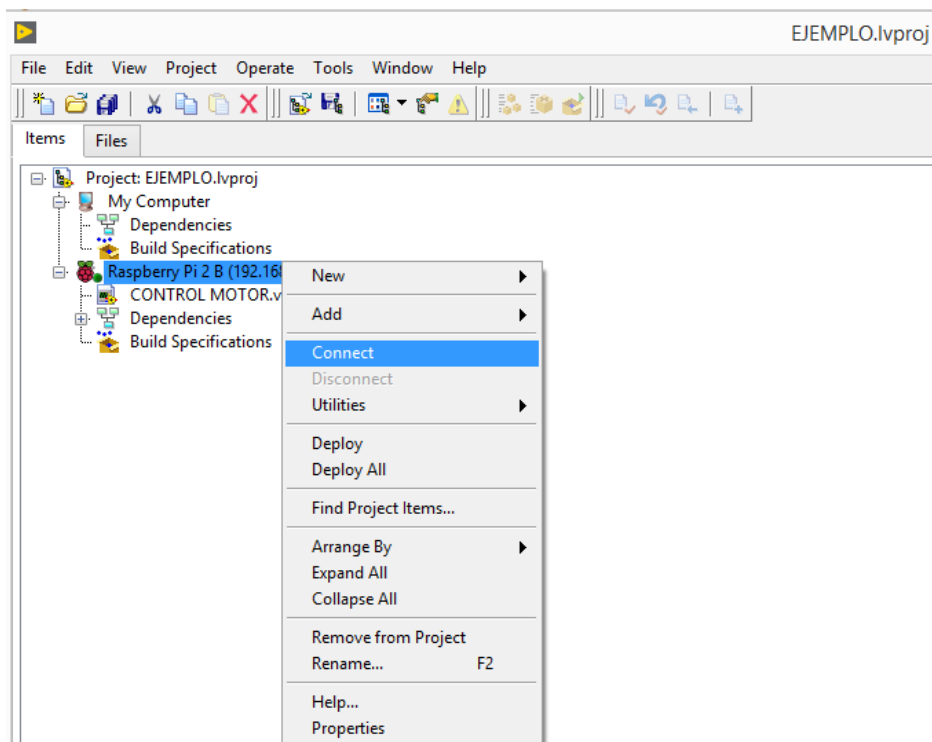


Figura 128. Conexión con la tarjeta Raspberry. Fuente: Elaboración propia.

Aparecerá una pantalla como la de la figura 129, en el que se introducirá el VI realizado a la Raspberry.

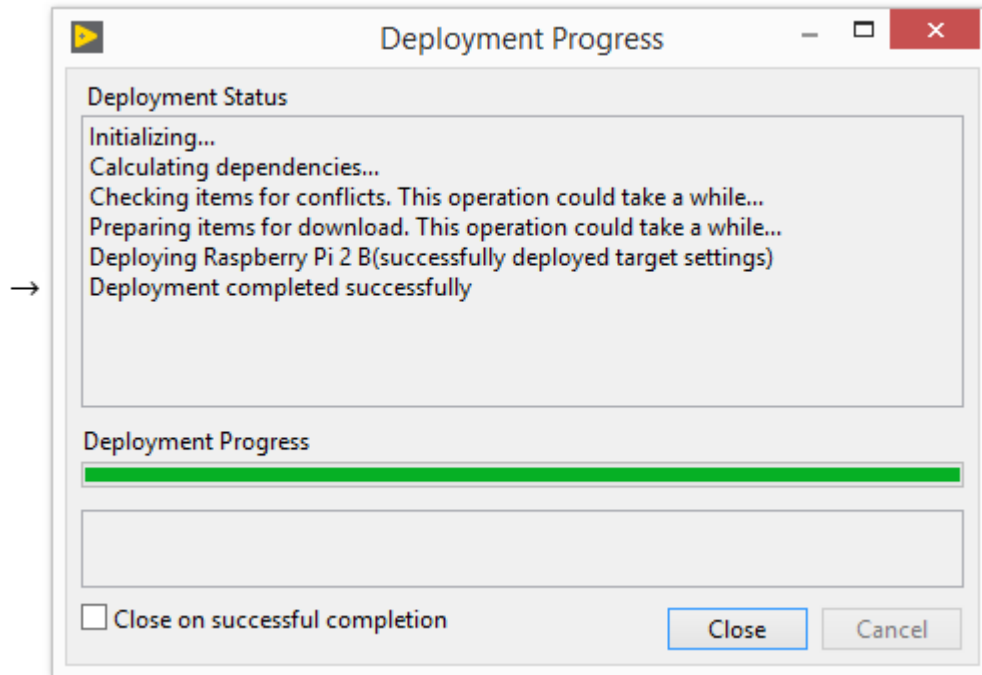


Figura 129. Introducción del programa en la tarjeta Raspberry. Fuente: Elaboración propia.

Una vez se haya desplegado correctamente el programa, en la ventana del proyecto se verá que el apartado de Raspberry Pi se ha puesto en verde, como en la figura 130.

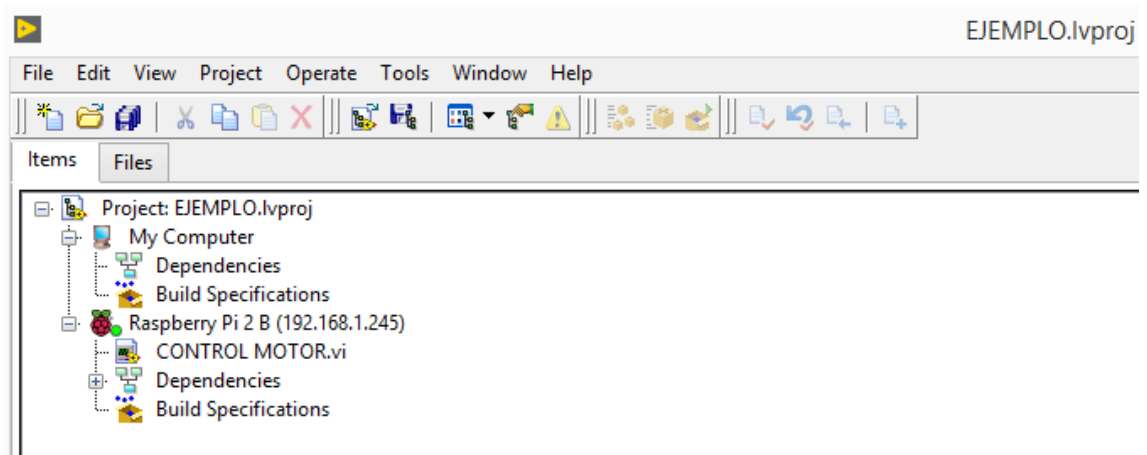


Figura 130. Despliegue satisfactorio del programa en la tarjeta Raspberry. Fuente: Elaboración propia.

En el programa deberán haberse incluido unos bloques especiales para interconectar LabVIEW con la tarjeta Raspberry. Estos bloques se encuentran en el diagrama de bloques, pulsando Botón derecho → MakerHub → LINX.

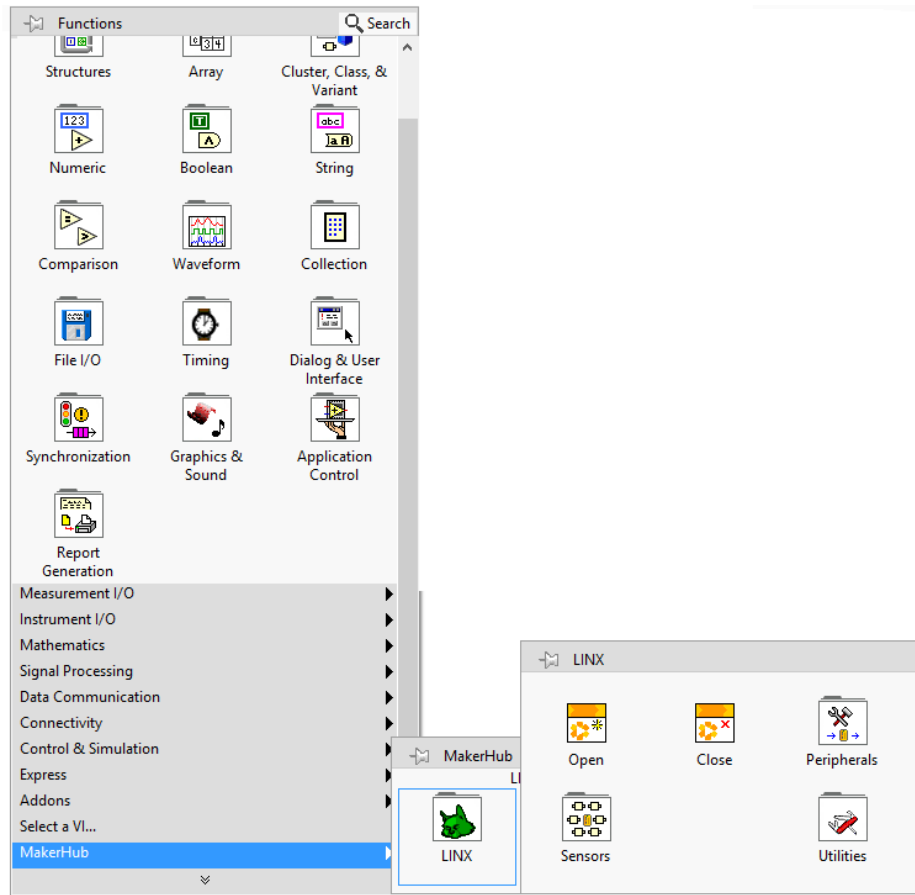


Figura 131. Bloques disponibles para la interconexión entre LabVIEW y la Raspberry. Fuente: Elaboración propia.

Hay que definir la entrada y la salida de la conexión con el dispositivo remoto. Para ello se coloca el bloque de Open y se elige la opción Local I/O y el bloque Close.

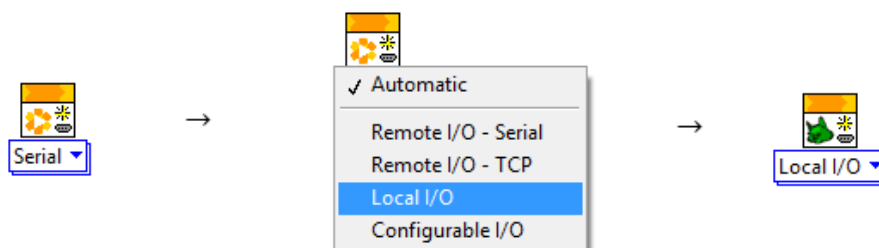


Figura 132. Colocación de la entrada. Fuente: Elaboración propia.

A continuación, se utilizarán el bloque **Digital Write** y el bloque **Digital Read**, que se encuentran dentro de Peripherals → Digital → Write/Read

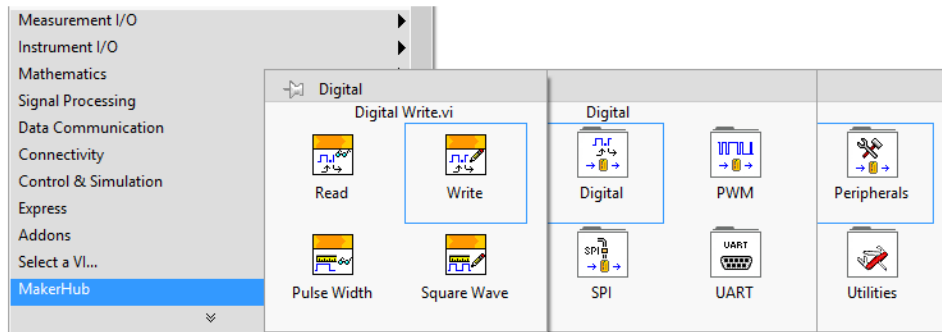


Figura 133. Colocación del bloque *Digital Write*. Fuente: Elaboración propia.

Cada bloque tiene una serie de conexiones en la entrada y en la salida, las cuales no son necesarias unir todas.

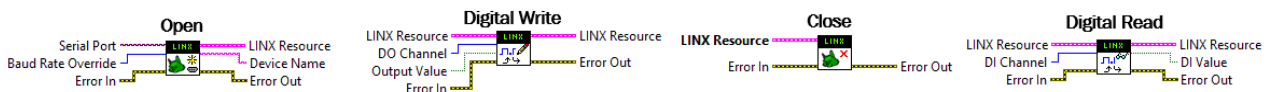


Figura 134. Conexiones disponibles en cada bloque. Elaboración propia.

La conexión más importante se encuentra en el bloque **Digital Write**, al cual se añadirán los pines de la Raspberry encargados de mandar el pulso correspondiente al motor para hacerlo girar.

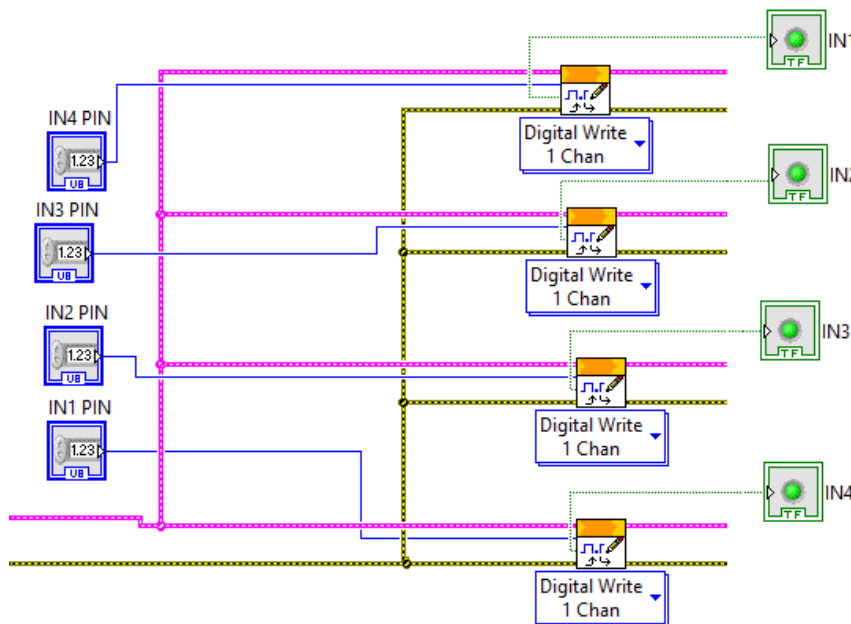


Figura 135. Conexión a realizar con el bloque *Digital Write*. Fuente: Elaboración propia.

Para controlar las entradas con pulsadores sin necesidad de utilizar el PC, se utilizarán los bloques **Digital Read**, y se asignará un pin a cada una de las posibles entradas. Además, como se mostrará posteriormente, será necesario crear un ejecutable.

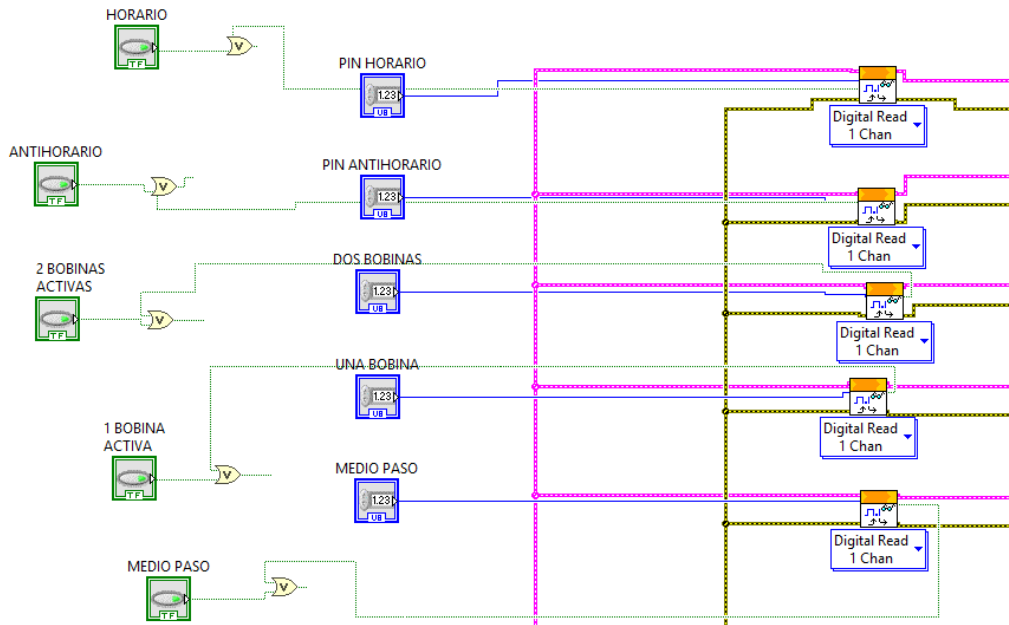


Figura 136. Conexión a realizar con el bloque Digital Read. Fuente: Elaboración propia.

En el panel frontal se elegirán los pines de la Raspberry. Para ello, hay que ver el propósito de cada pin de la Raspberry (como se muestra en la figura 79). Se elegirán los pines GPIO que se muestran en la siguiente figura.

Se colocarán esos valores por defecto al iniciar el programa y así no depender el PC para introducirlos.

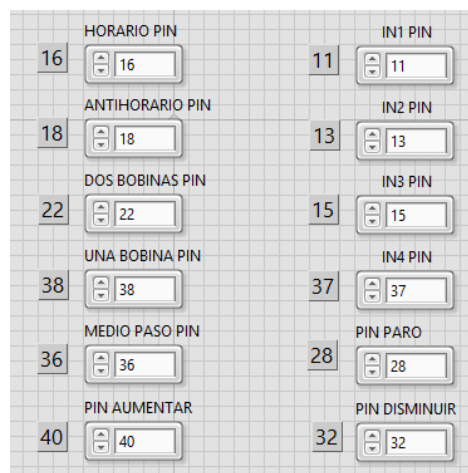


Figura 137. Pines a utilizar de la Raspberry. Fuente: Elaboración propia.

La interconexión entre el bloque Open, Close y Digital Read y Write quedaría de la siguiente manera:

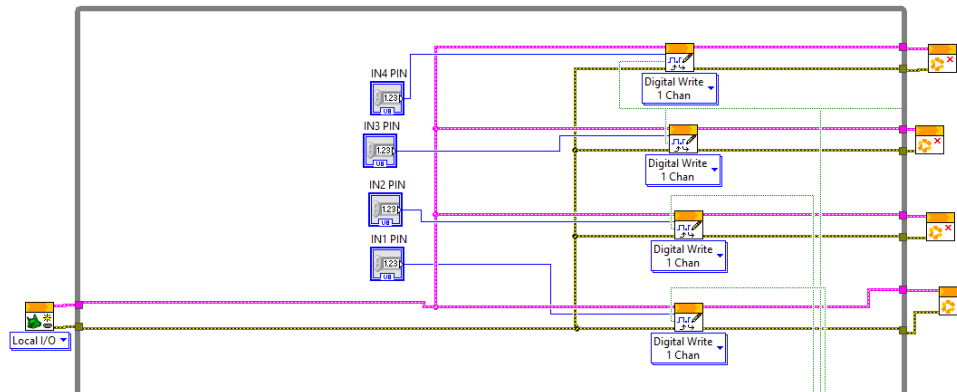


Figura 138. Conexión con la tarjeta Raspberry. Fuente: Elaboración propia.

Una vez terminadas las conexiones con el programa LabVIEW, se realizarán físicamente como se muestra en la figura 139, en el que se observa la conexión con el motor mediante el driver para la energización de las bobinas.

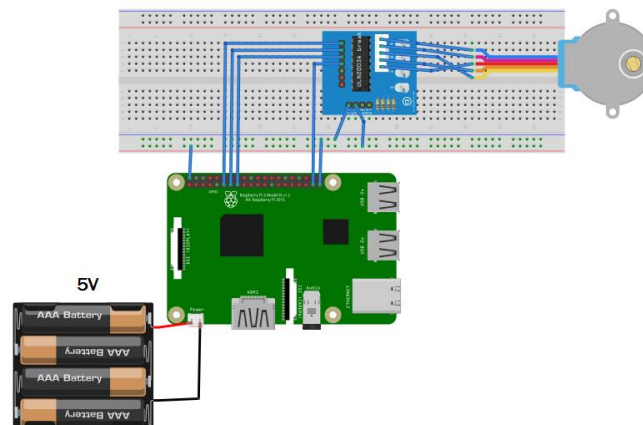


Figura 139. Esquema realizado. Fuente: Elaboración propia.

Se conecta la fuente de alimentación de 5V a la Raspberry.

Se utiliza el pin 2 para +5V y el pin 39 para 0V (GND). Se conectan a la protoboard como corresponde, el '+' con el pin 2 y el '-' con el pin 39.

Los pines 11,13,15 y 37 se unirán a través de la protoboard al driver del motor correspondiente con las entradas IN1, IN2, IN3 e IN4, respectivamente.

El motor se conecta directamente al driver con la secuencia de colores correcta.

El driver necesita alimentación también, por lo que se unirá al '+' y al '-' de la protoboard que está siendo alimentada por la Raspberry.

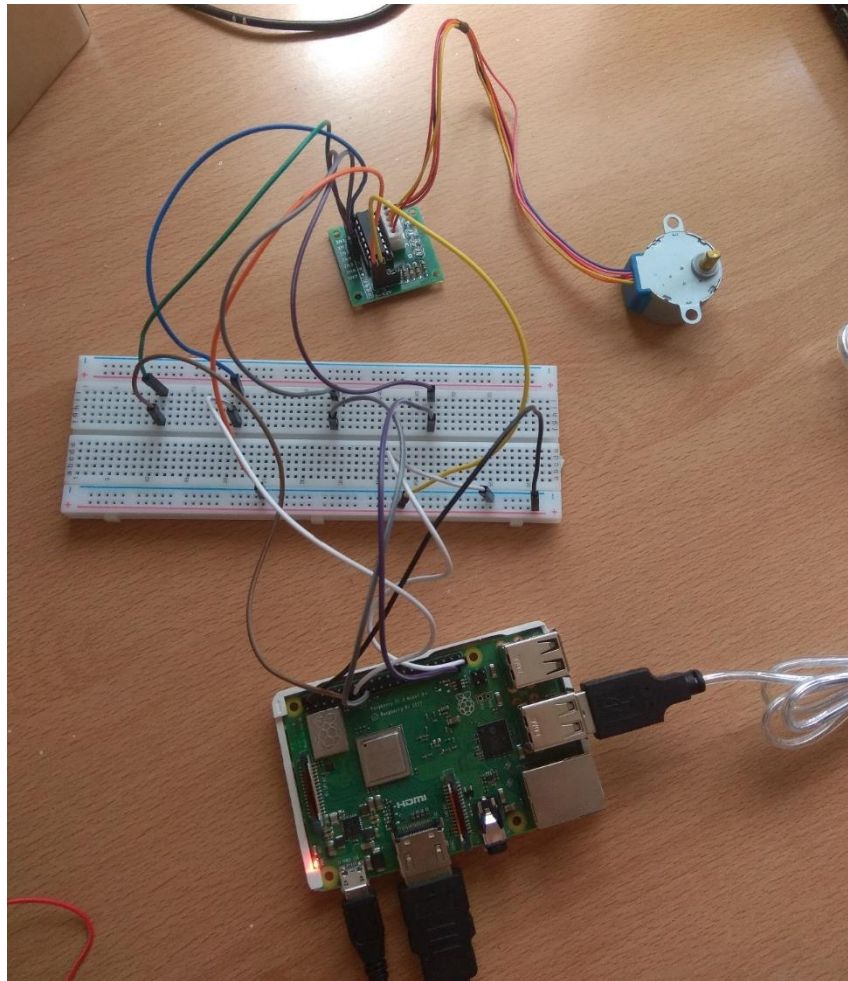


Figura 140. Montaje real. Fuente: Elaboración propia.

Una vez hecho esto, ya se puede correr el programa y comprobar todas las secuencias posibles para el control del motor mediante LabVIEW y mediante pulsadores reales.

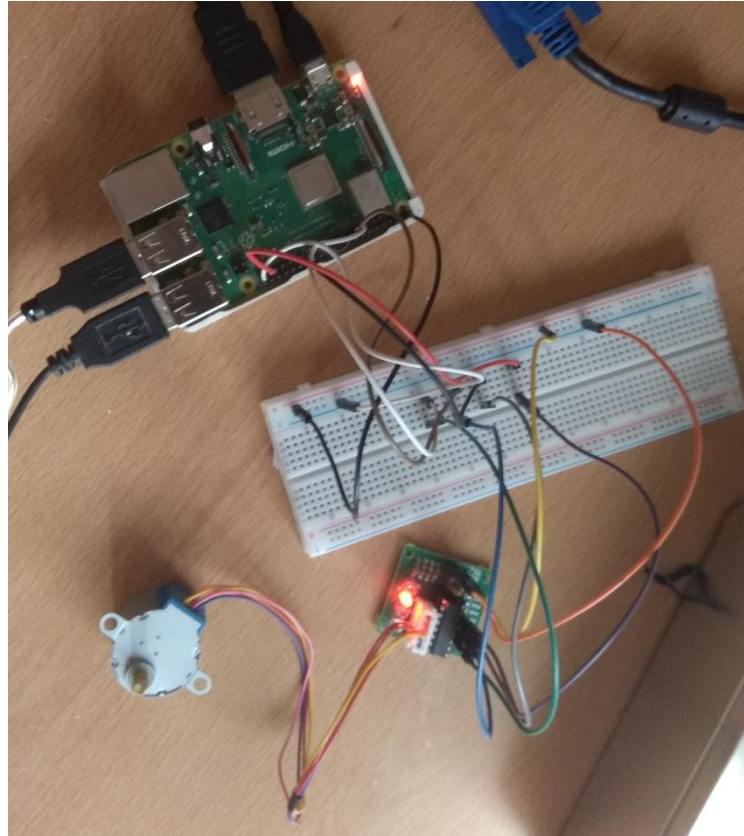


Figura 141. Motor girando a pasos completos con una bobina activa y sentido horario. Fuente: Elaboración propia.

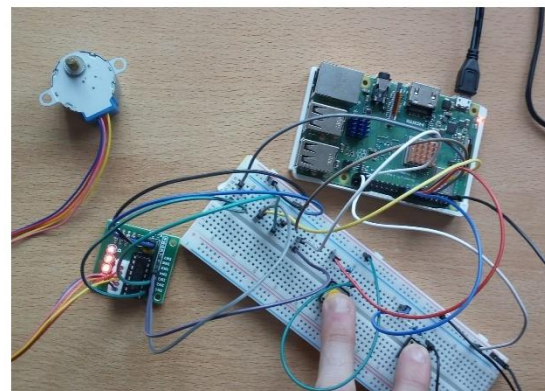
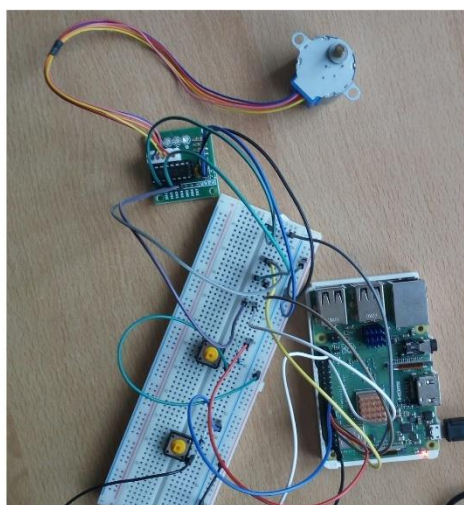


Figura 142. Motor girando a pasos completos con dos bobinas activas y sentido antihorario mediante dos pulsadores. Fuente: Elaboración propia.

Para mostrar las capacidades que tiene el programa, se va a realizar el **control remoto** del motor mediante una interface WEB, en el que será necesario utilizar LabVIEW NXG. Toda la información que aparece a continuación se ha tomado de [50].

Para el control remoto vía Web se utilizará la función **Web Service**. Botón derecho del ratón en el dispositivo → New → Web Service

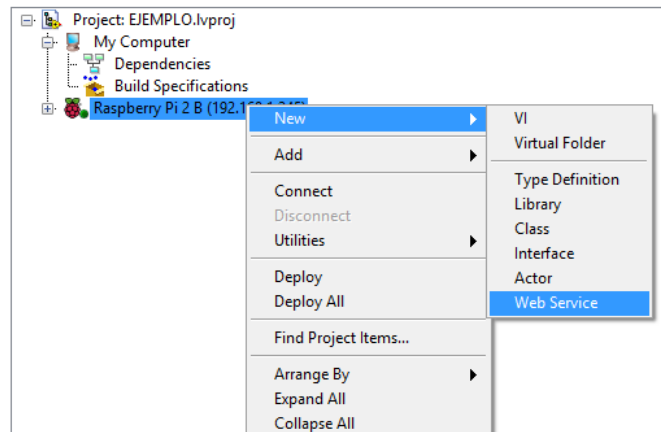


Figura 143. Creación del Web Service. Fuente: Elaboración propia.

En el apartado **Web Resources**, se añadirán las funciones de **Lectura** y **Escritura**. Botón derecho en Web Resources → New VI.

El VI de lectura tendrá la siguiente forma:

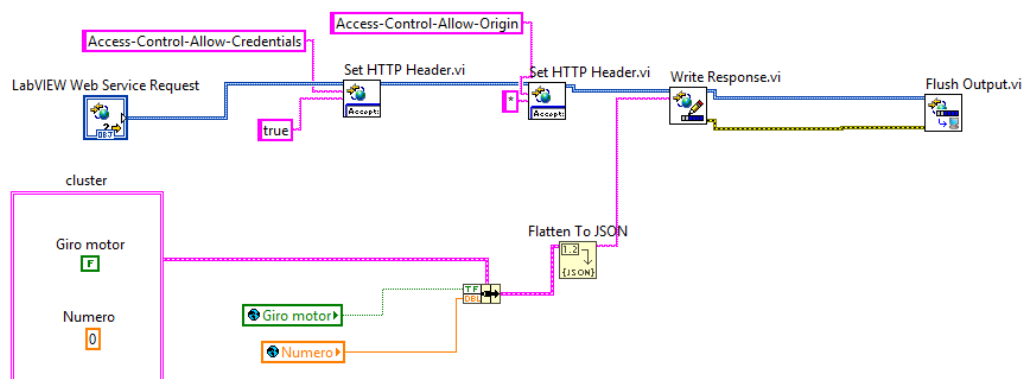


Figura 144. VI de Lectura. Fuente: Elaboración propia.

La variable giro motor indicará si está el motor en movimiento. Además, se ha añadido un control (Número) como ejemplo de envío de datos numéricos al control remoto.

El VI de escritura tendrá la siguiente forma:

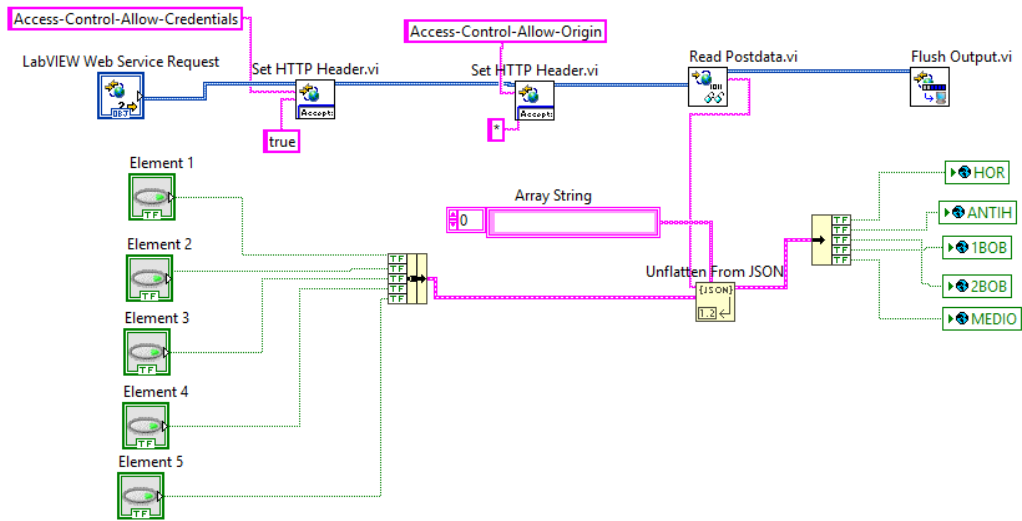


Figura 145. VI de Escritura. Fuente: Elaboración propia.

Para el envío de datos a la pantalla remota se utilizarán variables globales. Al tratarse de datos diferentes se agrupan en un clúster y se les da el formato JSON.

A continuación, se cambia a POST el método de Escritura.

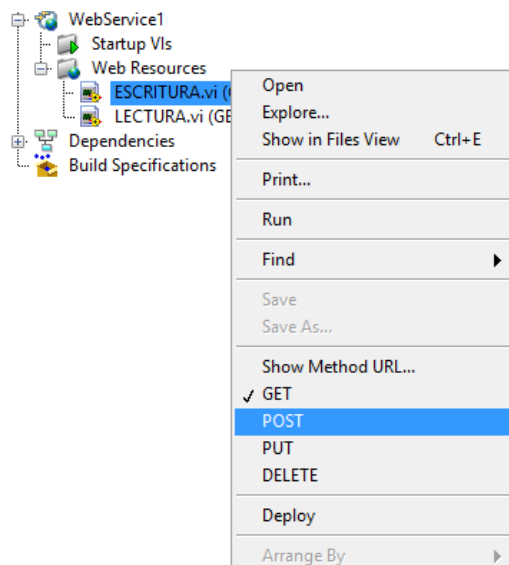


Figura 146. POST en el método de Escritura. Fuente: Elaboración propia.

Ahora se modifican las propiedades del Web Service como se muestra en la figura 147.

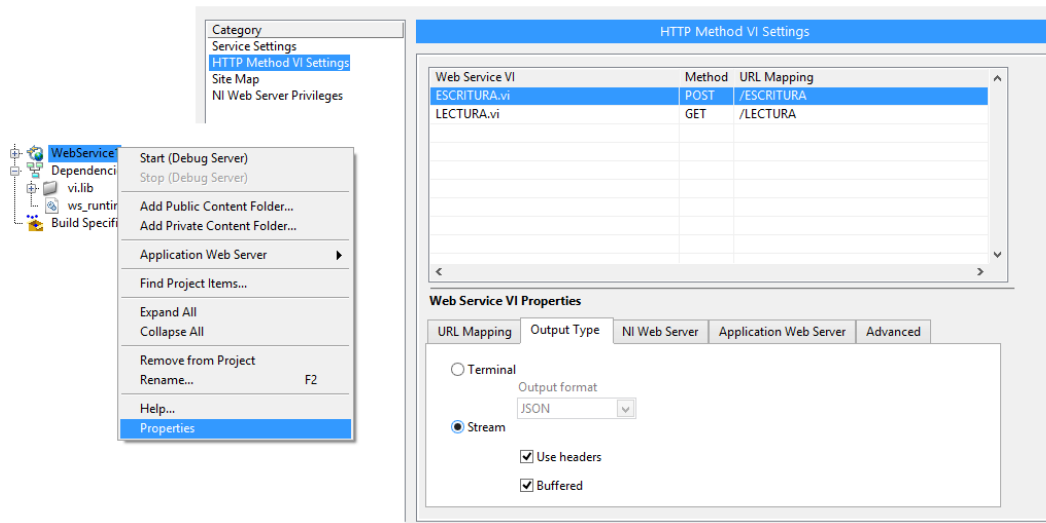


Figura 147. Modificación propiedades del Web Service. Fuente: Elaboración propia.

Se realiza el mismo cambio en el VI de Lectura.

Para comprobar el funcionamiento del Web Service, se ejecuta el VI principal y se inicia el Web Service pulsando Start.

Después, en el navegador se escribe la siguiente dirección, utilizando la IP del dispositivo: `http://xxx.xxx.xxx.xxx:8001/Webservice1/Lectura`

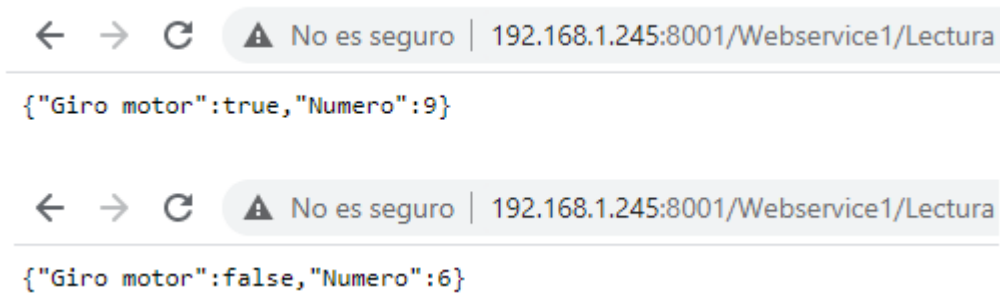


Figura 148. Prueba de funcionamiento del Web Service. Fuente: Elaboración propia.

Se observa que cuando el programa está ejecutándose escogiendo el modo de giro y los grados a girar, la variable Giro motor devuelve TRUE. En caso de que no esté girando, devuelve un FALSE. La variable Número también se actualiza al modificarla.

Para el control remoto del dispositivo se utilizará LabVIEW NXG en la creación de la interface WEB. Se pulsa en **Web Application Project** y se le da el nombre del proyecto.

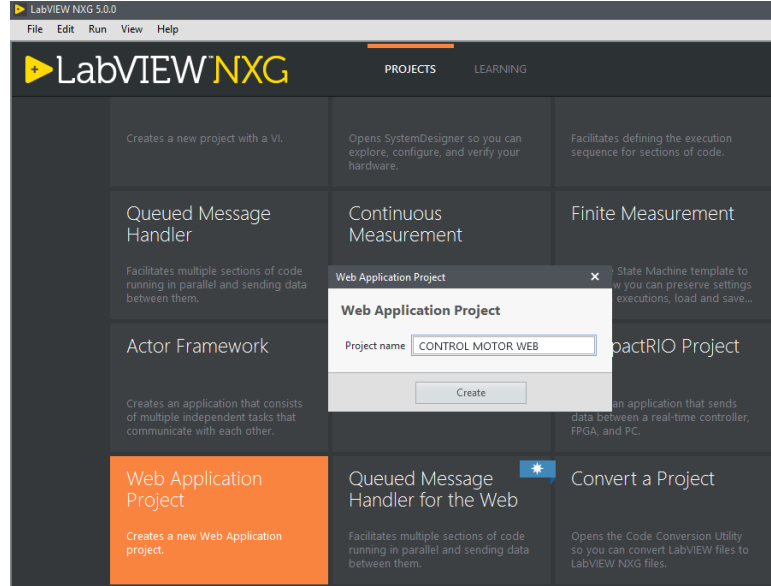


Figura 149. Creación Web Application Project. Fuente: Elaboración propia.

En la pantalla **index.giweb** se colocan controles e indicadores. Seleccionando cada control o indicador, aparece en la parte derecha un menú donde configurar los distintos parámetros. El panel frontal quedaría de la siguiente manera:

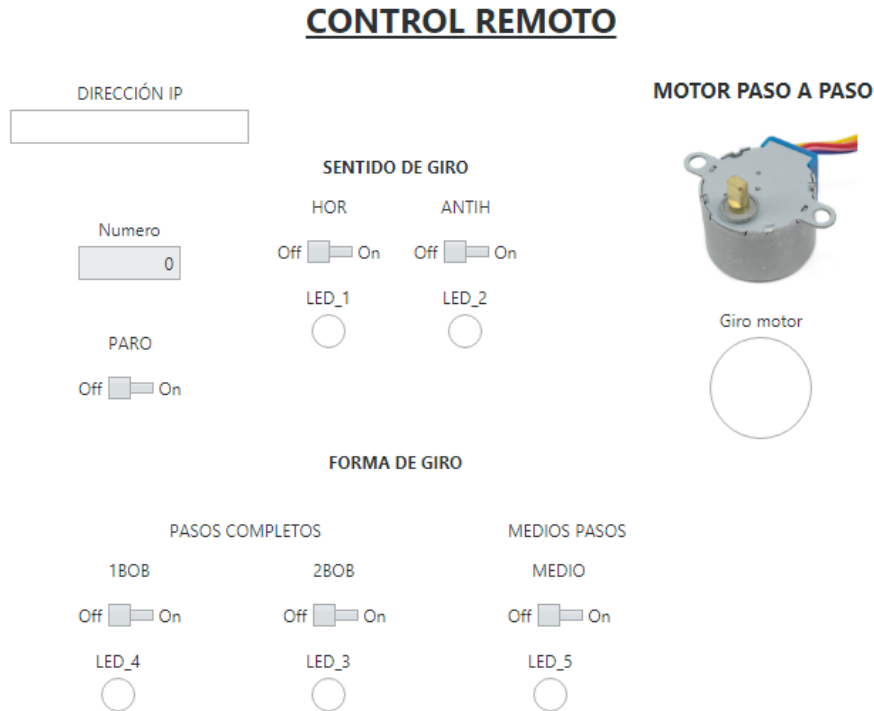


Figura 150. Panel frontal NXG. Fuente: Elaboración propia.

El diagrama de bloques quedaría de la siguiente manera:

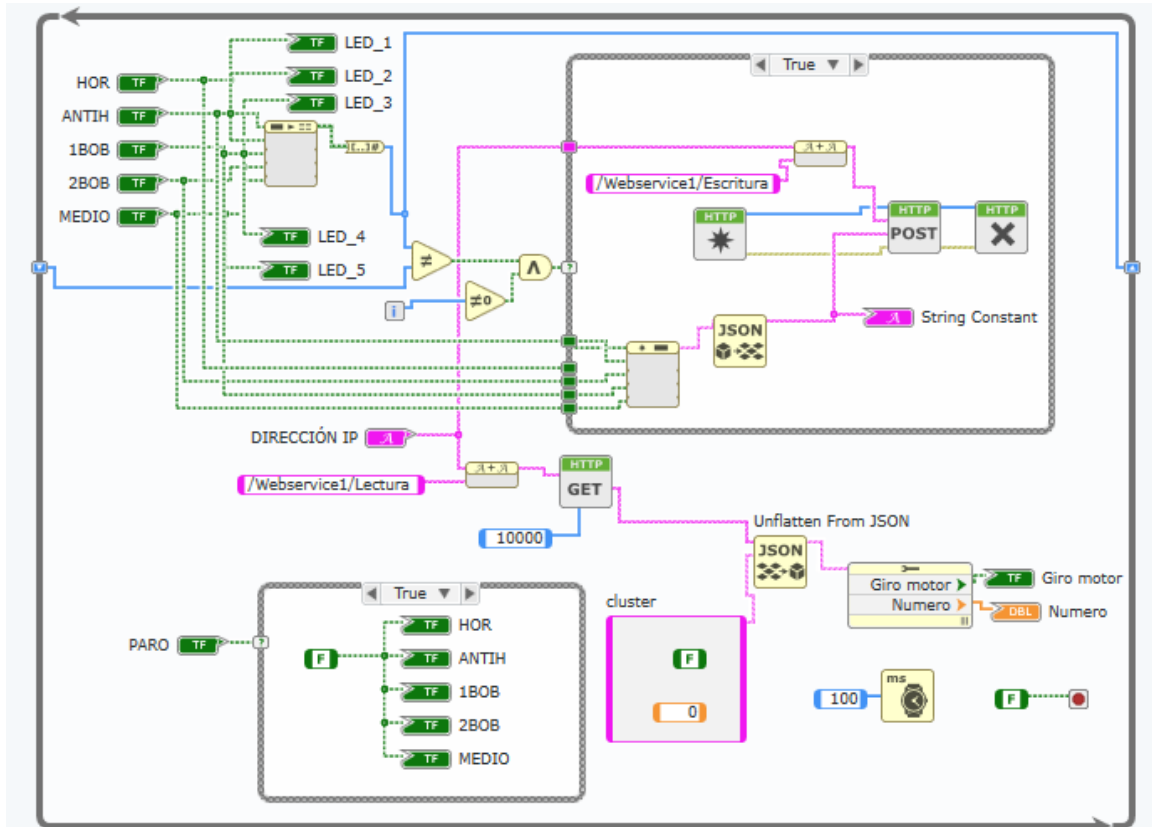


Figura 151. Diagrama de bloques NXG. Fuente: Elaboración propia.

El programa realiza la lectura al detectar cambios en los controles de entrada.

Para identificar desde qué punto se realiza el control, si es desde LabVIEW, utilizando pulsadores físicos o mediante control remoto, se han colocado indicadores en el panel frontal del programa principal.



Figura 152. Indicadores del lugar donde se realiza el control. Fuente: Elaboración propia.

Se comprueba el funcionamiento del programa. Se ejecuta el programa principal, se inicia el Web Service y, por último, index.giweb.

CONTROL REMOTO

DIRECCIÓN IP
http://192.168.1.245:8001

Numero
9

PARO
Off On

SENTIDO DE GIRO
HOR: Off On
ANTI H: Off On
LED_1:
LED_2:

MOTOR PASO A PASO
Giro motor

FORMA DE GIRO
PASOS COMPLETOS: 1BOB (Off On), 2BOB (Off On)
MEDIOS PASOS: MEDIO (Off On)
LED_4:
LED_3:
LED_5:

Figura 153. Ejecución del control remoto. Fuente: Elaboración propia.

Se elige la forma de control, en este caso sentido horario, girando a medios pasos. Se activa la variable de Giro motor y se observa en el programa principal de LabVIEW que el motor está girando y que las variables correspondientes son las que están activas.

LEDS PULSADORES FÍSICOS

LEDS PULSADORES REMOTOS

HORARIO
ANTI HORARIO
PASO COMPLETO 2 BOBINAS ACTIVAS
PASO COMPLETO 1 BOBINA ACTIVA
MEDIO PASO

LED 1
LED 2
LED 3
LED 4
LED 5

HORARIO
ANTI HORARIO
PASO COMPLETO 2 BOBINAS ACTIVAS
PASO COMPLETO 1 BOBINA ACTIVA
MEDIO PASO

LED_1
LED_2
LED_3
LED_4
LED_5

Figura 154. Control remoto activo. Fuente: Elaboración propia.

Finalmente se crea la página web que servirá para el control remoto del programa de la Raspberry.

Para ello, en **SystemDesigner**, en el apartado **WebApp.gcomp**, se selecciona **Build** hasta que la generación de la WEB esté completa y sin fallos.

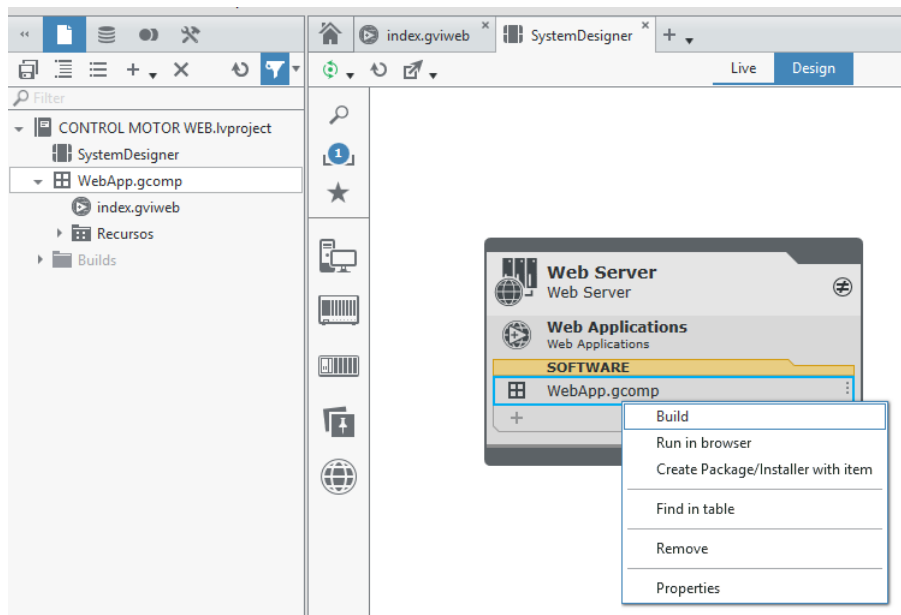


Figura 155. Creación página web para el control. Fuente: Elaboración propia.

La página web, con todos sus complementos, se habrá creado en el directorio donde se encuentre el proyecto.

Nombre	Fecha de modifica...	Tipo	Tamaño
ni-webvi-resource-v0	26/09/2020 19:54	Carpeta de archivos	
Recursos	26/09/2020 19:54	Carpeta de archivos	
index	26/09/2020 19:54	Chrome HTML Do...	24 KB
index.via	26/09/2020 19:54	Documento de tex...	30 KB
version.json	26/09/2020 19:54	Archivo JSON	1 KB

Figura 156. Archivos correspondientes creados correctamente. Fuente: Elaboración propia.

Se abre la página web y se comprueba que funciona correctamente.

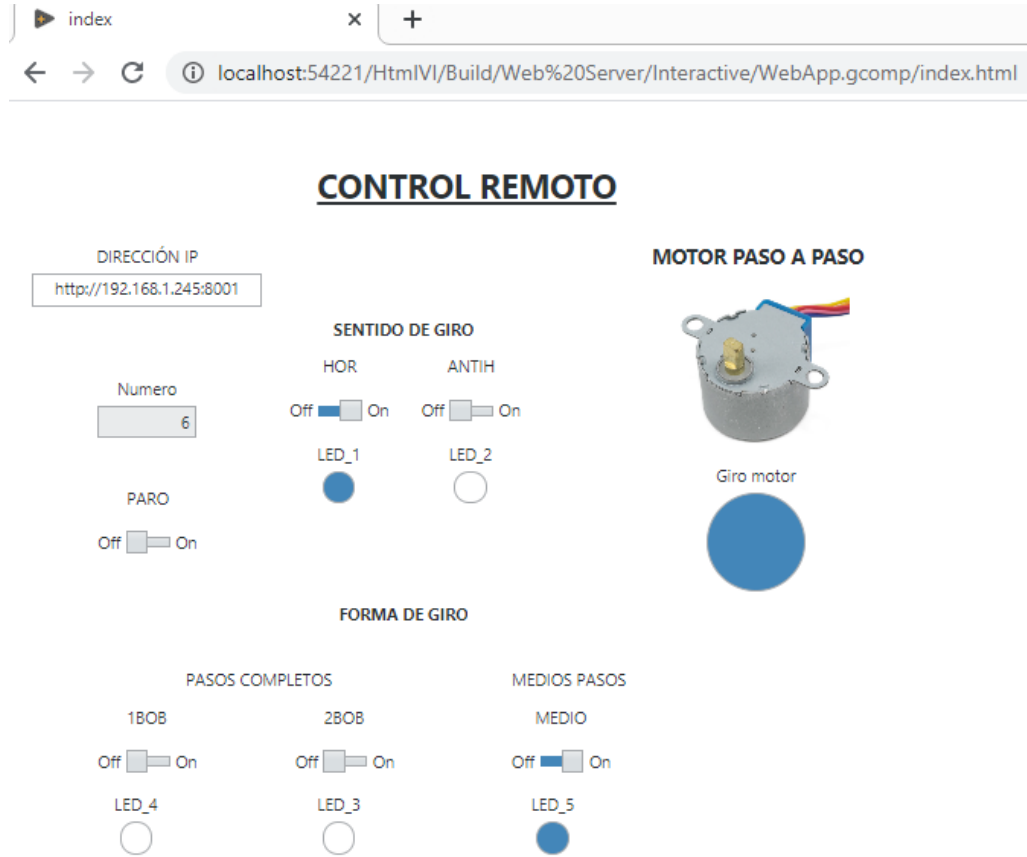


Figura 157. Página Web del control remoto. Fuente: Elaboración propia.

Para controlar el motor sin usar el PC, será necesario crear un **ejecutable**. En el apartado del proyecto se creará una aplicación en tiempo real. Para ello, botón derecho en Build Specifications en el apartado de Raspberry Pi → New → Real-Time Application.

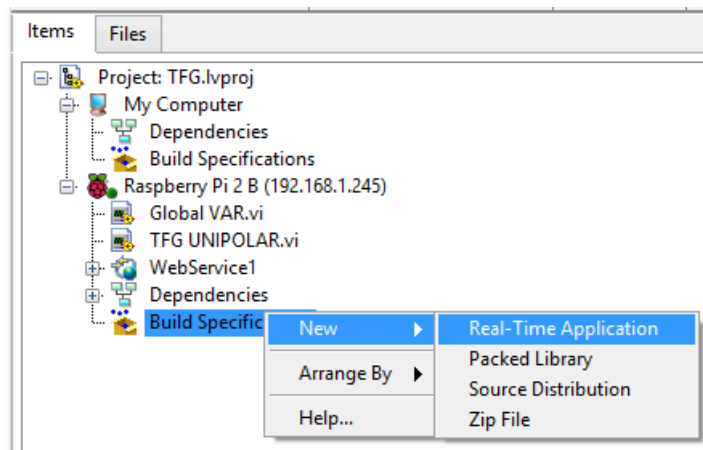


Figura 158. Creación de aplicación en tiempo real. Fuente: Elaboración propia.

En la categoría **Information** se le da el nombre que se desea.

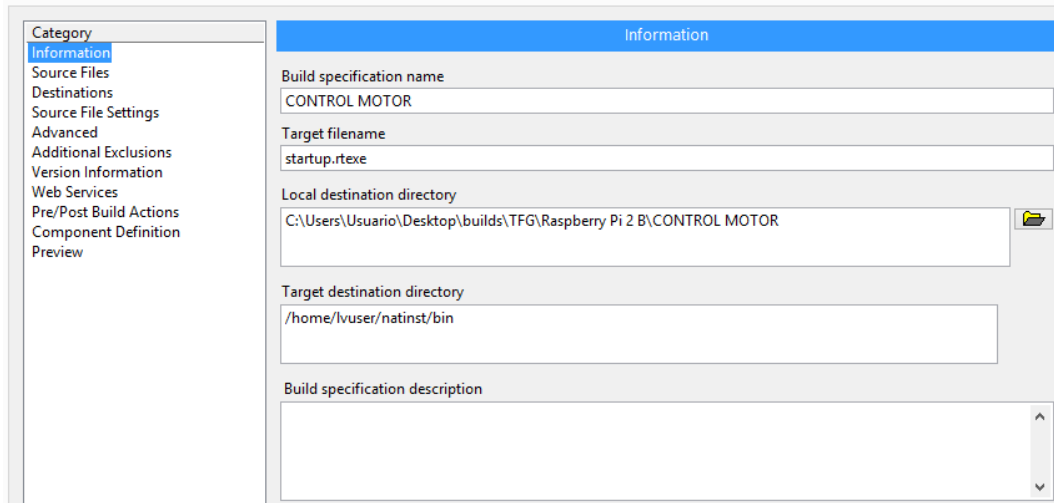


Figura 159. Modificación de la categoría Information. Fuente: Elaboración propia.

En la categoría **Source Files**, se añade el VI a la ventana de Startup Vis.

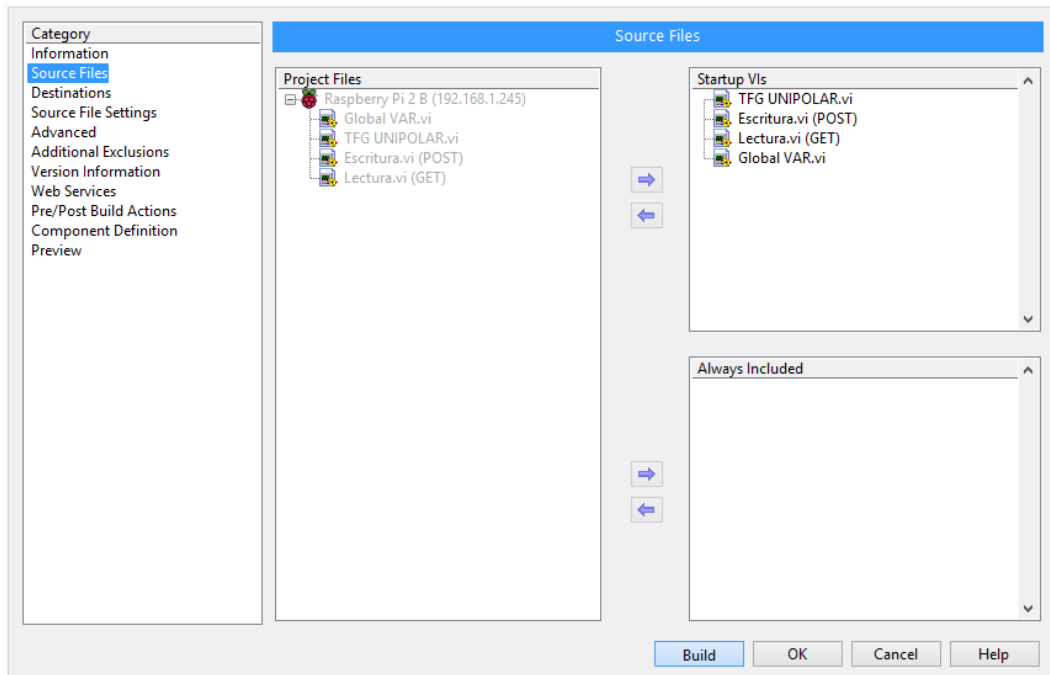


Figura 160. Modificación de la categoría Information. Fuente: Elaboración propia.

Finalmente, se construye pulsando en Build.

Una vez creado, en el proyecto aparecerá un nuevo apartado como se muestra en la figura 161. Botón derecho → Run as Startup.

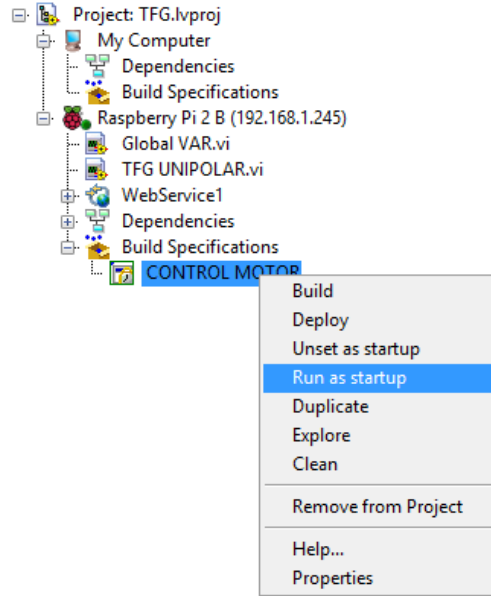


Figura 161. Ejecutable creado en el proyecto. Fuente: Elaboración propia.

Hecho esto, el programa pedirá reiniciar la Raspberry y se podrá utilizar sin la necesidad de utilizar el ordenador. Para comprobarlo, se desconecta la fuente de alimentación de la Raspberry y, al volver a conectarla, se observa que el programa instalado en la Raspberry funciona correctamente.

4. CONCLUSIONES

Con la realización de este trabajo se pueden extraer varias conclusiones:

Sobre los motores paso a paso.

- Los motores paso a paso son motores muy útiles para realizar tareas específicas que un motor convencional no es capaz de realizar, como aplicaciones que requieran alta precisión.
- Pueden ser de tres tipos, de imanes permanentes, de reluctancia variable e híbridos, según su estructura (rotor, polos en el estator...). También se pueden clasificar atendiendo a la disposición de los bobinados, pueden ser unipolares o bipolares, que será determinante conocer para realizar el control del motor, ya que es diferente la forma de alimentar las bobinas y, por tanto, de diseñar el sistema de control.
- Las características más importantes de los motores paso a paso son el ángulo de paso, que es el ángulo que gira el rotor por cada pulso que recibe el motor, y los pasos por vuelta, que dependerán según sea la forma de control.

Hay tres formas de realizar el control de los motores paso a paso:

Pasos completos: el rotor gira un ángulo de paso por cada paso que da. Los pasos completos se pueden realizar alimentando una bobina o dos bobinas simultáneamente por paso.

Medios pasos: el rotor gira la mitad del ángulo de paso por cada paso que da. Esto se consigue realizando una combinación adecuada de ambos métodos del control a pasos completos.

Micropasos: consiste en alimentar al mismo tiempo varias fases a la vez con corrientes medias distintas, por lo que el rotor girará fracciones del ángulo de paso dependiendo de la magnitud de las mismas, pudiendo llegar, por ejemplo, a $1/32$ veces el ángulo de paso.

- Los motores paso a paso se pueden utilizar como motores convencionales, girando de forma continua y la curva par-velocidad para el mismo motor será diferente, tal y como se observa en la figura 35.
- Es importante tener en cuenta ciertos efectos que se producen en los motores paso a paso, aunque sean poco tenidos en cuenta. Estos son la oscilación del rotor en una posición de equilibrio, la constante de tiempo eléctrica, que influirá en la corriente total que circula por el devanado y puede influir en el par del motor y el error de posición estática, que dependerá del ángulo de paso y del par de carga. Para todos estos problemas se comentaron soluciones en el

apartado 2.1.2 de este trabajo, tales como la amortiguación en la oscilación del rotor mediante métodos mecánicos, eléctricos y electrónicos, la disminución de la constante de tiempo eléctrica mediante un circuito auxiliar y aumentando la precisión mediante un control en lazo cerrado, en el que midiendo en tiempo real la posición del rotor, se actúa sobre la alimentación del estator.

Sobre Raspberry Pi.

La tarjeta Raspberry Pi es un ordenador de placa reducida capaz de realizar tareas específicas por un bajo coste y con un bajo consumo, lo cual es ideal, como se ha mostrado en este trabajo, para realizar el control de un motor paso a paso y para cualquier aplicación que requiera la implementación de un sistema de control, tal y como se describe en el apartado 5 de propuestas de trabajos.

Sobre LabVIEW.

- LabVIEW es un software que permite realizar tareas como la adquisición de datos, creación de sistemas de control embebidos y aplicaciones que proporciona visualización y manejo de gráficas con datos dinámicos y procesamiento digital de señales en tiempo real.

- Es un lenguaje de programación muy completo, en el que se ha demostrado su capacidad de creación de programas de forma gráfica e intuitiva, que además funciona perfectamente interactuando con la tarjeta Raspberry Pi mediante librerías y bloques en el propio software, y sin conexiones físicas complejas, únicamente de forma inalámbrica.

Sobre las formas de control del motor.

- En cuanto a la creación del software, se ha mostrado la facilidad con la que se puede controlar un motor paso a paso mediante el uso de bloques, bucles y operaciones lógicas en LabVIEW. Esto se visualiza en la forma de elección en la forma de control y la secuencia de alimentación del motor en el diagrama de bloques, explicado en el apartado 3.3, así como la obtención de todos los parámetros necesarios en el control y visualizarlos en tiempo real.

- En cuanto al control con entradas físicas, una vez configurada la tarjeta a utilizar, se consigue controlar el motor mediante el uso de pulsadores físicos alimentando determinados pines en la tarjeta Raspberry. Estos pines se eligen en el panel frontal del programa principal en LabVIEW, que corresponderán a las diferentes formas de control (horario o antihorario y pasos completos o medios pasos). Para ello hay que fijarse en el PINOUT de la tarjeta y utilizar los pines GPIO adecuados.



La Raspberry no permite entradas analógicas, que sería lo intuitivo a la hora de variar la velocidad de un motor, por lo que se puede variar utilizando pulsadores físicos conectados a los pines que activen los bloques de aumentar y disminuir velocidad, eligiéndolos en el panel frontal.

- En cuanto al control remoto vía Web, se explica la utilización de la función Web Service en LabVIEW, en la cual se crean de dos VIs que servirán para la lectura y escritura de datos en formato JSON. Además, se muestra el programa creado en LabVIEW NXG para la creación de la interface web y conseguir realizar el control remoto del motor paso a paso.



5. PROPUESTAS DE TRABAJOS

En este apartado se proponen trabajos relacionados que pueden ser interesantes, teniendo que ver con el uso de un microcontrolador para mostrar su potencial, el uso de LabVIEW como lenguaje de programación gráfico, y el estudio y control de motores especiales, ya sean motores paso a paso, servomotores o motores brushless. La información principal ha sido extraída de [39], [40] y [45].

1.- Control de motor brushless

Para el control de un motor brushless es necesario conocer en todo momento la posición del rotor para que la secuencia de alimentación del estator sea la correcta en cualquier instante, por lo que deberá realizarse un control en lazo cerrado.

Para conocer la posición relativa de rotor y estator, se utilizaría un sensor Hall, que indicará qué par de bobinas debe ser alimentado.

Para controlar la velocidad del motor brushless se utiliza la modulación PWM, modulación por ancho de pulsos. Consiste básicamente en apagar y encender rápidamente el interruptor que permite el paso de energía.

Al ser una conmutación rápida, el equipo observa un valor medio de señal. Si en cada período se alimenta con una tensión V y el ciclo de trabajo es el 50%, el motor verá que es alimentado con una tensión de $V/2$ y girará a una velocidad proporcional a esa tensión.

El microcontrolador será el encargado de enviar las señales a la frecuencia que se desee y, dependiendo de las características del motor, será necesario una etapa de potencia para amplificar las señales que recibe.

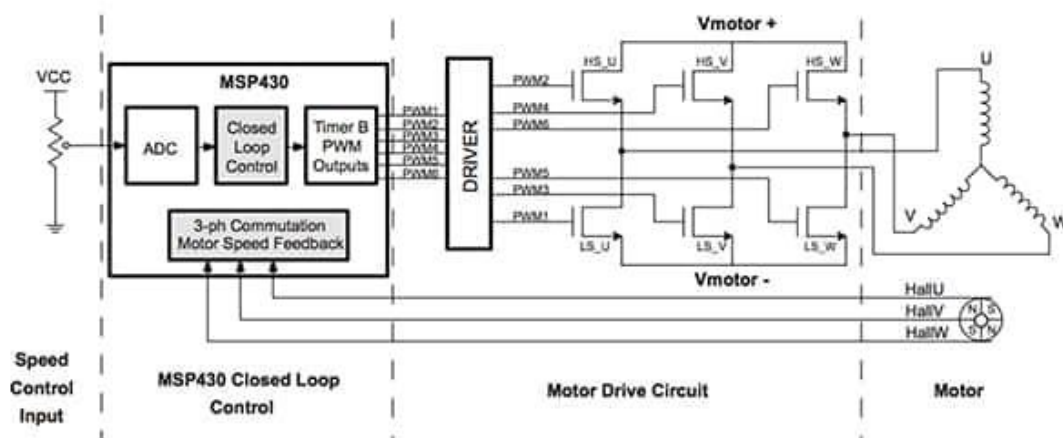


Figura 162. Esquema de control motor brushless. Fuente: [39]

2.- Control de un servomotor

Los servomotores son dispositivos actuadores con la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y de mantenerse estable en dicha posición. Está formado por un motor de corriente continua, una caja reductora y un circuito de control, lo que proporciona un alto par y precisión, pero baja velocidad de rotación en comparación con los motores convencionales, y su margen de funcionamiento suele ser de media vuelta completa.

El servomotor girará a la posición que se le indique. Para alcanzar esa posición, el circuito de control deberá de aplicar un pulso de una duración determinada.

Normalmente el ángulo de giro y la duración del pulso son las que se indican en la figura 163.

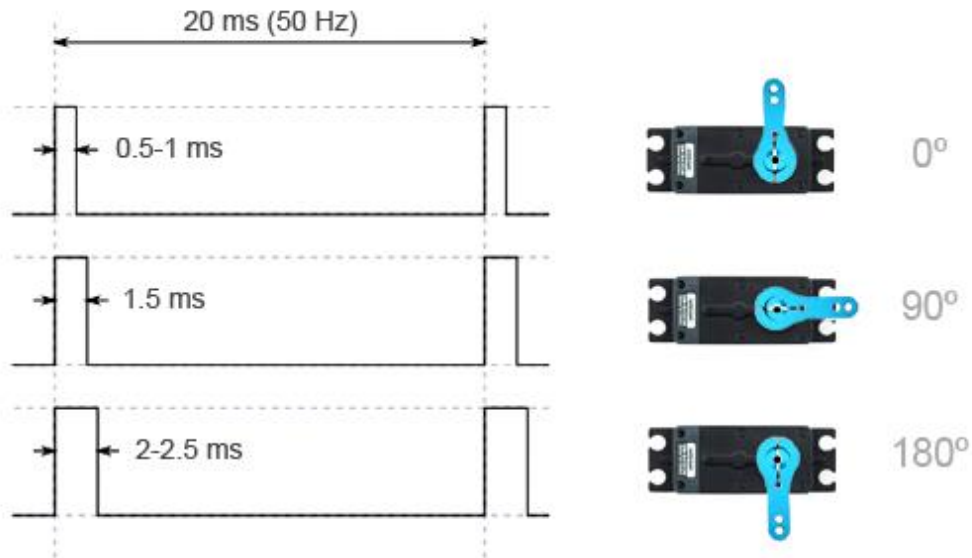


Figura 163. Funcionamiento de un servomotor convencional. Fuente: [40]

El control de un servomotor en lazo abierto es sencillo, simplemente dispone de 3 cables, dos para la alimentación y otro para la **señal de la posición deseada**.

Para un control preciso de un servomotor es necesario un control en lazo cerrado, conociendo en tiempo real la posición angular y la velocidad del rotor, ambos parámetros simultáneamente. Se pueden utilizar encoders, tacómetros o potenciómetros y comparar la respuesta real con la posición y velocidad deseada.

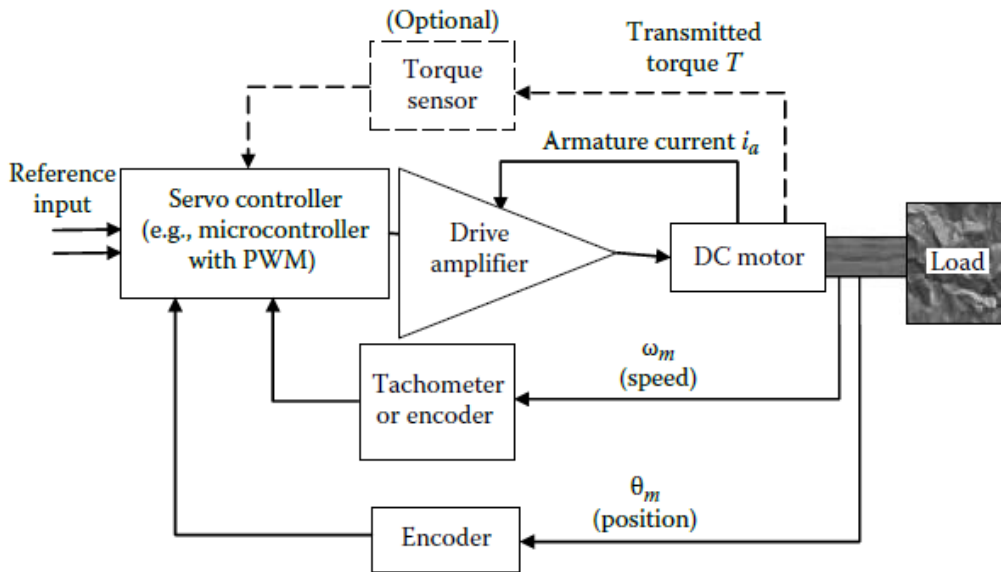


Figura 164. Control de un servomotor en lazo cerrado. Fuente: [10]

Controlando la velocidad y la posición, se podría controlar indirectamente el par. Sería necesario para aplicaciones específicas como operaciones para el conformado de metal, micromanipulación...

También existen **servomotores de rotación continua**, que pueden girar 360° como un motor convencional con las características de un servomotor.

El control es idéntico al de uno convencional, pero la señal que se envía controla la velocidad de giro y no la posición angular deseada. Pueden girar en un sentido o en otro dependiendo de la señal. En la figura 165 se hace variar la velocidad del servomotor entre parado, al 100% de velocidad en un sentido y al 100% en el sentido contrario, manteniendo cada velocidad 1.5 segundos.

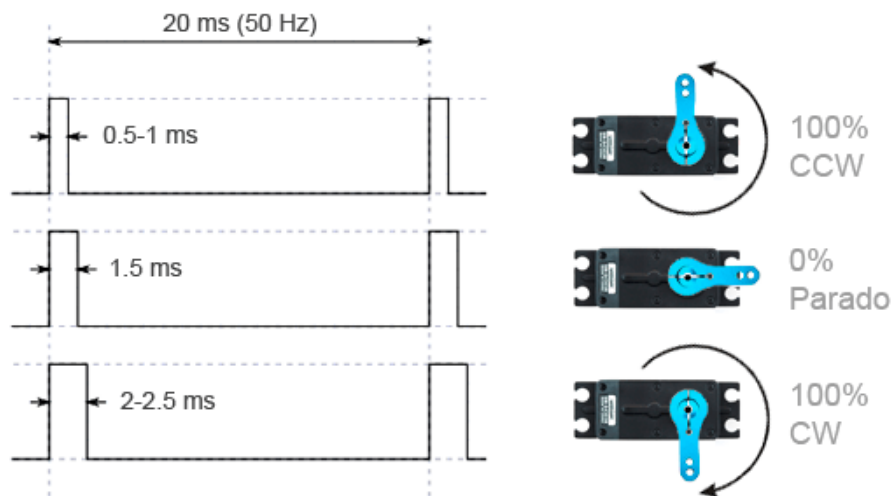


Figura 165. Funcionamiento de un servomotor de rotación continua. Fuente: [41]

3.- Control de un motor paso a paso en lazo cerrado

Se podría realizar el control de un motor paso a paso en lazo cerrado y comprobar su respuesta ante micropasos, compensando la pérdida de precisión con la medición en tiempo real de la respuesta.

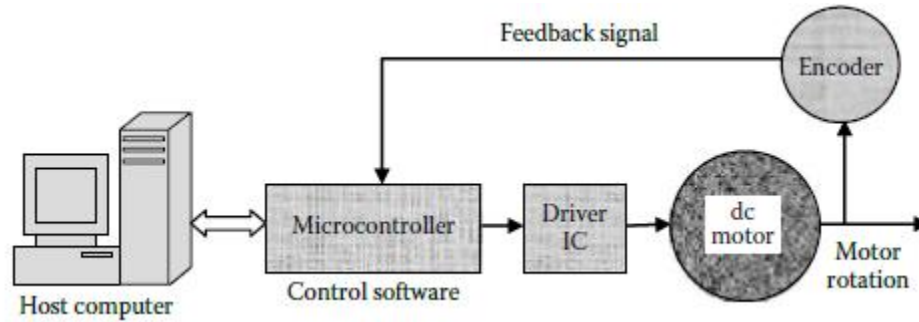


Figura 166. Control en lazo cerrado de un motor paso a paso. Fuente: [10]

Para ello es necesario un encoder rotatorio como por ejemplo el LPD3806, el cual convierte la posición del eje en señales digitales que recibirá el microcontrolador.



Figura 167. Encoder rotatorio. Fuente: [42]

Para realizar el control en micropasos es necesario un Driver como el TB6600, el cual es necesario ajustar los valores de corriente y tensión para no quemar el motor y para que pueda girar. Dependiendo de las señales de entrada del microcontrolador, el motor girará en sentido horario o antihorario realizará pasos completos, medios pasos o micropasos, desde 1/4 del paso nominal hasta 1/32.

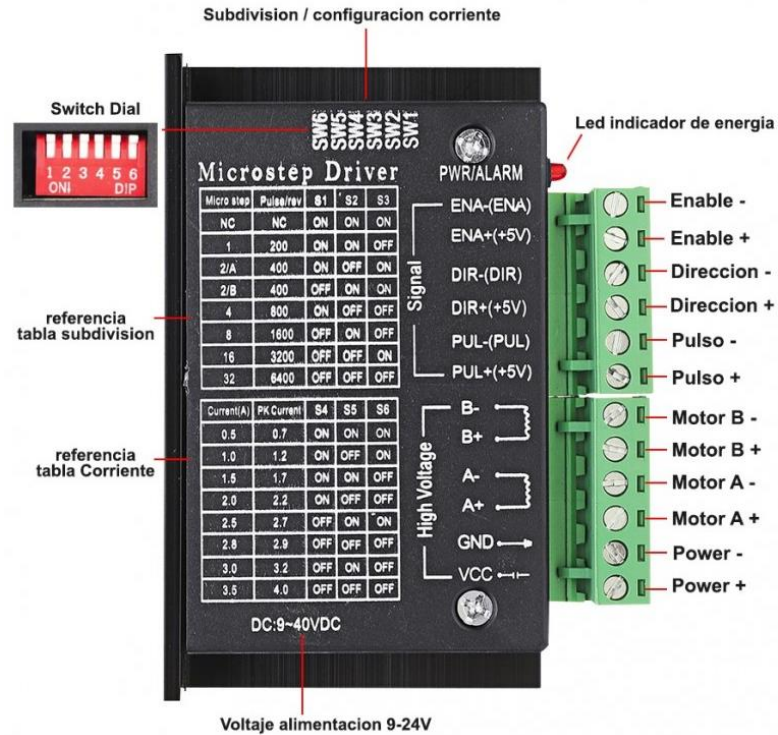


Figura 168. Driver para el control con micropasos. Fuente: [43]

Finalmente, la conexión quedaría de la siguiente manera:

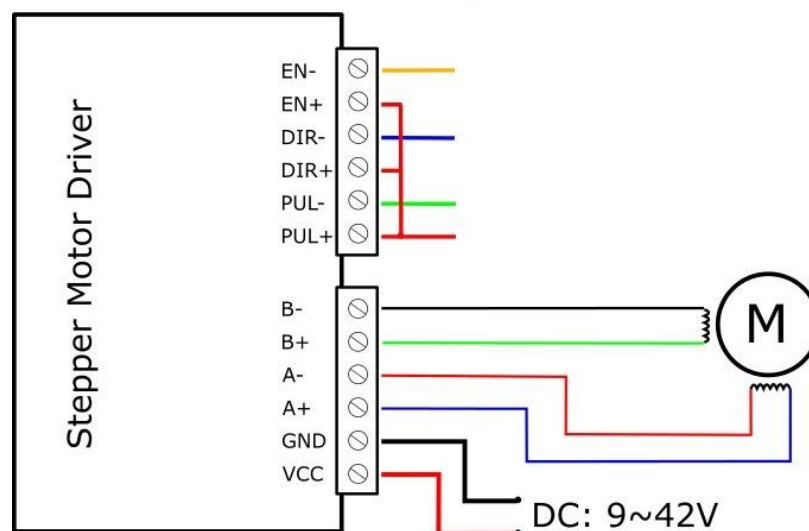


Figura 169. Esquema de conexión control con micropasos. Fuente: [44]

4.- Simulación de un motor paso a paso

Se podría realizar una simulación dinámica de un motor paso a paso gracias a las ecuaciones que describen el movimiento y el par según sea un motor de reluctancia variable o de imanes permanentes, simular un amortiguador mecánico o electrónico y comparar gráficas en las respuestas ante pasos, tal y como se explica en el apartado 2.1.2.8. de este trabajo.

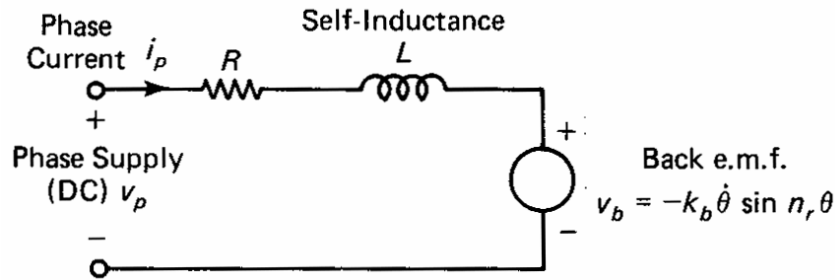


Figura 170. Circuito equivalente aproximado por fase de un motor paso a paso. Fuente: [11]

Ecuación dinámica a partir del circuito equivalente del motor paso a paso por fase:

$$v_p = R \cdot i_p + L \cdot \frac{di_p}{dt} + v_b$$

Teniendo en cuenta que:

$$v_b = -k_b \cdot \dot{\theta} \cdot \text{sen}(n_r \cdot \theta)$$

$$L = L_o + L_a \cdot \cos(n_r \cdot \theta)$$

Ecuación del par para un motor de imanes permanentes:

$$T = -k_m \cdot i_p \cdot \text{sen}(n_r \cdot \theta)$$

Ecuación del par para un motor de imanes permanentes:

$$T = -k_r \cdot i_p^2 \cdot \text{sen}(n_r \cdot \theta)$$

Ecuación mecánica del movimiento del motor:

$$T - T_L - T_b(\theta, \dot{\theta}) = J \ddot{\theta}$$

Se darían distintos valores según la potencia del motor, el momento de inercia y la constante de par y se podrían comparar las respuestas para diferentes casos (motor de gran potencia, alto par, poca inercia...).

Una vez estudiado distintos casos, se procedería a simular un amortiguador que variaría la respuesta para los mismos parámetros utilizados.

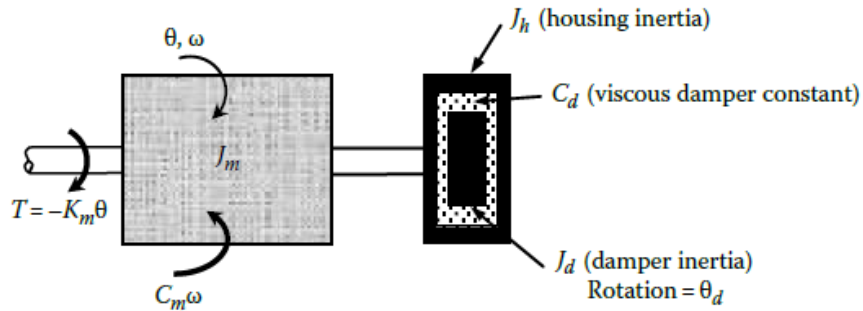


Figura 171. Motor paso a paso con un amortiguador Houdaille. Fuente: [10]

Y se utilizarían las nuevas ecuaciones de movimiento:

$$(J_m + J_h) \cdot \frac{d^2\theta}{dt^2} = -C_m \cdot \frac{d\theta}{dt} - K_m \cdot \theta - C_d \left(\frac{d\theta}{dt} - \frac{d\theta_d}{dt} \right)$$

$$J_d \cdot \frac{d^2\theta_d}{dt^2} = C_d \left(\frac{d\theta}{dt} - \frac{d\theta_d}{dt} \right)$$

5.- Diseño de automatismos con dispositivos embebidos.

Se podrían realizar automatismos eléctricos en LabVIEW y mostrar la utilidad de la tarjeta Raspberry, mediante entradas y salidas digitales con pulsadores y diodos led.

En el ejemplo se muestra el esquema de arranque de un motor asíncrono trifásico mediante dos pulsadores. Una vez accionado el pulsador de marcha (S1), se conecta el motor a la red. El motor quedará conectado después de soltar el pulsador. Para la desconexión se utilizará el pulsador (S0) de paro. La desconexión tendrá prioridad sobre la conexión en el caso de accionar los dos pulsadores simultáneamente.

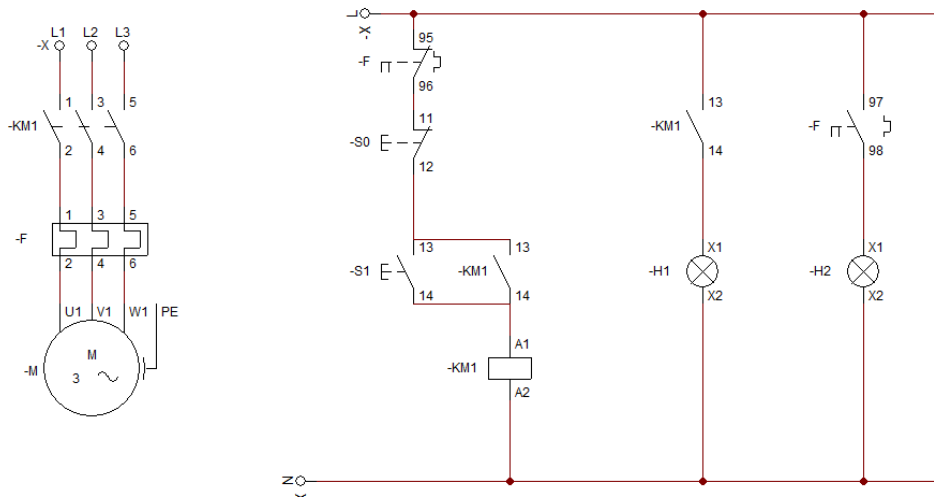


Figura 172. Ejemplo de marcha-paro de un motor asíncrono trifásico en CAdE_SIMU. Fuente: Elaboración propia.

A continuación, se realiza el esquema en LabVIEW, utilizando los pines GPIO 13 y 15 de la Raspberry para activar dos diodos leds, uno para indicar que el motor está girando (KM1) y otro en caso de que se accione F1, que indicará un sobrecalentamiento del motor.

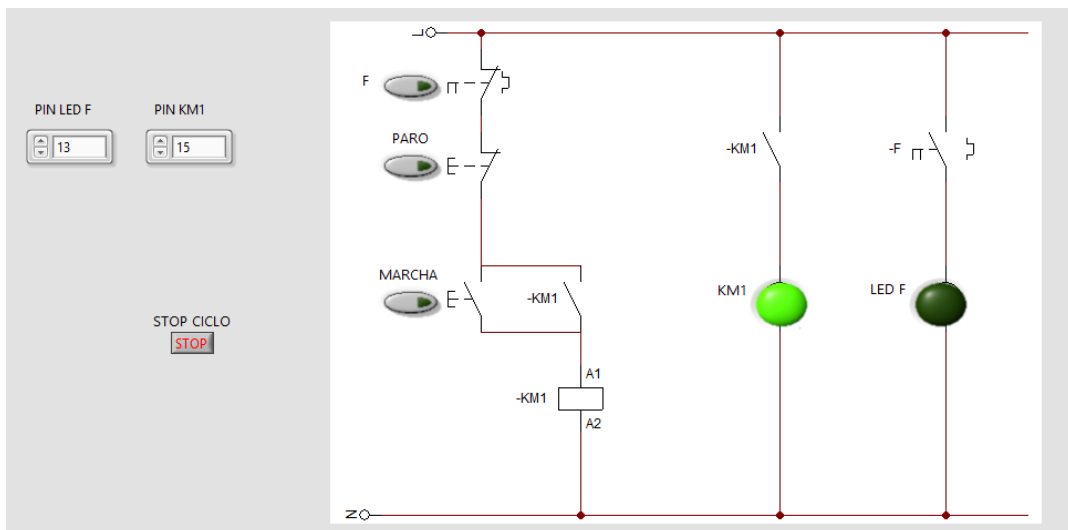


Figura 173. Ejemplo simple de un marcha-paro en LabVIEW con Raspberry Pi, panel frontal. Fuente: Elaboración propia.

Se muestra el diagrama de bloques en LabVIEW:

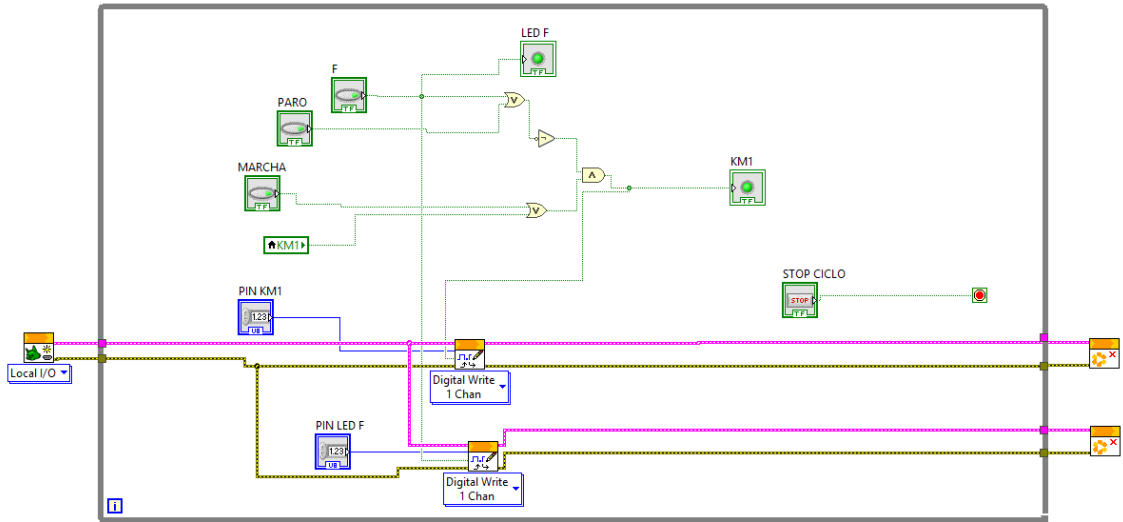


Figura 174. Ejemplo de marcha-paro en LabVIEW con Raspberry Pi, diagrama de bloques. Fuente: Elaboración propia.

Para la conexión real, se utilizarán resistencias de 200 Ω en serie con cada led para no quemarlos, ya que tienen una caída de tensión máxima de unos 3V. Se observa que el pin 15 manda señal al led, esto mostraría que el motor está girando.

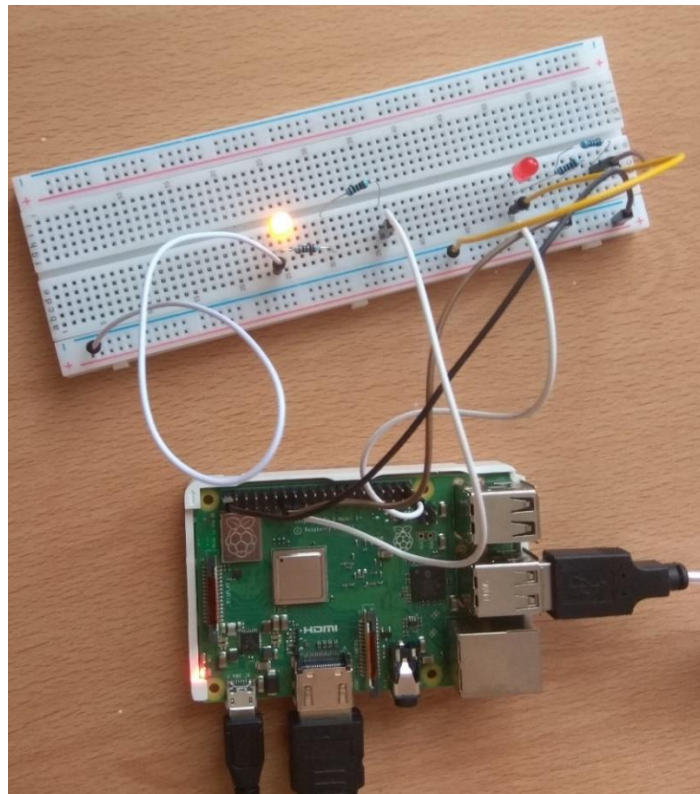


Figura 175. Led encendido con el ejemplo del marcha-paro. Fuente: Elaboración propia.

6.- Control de un motor unipolar como bipolar

Observando los esquemas de funcionamiento de un motor paso a paso unipolar y uno bipolar, se deduce que para utilizar el unipolar como bipolar simplemente hay que dejar en vacío el cable N, que une las mitades de cada bobina.

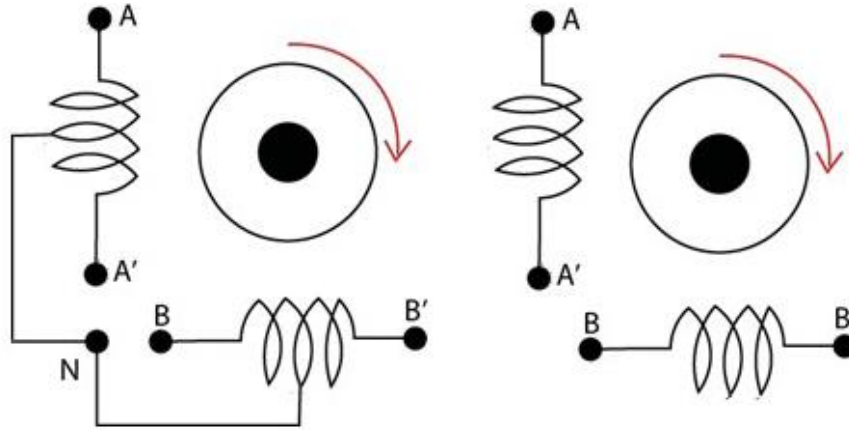


Figura 176. Esquema simplificado de un motor paso a paso unipolar y bipolar, respectivamente. Fuente: [8]

Para controlar el motor bipolar es necesario el uso de dos puentes H, uno para cada bobina.

Un puente H está formado por 4 transistores que, dependiendo de la alimentación de un par de ellos, permitirán el paso de corriente en la bobina en un sentido o en otro. Anteriormente se había explicado la secuencia de alimentación necesaria para controlar un motor paso a paso bipolar, en el apartado 2.1.2.3 de control de motores paso a paso.

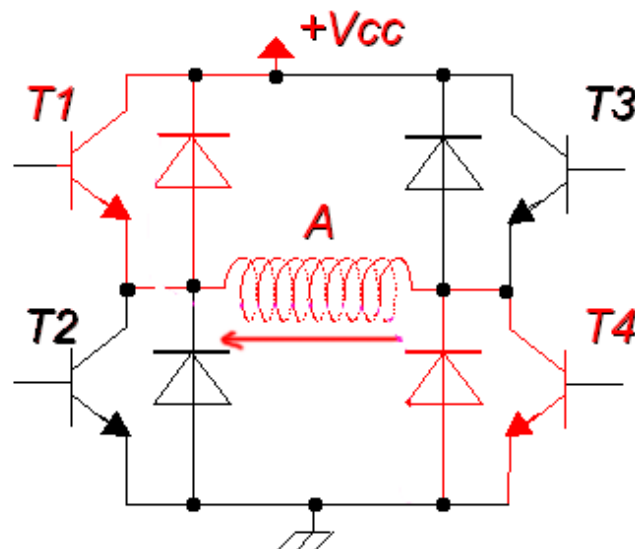


Figura 177. Puente H para devanado de un motor paso a paso bipolar. Fuente: [45]



La señal encargada de alimentar cada par de transistores por bobina sería enviada por el microcontrolador, para ello se realizaría el sistema de control correspondiente en un lenguaje de programación como LabVIEW.





6. BIBLIOGRAFÍA

- [1] Apuntes de la asignatura Física II de la Universidad de Valladolid.
- [2] http://www.cartagena99.com/recursos/alumnos/apuntes/IEE_Tema3_MaquinasElectricas.pdf (Fecha de consulta: 30 de junio de 2020).
- [3] **Jesús Fraile Mora**, *Máquinas eléctricas*, Quinta Edición, Madrid, Mc Graw Hill 2003.
- [4] <https://es.slideshare.net/quasar.0360.7912/semana-2-maquinas-electricas-utpleetmarzo-2012> (Fecha de consulta: 30 de junio de 2020).
- [5] **Moisés San Martín Ojeda, José Andrés Serrano Sanz y Eduardo Parra Gonzalo**, *Laboratorio virtual de Electrotecnia*, 2004.
- [6] <http://www.sectorelectricidad.com/19555/maquina-de-corriente-continua/> (Fecha de consulta: 30 de junio de 2020).
- [7] <https://w3.ual.es/~jmruiz/PRACTICA5.pdf> (Fecha de consulta: 1 de julio de 2020).
- [8] Apuntes de la asignatura Motores Eléctricos Especiales de la Universidad de Valladolid.
- [9] <https://www.luisllamas.es/tipos-motores-rotativos-proyectos-arduino/> (Fecha de consulta: 1 de julio).
- [10] **Clarence W. de Silva**, *Sensors and Actuators. Enineering System Instrumentation*, Segunda Edición, 2016.
- [11] <https://pdfs.semanticscholar.org/88c6/b346bf5d6a6b72f21dfb9fb44e5fb4826d34.pdf> (Fecha de consulta: 15 de junio de 2020).
- [12] **T. Kenjo and A. Sugawara**, *Stepping Motors and Their Microprocessor Controls*, Segunda Edición, 1995.
- [13] <https://www.designworldonline.com/encoder-options-for-all-angles-of-stepping-motors/> (Fecha de consulta: 23 de junio de 2020).
- [14] <https://www.gadgetronicx.com/stepper-motor-working-construction-types-drive/> (Fecha de consulta: 1 de julio de 2020).
- [15] <https://www.oyostepper.com/article-1076-Some-differences-between-bipolar-and-unipolar-stepper-motor.html> (Fecha de consulta: 1 de julio de 2020).



- [16] <https://www.ingmecafenix.com/electricidad-industrial/motor-paso-a-paso/> (Fecha de consulta: 1 de julio de 2020).
- [17] V.V.Athani, *Stepper motors. Fundamentals, Applications and Design*, 2005.
- [18] <http://machineryequipmentonline.com/electric-equipment/stepping-motortransient-performance/> (Fecha de consulta: 24 de junio de 2020).
- [19] <https://disway3dprinter.en.made-in-china.com/product/jZhJRL0DfaWK/China-Fast-Printing-Speed-High-Accuracy-Stepper-Motor-3D-D-Printer.html> (Fecha de consulta: 28 de junio de 2020).
- [20] <https://stargazerslounge.com/topic/291849-my-low-cost-diy-ra-drive-for-eq5-mount/> (Fecha de consulta: 28 de junio de 2020).
- [21] <https://en.wikipedia.org/wiki/File:CNC-Kreuztisch.jpg> (Fecha de consulta: 28 de junio de 2020).
- [22] <https://www.luisllamas.es/brazo-robotico-arduino/> (Fecha de consulta: 28 de junio de 2020).
- [23] <https://www.uelectronics.info/backup/how-use-steppermotor-old-disk-drives.html> (Fecha de consulta: 28 de junio de 2020).
- [24] <https://engresources.uncc.edu/unccengkit/mechanical/motors/stepper-motors/> (Fecha de consulta: 11 de julio de 2020).
- [25] https://es.wikipedia.org/wiki/Raspberry_Pi (Fecha de consulta: 11 de julio de 2020).
- [26] <https://www.xatakahome.com/trucos-y-bricolaje-smart/quieres-una-distribucion-raspbian-ligera-con-este-tutorial-no-puede-ser-mas-sencillo> (Fecha de consulta: 12 de julio de 2020).
- [27] http://www.embeddedknowhow.co.uk/Book%203_files/LN_PDFs/mbed_bk_Ed2_Ch_9.pdf (Fecha de consulta: 12 de julio de 2020).
- [28] <https://tecnoedu.com/Automatizacion/ListadoSitiolabVIEW.php> (Fecha de consulta: 3 de julio de 2020).
- [29] <http://manueldelgadocrespo.blogspot.com/p/motores-paso-paso.html> (Fecha de consulta: 19 de julio de 2020).
- [30] <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (Fecha de consulta: 26 de julio de 2020).



- [31] <https://www.raspberrypi.org/documentation/usage/gpio/> (Fecha de consulta: 26 de julio de 2020).
- [32] <http://robocraft.ru/files/datasheet/28BYJ-48.pdf> (Fecha de consulta: 26 de julio de 2020).
- [33] <https://www.luisllamas.es/motor-paso-paso-28byj-48-arduino-driver-uln2003/> (Fecha de consulta: 26 de julio de 2020).
- [34] http://electronics-diy.com/electronics/stepper_motors.php (Fecha de consulta: 26 de julio de 2020).
- [35] <https://es.aliexpress.com/item/32795760628.html?spm=a2g0s.9042311.0.0.173a63c0vPOyje> (Fecha de consulta: 2 de julio de 2020).
- [36] https://es.wikipedia.org/wiki/Placa_de_pruebas#/media/Archivo:Breadboard_scheme.svg (Fecha de consulta: 26 de julio de 2020).
- [37] <https://circuitglobe.com/hybrid-stepper-motor.html> (Fecha de consulta: 22 de agosto de 2020).
- [38] <http://www.cifpn1.com/electronica/?p=4280> (Fecha de consulta: 23 de agosto de 2020).
- [39] <https://www.digikey.es/es/articles/how-to-power-and-control-brushless-dc-motors> (Fecha de consulta: 9 de agosto de 2020).
- [40] <https://www.luisllamas.es/controlar-un-servo-con-arduino/> (Fecha de consulta: 8 de agosto de 2020).
- [41] <https://www.luisllamas.es/controlar-un-servo-de-rotacion-continua-con-arduino/> (Fecha de consulta: 8 de agosto de 2020).
- [42] https://www.industrialshields.com/es_ES/blog/arduino-industrial-1/post/stepper-motor-speed-control-using-an-arduino-based-plc-and-a-rotary-encoder-64 (Fecha de consulta: 2 de agosto de 2020).
- [43] <https://www.todomicro.com.ar/motores-y-drivers-motores/692-controlador-driver-motor-paso-a-paso-9-a-42v-4a-tb6600.html> (Fecha de consulta: 2 de agosto de 2020).
- [44] <https://tienda.bricogeek.com/controladores-motores/992-controlador-de-motores-paso-a-paso-35a-tb6600.html> (Fecha de consulta: 2 de agosto de 2020).
- [45] <https://forum.arduino.cc/index.php?topic=251836.0> (Fecha de consulta: 1 de agosto de 2020).



- [46] <https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando> (Fecha de consulta: 2 de julio de 2020).
- [47] Manuales de LabVIEW en www.ni.com (Fecha de consulta: 3 de julio de 2020).
- [48] <https://www.electronics-notes.com/articles/test-methods/labview/what-is-labview.php> (Fecha de consulta: 2 de julio de 2020).
- [49] <https://www.alldatasheet.es/datasheet-pdf/pdf/29528/TI/Uln2003AN.html> (Fecha de consulta: 26 de julio de 2020).
- [50] <https://blog.aulamoisan.com/2020/06/rasspberry-beaglebone-labview-interfaceWeb.html> (Fecha de consulta: 28 de junio de 2020).