



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORES: GUERRA GAVILANES, CHRISTIAN SALVADOR

NOVILLO VIVANCO, DAVID GUSTAVO

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO ELECTROMECAÁNICO QUE DESPLIEGUE FÍSICAMENTE CARACTERES BRAILLE PARA LA TRANSCRIPCIÓN DE DOCUMENTOS DE TEXTO DIGITALES

DIECTOR: ING. ORTIZ, HUGO

CODIRECTOR: ING. ALULEMA, DARWIN

SANGOLQUÍ, MAYO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICADO

Ing. Hugo Ortiz, M.Gs
Ing. Darwin Alulema, M.Sc

CERTIFICAN

Que el trabajo titulado **“Diseño e implementación de un prototipo electromecánico que despliegue físicamente caracteres braille para la transcripción de documentos de texto digitales”**, realizado por Christian Salvador Guerra Gavilanes y David Gustavo Novillo Vivanco, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas - ESPE.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Christian Salvador Guerra Gavilanes y David Gustavo Novillo Vivanco que lo entregue al Ingeniero Luis Orozco Brito, en su calidad de Coordinador de la Carrera.

Sangolquí, 19 de Mayo de 2015



Ing. Hugo Ortiz
DIRECTOR



Ing. Darwin Alulema
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

DECLARACIÓN DE RESPONSABILIDAD

Christian Salvador Guerra Gavilanes

David Gustavo Novillo Vivanco

DECLARAMOS QUE:

El proyecto de grado denominado **“Diseño e implementación de un prototipo electromecánico que despliegue físicamente caracteres braille para la transcripción de documentos de texto digitales”**, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

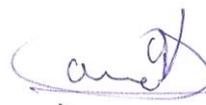
Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 19 de Mayo de 2015



Christian Salvador Guerra Gavilanes



David Gustavo Novillo Vivanco

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Nosotros, Christian Salvador Guerra Gavilanes y
David Gustavo Novillo Vivanco

Autorizamos a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo "Diseño e implementación de un prototipo electromecánico que despliegue físicamente caracteres braille para la transcripción de documentos de texto digitales", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 19 de Mayo de 2015



Christian Salvador Guerra Gavilanes



David Gustavo Novillo Vivanco

AGRADECIMIENTO

Queremos agradecer a la Universidad de las Fuerzas Armadas por darnos la oportunidad de realizar este proyecto.

A nuestros docentes que nos impartieron su conocimiento en el transcurso de nuestra carrera universitaria, en especial al Ing. Hugo Ortiz y al Ing. Darwin Alulema por su motivación y apoyo en la elaboración de este trabajo.

A la empresa Kradac Cia. Ltda. por depositar su confianza en nosotros para el desarrollo del presente proyecto.

Al Instituto para niños ciegos y sordos “Mariana de Jesús”, con especial consideración a la Lic. Gladys Segovia cuya participación en el proyecto fue de vital importancia.

A nuestros padres por su apoyo incondicional.

A nuestros familiares y amigos.

Christian Guerra Gavilanes

David Novillo Vivanco

DEDICATORIA

Dedico este proyecto de tesis a mis padres Salvador y Laurita quienes con mucho amor han estado junto a mí en cada paso que he dado.

A mi hermano Ricardo, mi familia y mis amigos.

Christian Guerra Gavilanes

DEDICATORIA

Dedico este proyecto de tesis a mis padres César y Mónica quienes me han brindado su amor y apoyo incondicional en todo momento. A mi hermano Carlos por su motivación a lo largo del desarrollo de este proyecto.

A mi familia y amigos.

David Novillo Vivanco

Índice de Contenido

CERTIFICADO	i
DECLARACIÓN DE RESPONSABILIDAD	ii
AUTORIZACIÓN.....	iii
AGRADECIMIENTO	iv
DEDICATORIA.....	v
DEDICATORIA	vi
CAPÍTULO 1.....	1
INTRODUCCIÓN.....	1
1.1 Antecedentes	2
1.2 Justificación e importancia	4
1.3 Alcance	4
1.4 Objetivos	5
1.4.1 General	5
1.4.2 Específicos.....	5
CAPÍTULO 2.....	6
MARCO TEÓRICO	6
2.1 Fundamentos básicos	6
2.1.1 Discapacidad visual.....	6
2.1.2 Datos y cifras	6
2.2 Antecedentes históricos	7
2.3 Celda Braille.....	8
2.3.1 Parámetros de la celda Braille.....	10
2.4 Sistema Braille	11
2.4.1 Alfabeto Braille español.....	11

2.4.2	Letras mayúsculas	14
2.4.3	Notación numérica	15
2.5.2	Máquina de escribir Braille	20
2.5.3	Anotador parlante.....	22
2.5.4	Impresora Braille	22
2.5.5	Línea Braille	24
2.6	Recursos de hardware	25
2.6.1	Microcontrolador	25
2.6.2	Convertidor serial USB - TTL	29
2.6.3	Servomotor.....	30
2.6.4	Módulo controlador de servomotores	32
2.7	Recursos de software.....	33
2.7.1	Java	33
2.7.2	NetBeans IDE	34
2.7.3	CCS Compiler	34
CAPÍTULO 3.....		35
DISEÑO DEL PROTOTIPO		35
3.1	Diseño electrónico.....	35
3.1.1	Actuadores	36
3.1.2	Periférico de Entrada.....	37
3.1.3	Unidad de Control	39
3.1.4	Alimentación Eléctrica	46
3.2	Diseño de software.....	48
3.2.1	Funcionamiento General del Software.....	49
3.2.2	Programación en Java	50
3.2.3	Programación del Microcontrolador	80

3.3	Diseño mecánico.....	84
3.3.1	Celdas Braille	84
3.3.2	Bases para los servomotores	86
3.3.3	Piñones sujetos a los servomotores	89
3.3.4	Cremalleras con adaptación para levantar pines.....	93
3.3.5	Pines para formar los caracteres Braille	100
3.3.6	Caja del prototipo	101
3.3.7	Ubicación de los servomotores.....	106
3.3.8	Proceso para desplegar caracteres Braille	108
CAPÍTULO 4.....		114
IMPLEMENTACIÓN DEL PROTOTIPO.....		114
4.1	Selección de componentes mecánicos.....	114
4.1.1	Celdas Braille, piñones, cremalleras y pines	114
4.1.2	Bases para servomotores y caja del prototipo	116
4.2	Selección de componentes electrónicos.....	118
4.2.1	Microcontrolador	118
4.2.2	Controladores para servomotores	121
4.2.3	Servomotores.....	123
4.2.4	Botones para el manejo del dispositivo	124
4.2.5	Convertidor USB-TTL.....	125
4.2.6	Fuente de alimentación	126
4.3	Integración	128
4.3.1	Integración de componentes mecánicos	128
4.3.2	Integración de componentes electrónicos	135
4.3.3	Integración de componentes de software	146

CAPÍTULO 5.....	152
PRUEBAS Y ANÁLISIS DE RESULTADOS	152
5.1 Pruebas de servomotores	152
5.1.1 Distancia recorrida por las cremalleras.....	152
5.2 Pruebas de software	158
5.2.1 Prueba de traducción a Braille de computador	158
5.2.2 Prueba de conversión de Braille a posiciones para servomotores.....	159
5.3 Pruebas de comunicación	161
5.3.1 Prueba de envío de datos desde el computador	161
5.3.2 Pruebas de comunicación del microcontrolador	164
5.4 Prueba de funcionamiento.....	167
5.5 Manual de Usuario	175
5.6 Análisis Económico	175
5.6.1 Detalle de costo	176
5.6.2 Análisis de costo y beneficio	177
CAPÍTULO 6.....	184
CONCLUSIONES Y RECOMENDACIONES	184
6.1 Conclusiones.....	184
6.2 Recomendaciones.....	186
6.3 Trabajos Futuros	188
ANEXOS	189
ANEXO 1: Tabla de Braille de computadora	190
ANEXO 2: Manual de usuario	192
ANEXO 3: Datasheet del microcontrolador PIC 18F2550.....	216
ANEXO 4: Datasheet del controlador para servomotores SSC-32U.....	223

ANEXO 5: Datasheet de la fuente de alimentación	228
ANEXO 6: Programación del microcontrolador PIC 18F2550	230
REFERENCIAS	234

Índice de Figuras

Figura 1. Distribución de puntos en el signo generador (celda Braille).....	9
Figura 2. Representación de la letra "h" en el sistema Braille	9
Figura 3. Distribución de puntos en las celdas Braille	10
Figura 4. Alfabeto Braille español	12
Figura 5. Signos de puntuación Braille.....	13
Figura 6. Caracteres especiales y de uso frecuente	14
Figura 7. Diferenciación entre letra minúscula y mayúscula.....	15
Figura 8. Representación en Braille de las siglas ONU (alternativa 1)	15
Figura 9. Representación en Braille de las siglas ONU (alternativa 2)	15
Figura 10. Prefijo para la representación de números en Braille	16
Figura 11. Números arábigos en Braille	16
Figura 12. Representación Braille del número 12 000	17
Figura 13. Representación Braille del número telefónico 02 600 4000	17
Figura 14. Representación Braille de un número decimal	17
Figura 15. Representación en Braille de un número seguido de una letra ...	18
Figura 16. Número ordinal en Braille	18
Figura 17. Representación en Braille de un número fraccionario	19
Figura 18. Representación en Braille alternativa de una división	19
Figura 19. Representación Braille del número 8 en números romanos	19
Figura 20. Regleta y punzón para la escritura Braille	20
Figura 21. Máquina Perkins.	21
Figura 22. Anotador parlante (Braille 'n Speak).....	22
Figura 23. Impresora Braille moderna.....	23
Figura 24. Línea Braille.....	25
Figura 25. Arquitectura interna de un microcontrolador	26
Figura 26. CP2102 Controlador USB – TTL.....	29

Figura 27. Cables de un servo estándar	31
Figura 28. Ejemplos de posicionamiento de un servomotor	31
Figura 29. Esquema del hardware del controlador SSC-32	32
Figura 30. Diagrama de Bloques del prototipo	35
Figura 31. Diagrama de Bloques con tipos de elementos del prototipo.....	47
Figura 32. Diseño electrónico (circuito esquemático).....	48
Figura 33. Diagrama de Caso de Uso.....	50
Figura 34. UML Clase ConvertirABraille	51
Figura 35. Diagrama de flujo de la clase ConvertirABraille, parte 1/5	52
Figura 36. Diagrama de flujo de la clase ConvertirABraille, parte 2/5	53
Figura 37. Diagrama de flujo de la clase ConvertirABraille, parte 3/5	54
Figura 38. Diagrama de flujo de la clase ConvertirABraille, parte 4/5	55
Figura 39. Diagrama de flujo de la clase ConvertirABraille, parte 5/5	56
Figura 40. UML clase GruposDe20.....	57
Figura 41. Diagrama de Flujo de la clase GruposDe20, parte 1/3.....	57
Figura 42. Diagrama de Flujo de la clase GruposDe20, parte 2/3.....	58
Figura 43. Diagrama de Flujo de la clase GruposDe20, parte 3/3.....	59
Figura 44. UML de la clase HiloComunicacion.....	61
Figura 45. Diagrama de flujo de la clase HiloComunicacion	62
Figura 46. UML de la clase LeerArchivosGuardados	63
Figura 47. UML de la clase LeerPDF	64
Figura 48. UML de la clase LeerTXT	64
Figura 49. UML de la clase PosicionDelServo	66
Figura 50. Diagrama de flujo de la clase PosicionDelServo	67
Figura 51. UML de la clase VentanaAcercaDe	68
Figura 52 Esquema de la Ventana “Acerca De...”	69
Figura 53. Implementación de la Ventana “Acerca de...”	69
Figura 54 Esquema de la Ventana del Braille de Computador	70
Figura 55. Implementación de la Ventana del Braille de Computador	71
Figura 56. UML de la clase VentanaComputerBraille.....	71
Figura 57. Esquema de la Ventana Principal	73
Figura 58. Implementación de la Ventana Principal	73

Figura 59. Esquema de la Ventana Principal al abrir un archivo.	74
Figura 60. Implementación de la Ventana Principal al abrir un archivo.	74
Figura 61. UML de la clase VentanaPrincipal	75
Figura 62. UML de la clase Principal.....	76
Figura 63. Diagrama de flujo del programa del microcontrolador	82
Figura 64. Diagrama de Flujo del proceso para convertir un dato en una posición de servomotor.....	83
Figura 65. Celdas Braille.....	84
Figura 66. Vista frontal de las celdas (dimensiones en milímetros)	85
Figura 67. Vista lateral de las celdas (dimensiones en milímetros)	85
Figura 68. Vista superior de las celdas (dimensiones en milímetros)	85
Figura 69. Dimensiones de la celda en milímetros.....	86
Figura 70. Bases para servomotores	87
Figura 71. Base superior de servomotores (dimensiones en milímetros)	87
Figura 72. Segunda base superior (dimensiones en milímetros)	88
Figura 73. Base inferior de servomotores (dimensiones en milímetros)	88
Figura 74. Distancia entre la base del servomotor y su engranaje	89
Figura 75. Ubicación del piñón entre las bases para servomotores (dimensiones en milímetros).....	90
Figura 76. Diseño de la rueda dentada (dimensiones en milímetros).....	90
Figura 77. Diseño del piñón (dimensiones en milímetros).....	91
Figura 78. Acople del piñón al servomotor	92
Figura 79. Ubicación del piñón en la base (dimensiones en milímetros)	92
Figura 80. Ubicación del servomotor en una de las bases	93
Figura 81. Cremallera con adaptación para levantamiento de pines	93
Figura 82. Ocho combinaciones posibles para cada fila de tres pines	93
Figura 83. Mecanismo para el levantamiento de pines	94
Figura 84. Espacio donde ingresa la punta de la cremallera (dimensiones en milímetros).....	94
Figura 85. Punta de la cremallera (dimensiones en milímetros).....	95
Figura 86. Punta de cremalleras inferiores (dimensiones en milímetros)	95
Figura 87. Interacción entre pines y cremallera	96

Figura 88. Consideraciones para el diseño de dientes de la cremallera.....	96
Figura 89. Diseño de los dientes de la cremallera (dimensiones en milímetros).....	97
Figura 90. Diseño final de la cremallera (dimensiones en milímetros)	97
Figura 91. Diseño de cremalleras superiores (dimensiones en milímetros) .	98
Figura 92. Diseño de cremalleras inferiores (dimensiones en milímetros) ...	98
Figura 93. Vista superior del diseño de la cremallera (dimensiones en milímetros).....	99
Figura 94. Vista lateral del diseño de la cremallera.....	99
Figura 95. Interacción entre el piñón y la cremallera.....	99
Figura 96. Diseño de los pines (dimensiones en milímetros)	100
Figura 97. Ubicación de los pines en las celdas Braille.....	100
Figura 98. Caja del prototipo.....	101
Figura 99. Tapa superior principal (dimensiones en milímetros)	102
Figura 100. Tapa frontal (dimensiones en milímetros)	102
Figura 101. Tapa lateral izquierda (dimensiones en milímetros)	102
Figura 102. Tapa lateral derecha (dimensiones en milímetros).....	103
Figura 103. Tapa inferior (dimensiones en milímetros)	103
Figura 104. Tapa posterior (dimensiones en milímetros)	104
Figura 105. Segunda tapa superior (dimensiones en milímetros)	104
Figura 106. Segunda tapa frontal (dimensiones en milímetros)	104
Figura 107. Tapa protectora para las celdas Braille (dimensiones en milímetros).....	104
Figura 108. Soportes para la tapa protectora (dimensiones en milímetros).....	105
Figura 109. Ensamblaje de las piezas de la caja	105
Figura 110. Ensamblaje de los soportes para la tapa protectora de las celdas Braille	106
Figura 111. Distribución de servomotores en el dispositivo.....	106
Figura 112. Ubicación de los servomotores en la base superior	107
Figura 113. Ubicación de los servomotores en la base inferior	107
Figura 114. Ubicación de servomotores en dos niveles	108

Figura 115. Motores montados en base inferior	108
Figura 116. Combinaciones de pines posibles y los movimientos de la cremallera.....	109
Figura 117. Movimiento de la cremallera	110
Figura 118. Diagrama de cuerpo libre de la cremallera.....	110
Figura 119. Movimiento de los pines	112
Figura 120. Diagrama de cuerpo libre de un pin	112
Figura 121. Nivel superior de las celdas Braille	115
Figura 122. Vista lateral de las celdas Braille.....	115
Figura 123. Piñones fabricados con impresión 3D	115
Figura 124. Cremallera fabricada con impresión 3D	115
Figura 125. Pines fabricados con impresión 3D.....	116
Figura 126. Tapa posterior de la caja	116
Figura 127. Tapa frontal de la caja	117
Figura 128. Lámina de acrílico para la base superior de servomotores	117
Figura 129. Tapa superior de la caja	117
Figura 130. Tapa lateral derecha de la caja	117
Figura 131. Tapa de protección para las celdas Braille acoplada a la segunda tapa superior	118
Figura 132. Circuito esquemático	119
Figura 133. Diseño de la placa para el microcontrolador	120
Figura 134. Vista superior del circuito implementado.....	120
Figura 135. Circuito implementado	120
Figura 136. Controlador para servomotores SSC-32U.....	122
Figura 137. Servomotor TowerPro SG90.....	124
Figura 138. Servomotor TURNIGY TS531A.....	124
Figura 139. Botones implementados en el prototipo	125
Figura 140. Convertidor USB - TTL C92102	125
Figura 141. Fuente de alimentación.....	127
Figura 142. Diagrama de conexión de la fuente de voltaje	127
Figura 143. Fuente de alimentación.....	128
Figura 144. Acople de las láminas de acrílico a las celdas Braille.....	128

Figura 145. Implementación de la base superior para cremalleras	129
Figura 146. Implementación de bases para servomotores	129
Figura 147. Implementación de las cremalleras en el nivel superior	130
Figura 148. Guía ubicada a un costado de la cremallera	130
Figura 149. Celdas Braille con pines para formar los caracteres	131
Figura 150. Acople del servomotor al piñón	131
Figura 151. Servomotores con piñón	131
Figura 152. Ubicación de los servomotores	132
Figura 153. Implementación de los servomotores en el nivel superior	132
Figura 154. Implementación de los servomotores en el nivel inferior	133
Figura 155. Numeración de los servomotores.....	133
Figura 156. Distribución de filas y numeración de servomotores	134
Figura 157. Ubicación de servomotores en dos niveles	134
Figura 158. Ubicación de los componentes en la caja	135
Figura 159. Diagrama de bloques del prototipo	135
Figura 160. Diagrama de conexión de la fuentes de alimentación	136
Figura 161. Diagrama de controladores para servomotores	137
Figura 162. Diagrama de conexión de la unidad de control	138
Figura 163. Diagrama de conexión entre el microcontrolador y el convertidor USB-TTL	139
Figura 164. Conexión entre un puerto del microcontrolador y los controladores de servomotores.....	139
Figura 165. Diagrama de conexión de los servomotores al controlador.....	140
Figura 166. Servomotores controlados por cada tarjeta (SSC1 y SSC2) ...	141
Figura 167. Diagrama de conexión de los botones	143
Figura 168. Diagrama de conexión de los componentes electrónicos.....	144
Figura 169. Conexiones de los componentes electrónicos	144
Figura 170. Puerto USB del prototipo	145
Figura 171. Caja del prototipo.....	145
Figura 172. Prototipo terminado.....	146
Figura 173. Archivo abierto en el programa	146

Figura 174. Asignación de caracteres Braille por cada letra de la trama recibida.....	150
Figura 175. Documento de prueba	158
Figura 176. Documento de prueba para posiciones de servomotores	160
Figura 177. Selección del puerto COM1 como puerto de comunicación	162
Figura 178. Terminal serial Hercules 3-2-8	162
Figura 179. Documento abierto con el software implementado.....	163
Figura 180. Trama obtenida en el terminal serial	163
Figura 181. Caracter obtenido después de presionar el botón avanzar	164
Figura 182. Caracter obtenido al presionar el botón retroceder	165
Figura 183. Caracter obtenido después de presionar el botón inicio de documento.....	165
Figura 184. Caracter obtenido después de presionar el botón fin de documento.....	165
Figura 185. Trama enviada desde el microcontrolador al controlador 1 de servomotores	166
Figura 186. Trama enviada desde el microcontrolador al controlador 2 de servomotores	166
Figura 187. Selección COM4	167
Figura 188. Texto “Documento de” en el programa.....	167
Figura 189. Texto “Documento de” en Braille.....	168
Figura 190. Primeros diez caracteres del texto “Documento de”	168
Figura 191. Últimos diez caracteres del texto “Documento de”	168
Figura 192. Texto “prueba.txt” en el programa.....	168
Figura 193. Texto “prueba.txt” en Braille	169
Figura 194. Primeros diez caracteres del texto “prueba.txt”	169
Figura 195. Últimos diez caracteres del texto “prueba.txt”	169
Figura 196. Texto “UNIVERSIDAD DE” en el programa	170
Figura 197. Texto “UNIVERSIDAD DE” en Braille	170
Figura 198. Primeros diez caracteres del texto “UNIVERSIDAD DE”	170
Figura 199. Últimos diez caracteres del texto “UNIVERSIDAD DE”	170
Figura 200. Texto “LAS FUERZAS” en el programa	171

Figura 201. Texto “LAS FUERZAS” en Braille	171
Figura 202. Primeros diez caracteres del texto “LAS FUERZAS”	171
Figura 203. Últimos diez caracteres del texto “LAS FUERZAS”	171
Figura 204. Texto “ARMADAS” en el programa	172
Figura 205. Texto “ARMADAS” en Braille	172
Figura 206. Primeros diez caracteres del texto “ARMADAS”	172
Figura 207. Últimos diez caracteres del texto “ARMADAS”	172
Figura 208. Texto “ESPE” en el programa	173
Figura 209. Texto “ESPE” en Braille	173
Figura 210. Primeros diez caracteres del texto “ESPE”	173
Figura 211. Últimos diez caracteres del texto “ESPE”	173
Figura 212. Texto “Fin del Archivo” en el programa	174
Figura 213. Texto “Fin del Archivo” en Braille	174
Figura 214. Primeros diez caracteres del texto “Fin del Archivo”	174
Figura 215. Últimos diez caracteres del texto “Fin del Archivo”	174
Figura 216. Dispositivo sin desplegar caracteres	175
Figura 217. Línea 1 traducida a Braille	179
Figura 218. Línea 2 traducida a Braille	179
Figura 219. Línea 3 traducida a Braille	180
Figura 220. Línea 4 traducida a Braille	180
Figura 221. Celdas con falla	181
Figura 222. Pines con falla	182
Figura 223. Archivo Setup.exe para instalar el programa	192
Figura 224. Selección de lenguaje	192
Figura 225. Asistente de Instalación	193
Figura 226. Selección de la carpeta de destino	193
Figura 227. Selección de carpeta de menú Inicio	194
Figura 228. Crear un Ícono de escritorio	194
Figura 229. Resumen de la configuración	195
Figura 230. Progreso de la instalación	195
Figura 231. Finalizar la instalación	196
Figura 232. Ícono del programa instalado	196

Figura 233. Archivos Instalados.....	197
Figura 234. Selección de Puerto COM.....	197
Figura 235. Trabajar sin conexión.....	198
Figura 236. Pantalla Principal	198
Figura 237. Menú Archivo.....	199
Figura 238. Explorador de archivos	199
Figura 239. Filtros de Archivos	200
Figura 240. Menú Configuración.....	200
Figura 241. Mensaje cuando no se ha guardado un puerto por defecto.....	201
Figura 242. Mensaje cuando se requiera configurar un nuevo COM	201
Figura 243. Verificación del puerto COM utilizado	202
Figura 244. Mensaje cuando se trabaja sin conexión	202
Figura 245. Menú Ayuda.....	202
Figura 246. Ventana Acerca De.....	203
Figura 247. Ventana con la Tabla del Braille de Computadora	204
Figura 248. Explorador de Archivos	205
Figura 249. Ventana Principal con un archivo abierto	205
Figura 250. Final del documento con "Fin del Archivo"	206
Figura 251. Apertura de un documento en Modo Sin Conexión	207
Figura 252. Botones izquierdos del dispositivo	208
Figura 253. Botones derechos del dispositivo.....	208
Figura 254. Búsqueda del Panel de Control en Windows 8	209
Figura 255. Búsqueda del Panel de Control en Windows 7	209
Figura 256. Panel de Control	210
Figura 257. Vista del Panel de Control con iconos grandes	210
Figura 258. Administrador de Dispositivos.....	211
Figura 259. Puertos COM del computador.....	211
Figura 260. Selección del puerto COM en el programa	212
Figura 261. Error librería libSerialPort.dll	212
Figura 262. Carpeta de Instalación	213
Figura 263. Librerías para la comunicación serial	213
Figura 264. Copiar los archivos de la carpeta	214

Figura 265. Pegar los archivos de la librería	214
Figura 266. Reemplazar los archivos en el destino.....	215

Índice de Cuadros

Cuadro 1. Dimensiones de la celda Braille	10
Cuadro 2. Familias de PICs de 8 bits.....	27
Cuadro 3. Características técnicas del PIC18F2550.....	28
Cuadro 4. Cuadro comparativo de los tipos de actuadores.....	36
Cuadro 5. Cuadro comparativo de los periféricos de entrada	38
Cuadro 6. Controladores para el control de servomotores	39
Cuadro 7. Cuadro comparativo de tipos de comunicación entre computador y controlador	41
Cuadro 8. Componentes con conexión con el controlador	43
Cuadro 9. Cuadro comparativo de controladores para el prototipo	43
Cuadro 10. Voltajes necesarios para la alimentación del prototipo	47
Cuadro 11. Cuadro con los caracteres ASCII y la posición de los servomotores.....	65
Cuadro 12. Cuadro comparativo de tipos de materiales	114
Cuadro 13. Cuadro comparativo de materiales para bases y caja del prototipo	116
Cuadro 14. Cuadro comparativo entre microcontroladores PIC	118
Cuadro 15. Cuadro comparativo de controladores para servomotores	121
Cuadro 16. Detalle del protocolo de comunicación	122
Cuadro 17. Cuadro comparativo de servomotores.....	123
Cuadro 18. Cuadro comparativo de fuentes de alimentación.....	126
Cuadro 19. Asignación de canales de controladores a servomotores.....	142
Cuadro 20. Asignación de caracteres de la trama para el controlador 1	150
Cuadro 21. Asignación de caracteres de la trama para el controlador 2	151
Cuadro 22. Datos obtenidos para el controlador 1	152
Cuadro 23. Datos obtenidos para el controlador 2.....	153
Cuadro 24. Cuadro de errores para los motores del controlador 1.....	154

Cuadro 25. Cuadro de errores para los motores del controlador 2.....	155
Cuadro 26. Resultados con los nuevos pasos para el servomotor 13l.....	156
Cuadro 27. Errores para los nuevos pasos del servomotor 13l.....	156
Cuadro 28. Resultados con los nuevos pasos para el servomotor 19l.....	157
Cuadro 29. Errores para los nuevos pasos del servomotor 19l.....	157
Cuadro 30. Resultado de la traducción a Braille de computador mediante el software implementado	159
Cuadro 31. Conversión a posiciones de servomotor	161
Cuadro 32. Cuadro de costos de componentes electrónicos	176
Cuadro 33. Cuadro de costos de componentes mecánicos	177
Cuadro 34. Cuadro de costos totales.....	177
Cuadro 35. Resultados de las pruebas	181

RESUMEN

El documento presenta el diseño e implementación de un prototipo capaz de desplegar de manera física los caracteres braille de un texto digital, el cual será enviado por un programa de computador que realiza la transcripción de documentos en formato PDF y TXT a Braille de computador. Para la traducción del texto a Braille se desarrolla un software utilizando Java como lenguaje de programación y bajo la plataforma de programación NetBeans. El proceso de traducción ejecutado por el programa considera las reglas gramaticales del alfabeto Braille en español, incluyendo caracteres especiales pero sin diferenciar si el texto traducido contiene expresiones matemáticas o científicas. En el diseño del prototipo se analizan los componentes electrónicos, de hardware y de software que se pueden utilizar para la implementación. El desarrollo del software cuenta con una interfaz gráfica que muestra el documento escogido en texto plano, además en la parte inferior se observa la línea de texto a ser desplegada en el prototipo en Braille de computadora. El dispositivo Braille se conecta con el computador mediante una conexión USB. La unidad de control es conformada por un microcontrolador PIC18F2550 y dos tarjetas controladoras que se encargan de posicionar 40 servomotores, éstos se comunican de forma serial con el microcontrolador. La implementación de hardware cuenta con el detalle de la construcción del dispositivo, y la integración de los elementos electrónicos con las piezas mecánicas, conformando el sistema electromecánico. Finalmente se muestran los resultados obtenidos del desarrollo del proyecto con las imágenes del funcionamiento del mismo.

PALABRAS CLAVES:

BRAILLE

JAVA

MICROCONTROLADOR PIC

SERVOMOTOR

FORMATO PDF

FORMATO TXT

ABSTRACT

The document shows the design and implementation of a prototype able to physically deploy the Braille characters of a digital text, which will be sent by a computer program that translates documents in PDF and TXT format, to computer Braille. To translate text to Braille, a software is developed using Java as the programming language and under the NetBeans Integrated Development Environment. The translation process executed by the program, contemplate the grammatical rules of Braille in Spanish, including special characters but without distinguishing if the translated text contains mathematical or scientific expressions. The prototype design analyzes the electronic components, hardware and software that could be used in the implementation. The software development has a graphic interface that show the chosen document in plain text, furthermore at the bottom shows the text line to be shown by the prototype, in computer Braille. The Braille device connects to the computer by an USB connection. The Control Unit is formed by a microcontroller PIC18F2550 and two controller boards that are in charge of setting the position of 40 servomotors, these boards communicates with the controller by a serial connection. The hardware implementation shows in detail the device construction and the electronic elements integration with the mechanical parts, conforming the electromechanical system. Finally the results obtained from the project development are shown with pictures of the project performance.

KEY WORDS:

BRaille

JAVA

MICROCONTROLLER PIC

SERVOMOTOR

PDF FORMAT

TXT FORMAT

CAPÍTULO 1

INTRODUCCIÓN

A través de los años el desarrollo tecnológico ha permitido la creación de herramientas para mejorar la calidad de vida de personas con capacidades especiales, con la intención de contribuir a su inclusión en la sociedad actual impulsando una igualdad de oportunidades para todas las personas.

Gracias al sistema Braille, el cual lleva el nombre de su creador Louis Braille, las personas con discapacidad visual han tenido acceso a una mayor cantidad de información, así como la capacidad de comunicarse a través de un lenguaje escrito.

La estandarización del código Braille como sistema de lectura y escritura para personas no videntes, dio paso a la búsqueda de instrumentos que permitan la creación de documentos en lenguaje Braille de una manera simple. A lo largo de la historia se han desarrollado diferentes tipos de herramientas para la escritura. La herramienta más básica es conocida como punzón a partir de la cual se fueron mejorando los sistemas de escritura, dando paso a la máquina de escribir y con el progreso hacia la era digital surge la impresora Braille y más tarde la “línea Braille”.

Actualmente las impresoras Braille tienen gran acogida debido a que permiten impregnar en papel la información que se encuentra en formato digital. Lamentablemente los costos de impresión son mayores a una impresión tradicional, puesto que cada carácter Braille está formado por una matriz de seis puntos de una dimensión mayor a la que un carácter alfanumérico puede tener, lo que implica un mayor volumen de impresión; además la impresión solamente se puede realizar a un lado de la hoja. El costo de la impresora es elevado, superando los US\$ 3000, por lo que su adquisición generalmente no es destinada para uso personal, sino para entidades o instituciones predestinadas a la ayuda de personas no videntes.

Las líneas Braille constituyen una solución de uso personal, ya que estos dispositivos permiten desplegar físicamente una cadena de caracteres Braille sin necesidad de una impresión en papel. La desventaja de estos aparatos se encuentra en su elevado costo (alrededor de US\$ 7000).

1.1 Antecedentes

El Braille es un sistema de lectura y escritura desarrollada para personas no videntes, también es conocido como cecografía. El Braille le permite a las personas invidentes leer y escribir mediante matrices de seis puntos, los cuales son distribuidos en dos columnas y tres filas; estos puntos tienen un alto relieve por medio del cual se representan los diferentes caracteres, es decir, según la posición y el número de puntos en alto relieve dentro de una matriz se representa cada letra del abecedario así como caracteres especiales, de esta forma las personas pueden sentir por medio de las yemas de sus dedos la distribución de dichos puntos e interpretar el texto. (American Foundation for the Blind, 2014)

En 1808 Charles Barbier propuso la utilización de puntos en relieve para la escritura de textos, este sistema se llamó "escritura nocturna"; más adelante en 1819, quiso desarrollar un sistema para los no videntes, el cual se basaba en la utilización de una matriz de 6 líneas y 6 columnas para representar las letras del alfabeto y ciertos sonidos habituales. Este sistema se ganó el apelativo de "sonográfico" de Barbier. En 1821 inventó un sistema de letras de puntos, código alfabético basado en grupos de puntos. Louis Braille adaptó el sistema de Barbier utilizando grupos de 1 a 6 puntos. (Discapnet, 2013)

La conferencia de Londres de educadores de invidentes de 1932, estableció el modelo para los sistemas de puntos, adoptando un alfabeto conocido como Standard English Braille. Éste es idéntico al Braille original con pequeñas modificaciones. Este sistema se usa en todo el mundo angloparlante y ha sido adaptado a la mayoría de las lenguas. (Discapnet, 2013)

Entre los primeros instrumentos que se diseñaron para la escritura en Braille destaca la pizarra o pizarrón de surcos, que consiste en una hoja metálica o de madera labrada en forma de pequeños cuadrados. El papel se colocaba sobre la pizarra, y el lápiz, guiado por los surcos, marcaba un relieve sobre el papel, obteniéndose una caligrafía manual cuadrangular que se podía ver y tocar. (Discapnet, 2013)

En la actualidad se utilizan máquinas de escribir adaptadas para comunicarse con las personas invidentes, y pizarras para guiar al escritor en la formación de caracteres Braille. Además se dispone de la máquina de escribir Hall Braille, una modificación de la máquina de escribir normal, que crea caracteres Braille en el papel en lugar de imprimir en él caracteres alfanuméricos.

Existen dispositivos que fueron desarrollados con un enfoque hacia la interacción de personas no videntes con la tecnología, es así que existen programas computacionales que permiten reproducir el audio correspondiente al texto que hay en la pantalla.

El desarrollo de los equipos mencionados anteriormente, dan paso a la creación de dispositivos conocidos como “líneas Braille” o con su nombre en inglés “Refreshable Braille Display”, los cuales se basan en un mecanismo electro-mecánico que permite representar caracteres Braille, mediante la elevación de puntos por medio de una superficie plana con agujeros dispuestos de manera que estos puntos puedan sobresalir formando las celdas Braille. (Freedom Scientific, 2014)

Debido a la complejidad y elevado coste de estos dispositivos, sólo suelen tener 20 o 40 celdas. También suelen disponer de botones para desplazar el texto y para realizar otras funciones especiales. En algunos modelos la posición del cursor es representada por la vibración de los puntos y en muchos de ellos existe un botón por cada celda para llevar el cursor a esa posición asociada. (Metec, 2014)

1.2 Justificación e importancia

El proyecto se realiza con la finalidad de brindar una ayuda fundamental a personas no videntes, ya que les permite acceder a documentos digitales desde un computador; si bien es verdad que existen dispositivos que funcionan emitiendo sonidos o voz, el leer Braille es fundamental en la formación intelectual de una persona invidente.

El desarrollo de este proyecto permite que las personas no videntes tengan acceso a información en formato digital; además de que al tratarse de documentos digitalizados se reduce el gasto en impresiones y la adquisición de textos en Braille. Los beneficiarios tienen al alcance cualquier libro o documento digital, lo que representa el acceso a una innumerable cantidad de contenidos disponibles en la red.

En el Plan Nacional Para el Buen Vivir, dentro de sus políticas y lineamientos establece: “Reconocer y respetar las diversidades socioculturales y erradicar toda forma de discriminación, sea ésta por motivos de género, de opción sexual, étnico-culturales, políticos, económicos, religiosos, de origen, migratorios, geográficos, etéreos, de condición socioeconómica, condición de discapacidad u otros.”, por lo que el desarrollo de este prototipo promueve esta política de inclusión social, en este caso para personas con una condición de discapacidad, de manera que puedan tener un mejor acceso y uso de toda forma de comunicación. (Plan Nacional Para el Buen Vivir, 2009-2013)

Actualmente ya existen dispositivos que realizan este tipo de traducción, pero tener acceso a ellos representa un problema económico debido a su elevado coste. Es por esto que este proyecto presenta una solución económicamente accesible para el usuario.

1.3 Alcance

El presente proyecto abarca la implementación de un dispositivo electrónico, que permite la transcripción de documentos de texto digitales a lenguaje Braille.

El sistema funciona tomando un conjunto de palabras consecutivas del texto, y a través de un mecanismo electro-mecánico forma los caracteres en Braille de manera dinámica, actualizándose con cada nuevo grupo de palabras.

Este proyecto permite a una persona no vidente leer documentos de texto de un computador a través de un dispositivo electrónico Braille. Se recomienda la asistencia de otra persona que maneje el software donde se importa el texto y para la puesta en marcha del dispositivo. El formato de archivos de texto que se maneja es “.txt” y “.pdf”, y la implementación del software se realizó en el lenguaje abierto Java. El dispositivo permite traducir hasta 20 caracteres por línea incluyendo espacios, además tendrá cuatro botones (avanzar, retroceder, inicio y fin) que le permitirá al usuario navegar por las líneas del documento.

1.4 Objetivos

1.4.1 General

- Implementar un prototipo electromecánico que despliegue físicamente caracteres Braille para la transcripción de documentos de texto digitales.

1.4.2 Específicos

- Desarrollar un programa de computadora que permita el manejo de archivos de texto para su posterior traducción a lenguaje Braille.
- Diseñar un mecanismo controlado electrónicamente que permita manejar el abecedario Braille de manera que refleje cada línea de texto del documento a traducir.
- Construir un prototipo de bajo costo a través del cual el usuario pueda leer cómodamente el texto.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Fundamentos básicos

2.1.1 Discapacidad visual

La Organización Mundial de la Salud (OMS) define una discapacidad como “un término general que abarca las deficiencias, las limitaciones de la actividad y las restricciones de la participación. Las deficiencias son problemas que afectan a una estructura o función corporal; las limitaciones de la actividad son dificultades para ejecutar acciones o tareas, y las restricciones de la participación son problemas para participar en situaciones vitales. Por consiguiente, la discapacidad es un fenómeno complejo que refleja una interacción entre las características del organismo humano y las características de la sociedad en la que vive”. (Organización Mundial de la Salud, 2014)

La discapacidad visual es una condición caracterizada por la limitación total o parcial del sentido de la vista. De acuerdo a la Clasificación Internacional de Enfermedades (CIE-10, 2006), la función visual se divide en cuatro niveles:

- Visión normal
- Discapacidad visual moderada
- Discapacidad visual grave
- Ceguera

Comúnmente el término baja visión hace referencia a la discapacidad visual moderada y grave.

2.1.2 Datos y cifras

Actualmente existen aproximadamente 285 millones de personas con discapacidad visual en todo el mundo, de las cuales 39 millones padecen de ceguera total y 246 millones poseen baja visión. Aproximadamente el 90% de

las personas con discapacidad visual se concentra en los países en desarrollo. (Organización Mundial de la Salud, 2014)

Según el registro nacional de discapacidades del Consejo Nacional de Igualdad de Discapacidades, en Ecuador hay 42082 personas con discapacidad visual. (CONADIS, 2013)

2.2 Antecedentes históricos

Los orígenes de la escritura en relieve datan de mediados del siglo XIII cuando Al-Imam Al-Amadi, al ser una persona no vidente, se vio en la necesidad de utilizar caracteres táctiles para la lectura. Al-Amadi vendía libros por lo que colocaba caracteres en relieve para poder reconocer el título y el precio de cada uno, de esta manera nace la lectura táctil.

En 1784 se desarrolló la impresión en relieve sobre papel por el francés Valentin Haüy. En un principio se utilizó letra itálica presionando una cartulina mojada sobre caracteres de gran tamaño hechos de plomo, pero se hacía necesario tener letras muy grandes por lo que los libros eran muy voluminosos y costosos. Dos años más tarde en 1786 Haüy fundó el primer colegio para ciegos en París (El Instituto para la Juventud Ciega), pues él consideraba que la ceguera no debía asociarse a la ignorancia y la dependencia de los demás.

En 1808, el capitán del ejército francés, Charles Barbier, desarrolló un sistema para que los soldados se puedan comunicar en la noche sin producir ningún sonido. El sistema de Barbier se basaba en la utilización de doce puntos para representar sonidos, por esta razón lo llamó "sonografía", mayormente conocido como escritura nocturna. (Discapnet, 2013)

Barbier pensó que su sistema podría ayudar a los no videntes, representando cada caracter del alfabeto con un máximo de doce puntos y un mínimo de dos puntos en relieve. Es así que en 1821 fue invitado al Instituto para la Juventud Ciega, para presentar a los estudiantes y maestros su sistema de escritura y lectura en relieve. A pesar de la dificultad del sistema,

la “sonografía” fue introducida como un método de enseñanza auxiliar en la escuela.

Durante esa época, Louis Braille era estudiante del instituto y junto a sus compañeros pronto pudieron identificar las falencias del sistema de Barbier además de su complejidad, es así que pasó tres años desarrollando el sistema de puntos hasta que en la edad de quince años decide tener un código adecuado y lo compartió con el doctor Pignier, quien era su mentor y consideró que el sistema de Braille era simple y elegante. El sistema propuesto por Louis Braille se basaba en una celda de seis puntos distribuidos en dos columnas paralelas de tres puntos, permitiendo de esta forma 64 combinaciones posibles.

En 1878, en París, se adoptó oficialmente el Braille como el sistema internacional de escritura y lectura para personas no videntes. Sin embargo, esto no puso fin al uso de múltiples sistemas para la escritura en relieve. En 1917, Estados Unidos declara como estándar el sistema Braille, pero no fue hasta 1932 que en la conferencia de educadores de invidentes en Londres, se acepta un código uniforme para todos los países de habla inglesa alrededor del mundo, adoptando un alfabeto conocido como “Standard English Braille”. (American Foundation for the Blind, 2014)

2.3 Celda Braille

El sistema Braille se basa en una matriz de seis puntos dispuestos en tres filas y dos columnas; dicha matriz se denomina “signo generador”. El conjunto de puntos permite tener 64 combinaciones posibles, incluyendo el espacio en blanco (ningún punto en relieve), de esta manera se pueden representar las distintas letras, números, signos de puntuación y signos especiales pertenecientes a cada idioma.

La identificación de los puntos en el signo generador se establece de la siguiente manera:

- Punto 1 (arriba a la izquierda)

- Punto 2 (al centro a la izquierda)
- Punto 3 (abajo a la izquierda)
- Punto 4 (arriba a la derecha)
- Punto 5 (al centro a la derecha)
- Punto 6 (abajo a la derecha)

En la Figura 1 se muestra la distribución de puntos en la celda Braille.

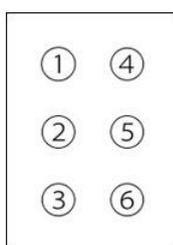


Figura 1. Distribución de puntos en el signo generador (celda Braille)

Fuente: Sistema Braille en Chile (2012).

Para formar un caracter en Braille, se coloca en relieve una combinación de puntos correspondiente al caracter a representar. Para referenciar un caracter se utiliza el número de orden asignado a cada punto de orden ascendente. (Guías de la Comisión Braille Española, 2005)

Por ejemplo, si se desea formar la letra “h” se debe poner en relieve la combinación 125 (leído como: uno dos cinco) tal como se muestra en la Figura 2; donde en azul se indican los puntos en relieve.



Figura 2. Representación de la letra "h" en el sistema Braille

Fuente: Salabrilie-CSBI-Xalapa (2011).

2.3.1 Parámetros de la celda Braille

Para que una línea en Braille sea legible es necesario respetar ciertas medidas establecidas respecto a la separación de los puntos de la matriz así como entre caracteres contiguos y entre líneas. En la Figura 3 muestra la distribución de los puntos en la matriz haciendo referencia a las dimensiones, en el Cuadro 1 se indica el detalle de cada medida.

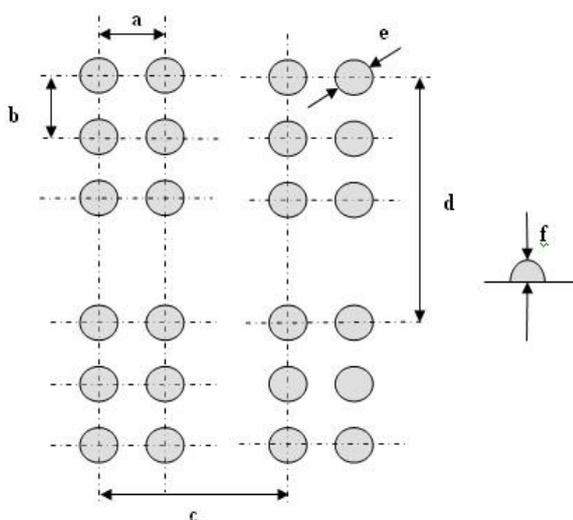


Figura 3. Distribución de puntos en las celdas Braille

Fuente: Guías de la Comisión Braille Española (2005).

Cuadro 1

Dimensiones de la celda Braille

DIMENSIÓN	VALOR MÍNIMO [mm]	VALOR MÁXIMO [mm]
DISTANCIA A	2.5	2.6
DISTANCIA B	2.5	2.6
DISTANCIA C	6.1	7.6
DISTANCIA D	10	10.8
DISTANCIA E	1.2	1.6
DISTANCIA F	0.2	0.5

Fuente: Guías de la Comisión Braille Española, 2005

La altura de la celda Braille debe estar entre 6.2 mm y 6.8 mm y el ancho debe estar entre 3.7 mm y 4.2 mm, el espacio en blanco que separa palabras ocupa la misma dimensión que cualquier caracter, ya que es representado por los seis puntos del signo generador sin relieve.

2.4 Sistema Braille

El Braille no es un idioma en sí, es un sistema de lectura y escritura táctil desarrollado para personas ciegas o con discapacidad visual grave, el cual permite la representación de los caracteres referentes a un idioma (letras, números, signos de puntuación y caracteres especiales).

Con la introducción de la computación, nace el “Código Braille Universal de Computador” también conocido como “Braille de Computador”, el cual se ha convertido en un estándar de hecho reconocido internacionalmente, utilizado para grabar caracteres Braille en archivos de computador. Dicho código asigna las 64 combinaciones de seis puntos de la celda Braille a un conjunto de caracteres ASCII. En el anexo 1 se muestra la tabla con las equivalencias correspondientes. (Alasdair King, 2001)

2.4.1 Alfabeto Braille español

El documento “World Braille Usage” desarrollado por la Escuela para Ciegos Perkins, documenta el estado del Braille alrededor del mundo (142 países) según la UNESCO. En el documento se establece que en el Ecuador se manejan los siguientes parámetros:

- Lenguaje: español (sin contracciones).
- Estándar Braille establecidos por: Comisión Braille Latinoamericana (CBL), Unión Latinoamericana de Ciegos (ULAC) y el Consejo Iberoamericano del Braille (CIB).
- Notación literaria: El Braille integral y la estenografía, 1987.
- Notación matemática: Código Matemático Unificado, 1988.

- Notación musical: International Manual of Braille Music Notation, 1997.
- Alfabeto español: El alfabeto Braille en español se muestra en las siguientes figuras. (Perkins School for the Blind, 2013)

En la Figura 4 se presenta el alfabeto Braille en español. A la derecha de cada letra se encuentra el número que hace referencia a los puntos en relieve de cada celda y a continuación un gráfico de la celda Braille correspondiente. La Figura 4 también incluye las vocales acentuadas con su correspondiente imagen en la celda Braille.

a (1)		r (1)	
b (12)		s (12)	
c (14)		t (14)	
d (145)		u (145)	
e (15)		v (15)	
f (124)		w (124)	
g (1245)		x (1245)	
h (125)		y (125)	
i (24)		z (24)	
j (245)		á (245)	
k (13)		é (13)	
l (123)		í (123)	
m (134)		ó (134)	
n (1345)		ú (1345)	
ñ (12456)		ü (12456)	
o (135)			
p (1234)			
q (12345)			

Figura 4. Alfabeto Braille español

Fuente: Perkins School for the Blind (2013)

La Figura 5 muestra la representación en Braille de los signos de puntuación. Al igual que en la Figura 4, a la derecha de cada letra se encuentra el número que hace referencia a los puntos en relieve de cada celda y a continuación un gráfico de la celda Braille correspondiente.

,	coma	(2)	⠠
;	punto y coma	(23)	⠠⠨
:	dos puntos	(25)	⠠⠨⠨
.	punto	(3)	⠠
¿...?	signos de interrogación	(26...26)	⠠⠨⠨
¡...!	signos de exclamación	(235...235)	⠠⠨⠨⠨
'	apóstrofe	(3)	⠠
"..."	comillas	(236...236)	⠠⠨⠨⠨
(...)	paréntesis	(126...345)	⠠⠨⠨⠨
*	asterisco	(35)	⠠
...	puntos suspensivos	(3,3,3)	⠠⠨⠨⠨
-	guión corto	(36)	⠠
—	guión bajo	(36, 36)	⠠⠨⠨
/	barra oblicua	(6, 2)	⠠⠨

Figura 5. Signos de puntuación Braille

Fuente: Perkins School for the Blind (2013)

La Figura 6 muestra la representación en Braille de algunos caracteres especiales y de uso frecuente.

-	signo de resta	(36)	⠠⠨
x	signo de multiplicar	(236)	⠠⠭
÷	signo de dividir	(256)	⠠⠸
=	signo de igualdad	(2356)	⠠⠨⠠⠸
%	tanto por ciento	(456, 356)	⠠⠠⠠⠠⠠⠠⠠⠠
©	copyright	(126, 46, 14, 345)	⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠
®	marca registrada	(126, 45, 1235, 345)	⠠⠠⠠⠠⠠⠠⠠⠠⠠⠠
&	ampersand	(12346)	⠠⠠⠠⠠⠠
@	arroba	(5)	⠠⠠
//	doble barra oblicua	(6, 26, 2)	⠠⠠⠠⠠⠠⠠
\	barra inversa	(5, 3)	⠠⠠⠠
\\	doble barra inversa	(5, 35, 3)	⠠⠠⠠⠠⠠
	barra vertical simple	(456, 0)	⠠⠠⠠
	barra vertical doble	(456, 123)	⠠⠠⠠⠠
€	euro	(456, 15)	⠠⠠⠠⠠
\$	dólar	(456, 234)	⠠⠠⠠⠠
¢	centavo	(45, 14)	⠠⠠⠠
£	libra esterlina	(5, 2346)	⠠⠠⠠⠠
¥	yen	(456, 13456)	⠠⠠⠠⠠

Figura 6. Caracteres especiales y de uso frecuente

Fuente: Perkins School for the Blind (2013)

2.4.2 Letras mayúsculas

La representación de letras mayúsculas en el sistema Braille es igual a la de las minúsculas con la diferencia de que las mayúsculas requieren de un prefijo formado por los puntos cuatro y seis.

En la Figura 7 se muestra un ejemplo de una letra en mayúscula escrita en Braille. (Guías de la Comisión Braille Española, 2005)

d	(145)		Letra "d" (minúscula)
D	(46, 145)		Letra "D" (mayúscula)

Figura 7. Diferenciación entre letra minúscula y mayúscula

Fuente: Perkins School for the Blind (2013)

Si se desea escribir una palabra completa en mayúsculas, como es el caso de las siglas se puede utilizar dos alternativas:

- Se puede escribir el prefijo de mayúscula antes de cada letra, como se muestra en la Figura 8.

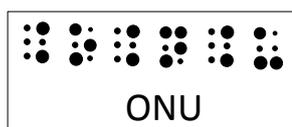


Figura 8. Representación en Braille de las siglas ONU (alternativa 1)

- Se puede escribir dos veces el prefijo de mayúscula consecutivamente previo a toda la palabra, tal como se indica en la Figura 9.

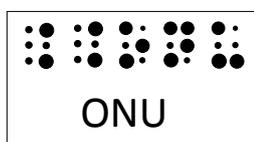


Figura 9. Representación en Braille de las siglas ONU (alternativa 2)

2.4.3 Notación numérica

Para la representación de números en el sistema Braille, al igual que en las mayúsculas es necesaria la utilización de un prefijo, ya que los números están formados por dicho prefijo junto con una letra desde la "a" hasta la "j".

En la Figura 10 se muestra el signo Braille para indicar que se trata de números.

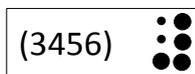


Figura 10. Prefijo para la representación de números en Braille

2.4.3.1 Números arábigos

Como ha sido mencionado, los números arábigos están formados por un prefijo seguido de una letra entre la “a” y la “j”.

En la Figura 11 se muestran los diez dígitos con su respectiva escritura en Braille así como la letra correspondiente a cada número.

a	⠁	→	1	⠼	⠠
b	⠠	→	2	⠼	⠠
c	⠠	→	3	⠼	⠠
d	⠠	→	4	⠼	⠠
e	⠠	→	5	⠼	⠠
f	⠠	→	6	⠼	⠠
g	⠠	→	7	⠼	⠠
h	⠠	→	8	⠼	⠠
i	⠠	→	9	⠼	⠠
j	⠠	→	0	⠼	⠠

Figura 11. Números arábigos en Braille

En caso de que se desee escribir números de más de tres cifras, se debe utilizar en punto (punto 3 de la celda Braille) para la separación en millares tal como se muestra en la Figura 12, cabe mencionar que el prefijo se utiliza una sola vez para todos los dígitos de que formen el número.

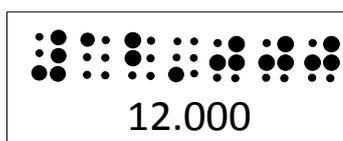


Figura 12. Representación Braille del número 12 000

Si se desea representar números telefónicos el punto que indicaba los millares se puede sustituir con un espacio en blanco, como se indica en el ejemplo de la Figura 13.

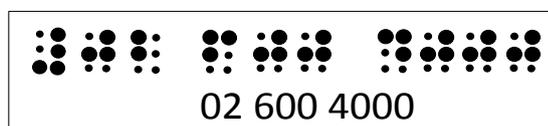


Figura 13. Representación Braille del número telefónico 02 600 4000

Para la escritura de números con decimales, la coma decimal es representada por el punto dos de la celda Braille, como se muestra en el ejemplo de la Figura 14.

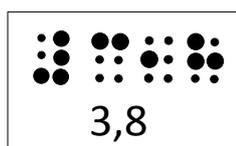


Figura 14. Representación Braille de un número decimal

2.4.3.2 Signo separador de número

Cuando se desea transcribir un número seguido de una letra entre la “a” y la “j”, existe el inconveniente de que este grupo de letras en Braille también se utilizan para la representación de números, es por esto que se necesita un signo separador entre el número y la letra para dar a conocer al lector que el carácter siguiente ya no representa un número.

El signo separador es formado por el punto cinco de la celda Braille como se muestra en el ejemplo de la Figura 15.

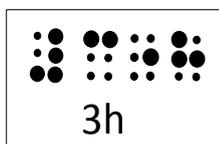


Figura 15. Representación en Braille de un número seguido de una letra

2.4.3.3 Números ordinales

Los números ordinales se forman utilizando “números en posición baja” seguido de las letras “a” u “o” dependiendo del género.

El término “números en posición baja” se refiere a que la representación en Braille de los números arábigos se recorre una posición hacia abajo, es decir que los puntos uno y cuatro de la celda siempre están vacíos. Esto se puede apreciar con claridad en la Figura 16.

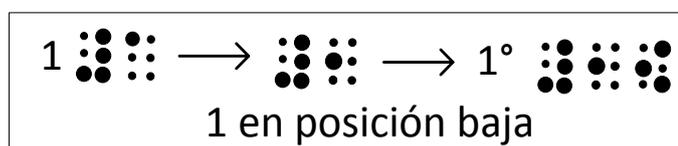


Figura 16. Número ordinal en Braille

2.4.3.4 Números fraccionarios

Cuando sea necesario escribir un número fraccionario, se utiliza el prefijo de número seguido del numerador en posición baja y el denominador en posición normal.

A continuación en la Figura 17 se muestra un ejemplo de número fraccionario.

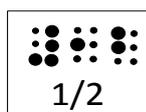


Figura 17. Representación en Braille de un número fraccionario

Dependiendo del texto, los números fraccionarios pueden tener otra representación como la que se muestra en la Figura 18 en donde se escriben los números separados por el signo de división (\div , puntos 256 de la celda Braille).

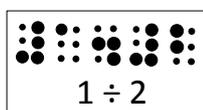


Figura 18. Representación en Braille alternativa de una división

2.4.3.5 Números romanos

Para la representación de los números romanos se utiliza el prefijo de mayúscula delante de los números romanos. En la Figura 19 se ilustra un ejemplo.

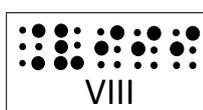


Figura 19. Representación Braille del número 8 en números romanos

2.5 Dispositivos Braille

Desde que Louis Braille desarrolló el sistema de lectura y escritura para las personas ciegas se buscó la mejor manera de lograr transcribir este sistema de forma clara, sencilla y rápida, es así que se desarrollaron las primeras herramientas para la escritura del Braille.

2.5.1 Herramientas básicas

Dentro de las herramientas más básicas se encuentra el punzón y la regleta, los cuales se muestran en la Figura 20. La regleta posee una bisagra y en el lado superior tiene agujeros llamados cajetines, la hoja inferior posee puntos en bajo relieve que guían la escritura del Braille.

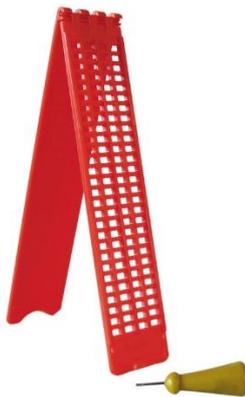


Figura 20. Regleta y punzón para la escritura Braille

Fuente: Didácticos Pinocho S.A (2013)

Para la escritura se coloca la regleta con la apertura en el lado derecho, se abre la regleta y se coloca el papel en el centro, dejando los cajetines hacia arriba. Una vez cerrada la regleta con el punzón se presionan los puntos adecuados para formar la letra deseada; la escritura se realiza de derecha a izquierda en espejo y para leer lo escrito se debe dar la vuelta a la hoja, leyendo de izquierda a derecha. (CEIBAL, 2014)

2.5.2 Máquina de escribir Braille

Desde los primeros niveles, la escritura del sistema Braille se enseñaba con regleta y punzón, una vez aprendida la escritura manual, se pasaba a la escritura a máquina, en la actualidad se enseña a escribir en Braille con la “máquina de punto positivo”, siendo la más utilizada la producida por “Perkins School for the Blind” la cual se la puede apreciar en la Figura 21; más adelante

cuando el alumno tiene las técnicas y destrezas mínimas, se le presenta la regleta como un material complementario para ser utilizado de forma muy esporádica.



Figura 21. Máquina Perkins.

Fuente: Instituto de Tecnologías Educativas (ITE) (2001).

La máquina de escribir Braille consta de nueve teclas:

- Seis teclas, uno para cada punto de la celda Braille.
- Una tecla espaciadora.
- Una tecla para retroceder un espacio.
- Una tecla de cambio de línea

Además consta de un timbre que indica cuando se aproxima el margen derecho. Para escribir una letra determinada se presiona las teclas que conforman dicha letra al mismo tiempo separando palabras con el espaciador y corrigiéndolas con la tecla de retroceso.

A pesar de representar una gran ayuda para las personas no videntes presenta ciertos inconvenientes como su precio, tamaño, peso y el ruido que produce. (Instituto de Tecnologías Educativas, 2001)

2.5.3 Anotador parlante

Como alternativa existen distintos modelos de adaptadores parlantes, que permiten procesar la información y crear archivos como en un computador.

En la Figura 22 se muestra la imagen de este dispositivo de la empresa “Blaze Engineering”. Constan de un teclado similar a la “máquina Perkins” que les permite grabar en memoria el documento escrito para posteriormente tener acceso a este de manera audible, es decir, reproduciendo el texto a través de una voz sintética.



Figura 22. Anotador parlante (Blaille 'n Speak)

Fuente: Productos de Apoyo T.O. (2014)

Este dispositivo presenta algunas ventajas frente a la máquina de escribir Braille al ser un aparato manejable, portátil, liviano y de tamaño reducido, puede comunicarse con otros dispositivos como impresoras, tanto Braille como impresoras tradicionales. (Instituto de Tecnologías Educativas, 2001)

2.5.4 Impresora Braille

La impresora Braille es un dispositivo electrónico que permite imprimir documentos editados desde un computador en el sistema de lecto-escritura Braille, en la Figura 23 se muestra una impresora Braille moderna. La impresión se puede realizar en dos tipos de materiales, papel Braille o papel plástico (“thermoform”).

El papel Braille es agradable al tacto y económico pero requiere de cierto cuidado al ser almacenado para poder preservar la calidad inicial del punto. El papel plástico se mantiene inalterable con el paso del tiempo pero desprende electricidad estática con el rozamiento lo que es desagradable al tacto. (Instituto de Tecnologías Educativas, 2012)



Figura 23. Impresora Braille moderna

Fuente: Productos de Apoyo T.O. (2014)

Una desventaja de la impresión de textos en Braille es que ocupan más espacio que un texto impreso en tinta, lo que triplica su extensión aproximadamente, lo cual ocasiona varios problemas entre ellos están: el transporte, el almacenamiento de los libros y la organización de apuntes. Para solucionar este aspecto se idearon dos formas; la primera es la escritura interpunto, esto implica escribir a los dos lados de la hoja haciendo coincidir de un lado con los espacios entre los puntos del otro. Este método que resulta evidente en una impresión a tinta es difícil de llevar a cabo en Braille para ello es necesario disponer de una impresora especial y como resultado el volumen del texto se vería reducido a la mitad. El segundo método es la estenografía (escritura abreviada) que hace referencia a una serie de normas para abreviar sílabas y terminaciones finales. La finalidad es aumentar la velocidad del lector y reducir el espacio del texto; la estenografía además contiene signos

propios que pueden representar palabras. En español la estenografía no es muy utilizada a pesar de que si existen normas para escribir utilizando este método, también es conocido como Braille grado dos.

Otra desventaja de la impresión Braille es que los puntos se van desgastando en función del uso y trato que se les da, afectando la velocidad del lector hasta el punto en el que el texto se vuelve ilegible. (Instituto de Tecnologías Educativas, 2012)

2.5.5 Línea Braille

Una línea Braille (“Refreshable Braille Display”) es un dispositivo compuesto por un conjunto de celdas que mediante un mecanismo electromecánico, sube o baja un conjunto de seis u ocho puntos por celda representando un carácter Braille en cada una. (Universidad de Alicante, 2008)

Las líneas Braille muestran de forma táctil la información que un “lector de pantalla” leería. Por ello representan una gran ayuda para las personas ciegas y en especial para las sordo-ciegas. Los puntos, a diferencia de un texto en Braille impreso, pueden alternar entre las posiciones de subida y bajada de manera que pueden variar su posición de forma dinámica, es decir, no es necesaria una impresión en papel ya que el dispositivo despliega los caracteres correspondientes a la línea de texto que se esté leyendo, además posee algunos botones que le permiten al usuario utilizarlo como un periférico de entrada en un computador. En la Figura 24 se muestra una línea Braille. (Tecno Accesible, 2011)

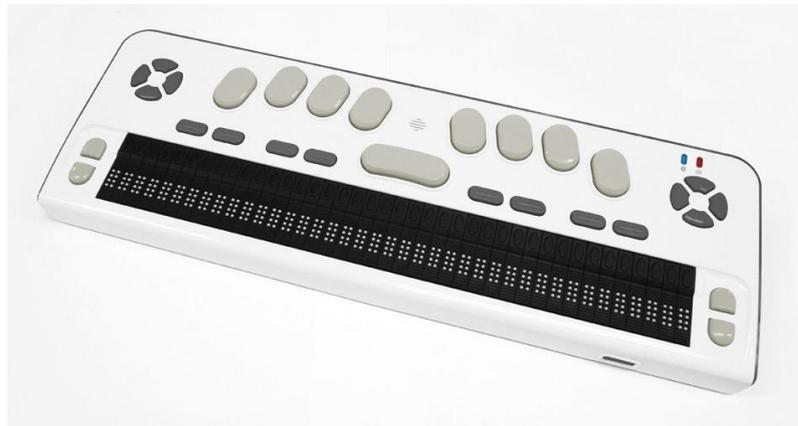


Figura 24. Línea Braille

Fuente: Perkins Products Inc. (2014)

2.6 Recursos de hardware

2.6.1 Microcontrolador

Un microcontrolador es un circuito integrado programable que incluye en su interior las tres unidades funcionales de un computador, CPU, memoria y unidades de entrada y salida.

Los microcontroladores son diseñados para disminuir el coste económico y el consumo de energía de un sistema específico. Por esta razón al momento de escoger las capacidades de CPU, memoria y los periféricos se debe tener en cuenta la aplicación para la cual va a ser destinado. (Galeón – hispavista, 2008)

En la Figura 25 se muestra la arquitectura de un microcontrolador con todos sus componentes y como se encuentran integrados dentro del chip.

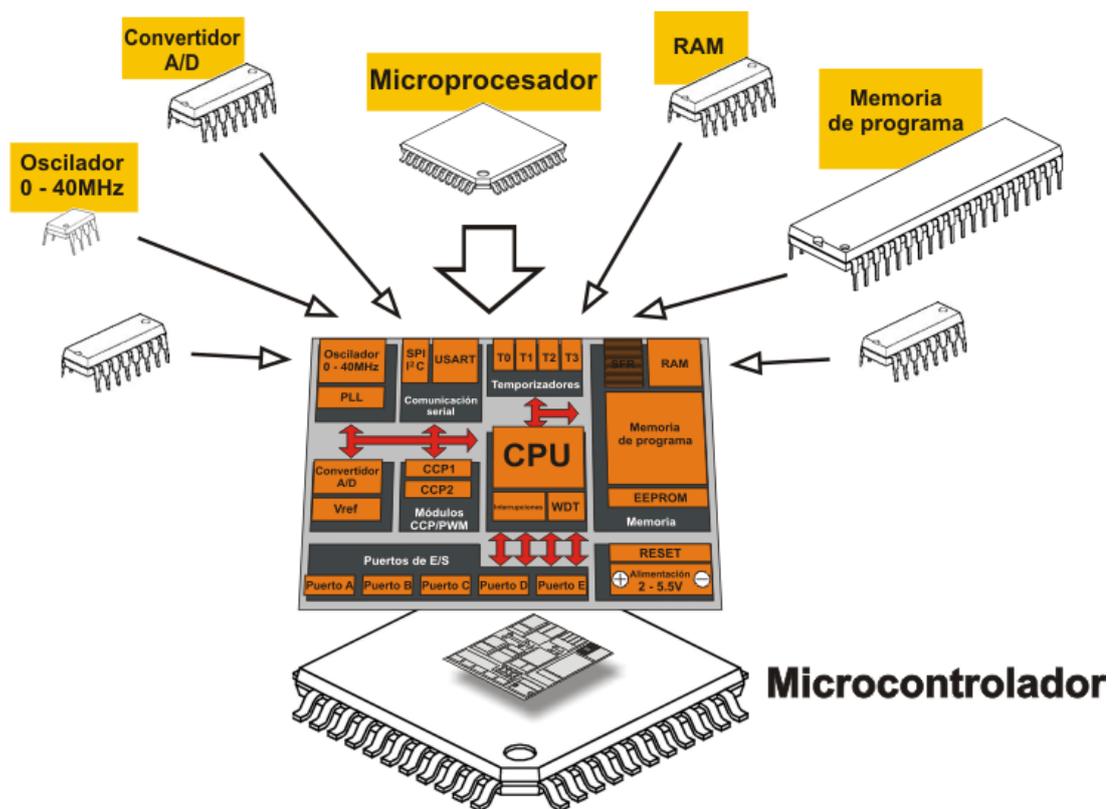


Figura 25. Arquitectura interna de un microcontrolador

Fuente: MikroElektronika (2014)

2.6.1.1 Microcontroladores PIC

Los microcontroladores PIC (nombre verdadero: PICmicro “Peripheral Interface Controller”), son desarrollados por la empresa “Microchip Technology” los cuales se encuentran entre los más comunes en el mercado. Estos microcontroladores están divididos por familias dependiendo de las prestaciones que cada uno posee. De esta manera se los puede clasificar como:

- Arquitectura de la gama baja de 8 bits, palabra de instrucción de 12 bits
- Arquitectura de la gama media de 8 bits, palabra de instrucción de 14 bits
- Arquitectura de la gama alta de 8 bits, palabra de instrucción de 16 bits

Los microcontroladores PIC trabajan con una arquitectura Harvard, es decir, que su memoria de programa está conectada a la CPU por más de ocho líneas. Existen microcontroladores de 12, 14 y 16 bits, dependiendo del ancho del bus. El Cuadro 2 muestra las características de cada familia de PIC. (Mikroelektronika, 2014)

Cuadro 2

Familias de PICs de 8 bits

FAMILIA	ROM [KB]	RAM [B]	FRECUENCIA DE RELOJ. [MHZ]	ENTRADAS A/D	RESOLUCIÓN DEL CONVERTIDOR A/D	TEMPORIZADOR DE 8/16 BITS	COMUNICACIÓN SERIAL
ARQUITECTURA DE LA GAMA BAJA DE 8 BITS, PALABRA DE INSTRUCCIÓN DE 12 BITS							
PIC10 FXXX	0.375 – 0.75	16 – 24	4 – 8	0 – 2	8	1 x 8	-
PIC12 FXXX	0.75 – 1.5	25 – 38	4 – 8	0 – 3	8	1 x 8	-
PIC16 FXXX	0.75 – 3	25 – 134	20	0 – 3	8	1 x 8	-
PIC16 HVXX X	1.5	25	20	-	-	1 x 8	-
ARQUITECTURA DE LA GAMA MEDIA DE 8 BITS, PALABRA DE INSTRUCCIÓN DE 14 BITS							
PIC12 FXXX	1.75 – 3.5	64 – 128	20	0 – 4	10	1 – 2 x 8 1 x 16	-
PIC12 HVXX X	1.75	64	20	0 – 4	10	1 – 2 x 8 1 x 16	-
PIC16 FXXX	1.75 – 14	64 – 368	20	0 – 13	8 o 10	1 – 2 x 8 1 x 16	USART I2C SPI
PIC16 HVXX X	1.75 – 3.5	64 – 128	20	0 – 12	10	2 X 8 1 x 16	USART I2C SPI
ARQUITECTURA DE LA GAMA ALTA DE 8 BITS, PALABRA DE INSTRUCCIÓN DE 16 BITS							
PIC18 FXXX	4 - 128	256 – 3936	32 – 48	4 – 16	10 o 12	0 – 2 x 8 2 – 3 x 16	USB2.0 CAN2.0 USART I2C SPI
PIC18 FXXJ XX	8 – 128	1024 – 3936	40 – 48	10 – 16	10	0 – 2 x 8 2 – 3 x 16	USB2.0 USART Ethernet I2C SPI
PIC18 FXXK XX	8 – 64	768 – 3936	64	10 – 13	10	1 x 8 3 x 16	USART I2C SPI

Fuente: Mikroelektronika, 2014

2.6.1.2 Microcontrolador PIC18F2550

El microcontrolador PIC18F2550 pertenece a la gama alta de “Microchip Technology” cuenta con 28 pines. Algunas de sus características se muestran en el Cuadro 3. En los anexos se adjunta el Datasheet con todas las especificaciones técnicas de este PIC. (Microchip, 2014)

Cuadro 3**Características técnicas del PIC18F2550**

PARÁMETRO	VALOR
Tipo de memoria de programa	Flash
Memoria de programa (kb)	32
Melocidad del cpu (mips)	12
Bytes en ram	2048
Datos en la memoria eeprom	256
Periféricos de comunicación digital	1-UART, 1-A/E/USART, 1-SPI,1-I2C1-MSSP(SPI/I2C)
Periféricos de captura/comparación/pwm	2 CCP
Temporizadores	1 x 8-bit, 3 x 16-bit
Convertidores analógico/digital	10 canales, 10-bit (resolución)
Comparadores	2
Usb (canal, velocidad, tipo)	1, Full Speed, USB 2.0
Rango de temperatura (°c)	-40 a 85
Rango de voltaje de operación (v)	2 a 5.5
Número de pines	28

Fuente: Microchip, 2014

La razón para utilizar un microcontrolador de gama alta es que existe la necesidad de manejar tres puertos seriales de manera simultánea, el primero para la comunicación con el computador, el segundo y el tercero para el manejo de los dos controladores para los servo motores, además el dispositivo requiere de cuatro entradas digitales para el control del programa. El número de pines del PIC18F2550 cumple con las necesidades mencionadas anteriormente en un tamaño reducido.

2.6.2 Convertidor serial USB - TTL

En el presente proyecto se utiliza el convertidor CP2102, el cual es un controlador serial USB – TTL altamente integrado que provee una solución simple para actualizar los diseños RS-232 a USB usando un mínimo número de componentes y espacio como se muestra Figura 26.

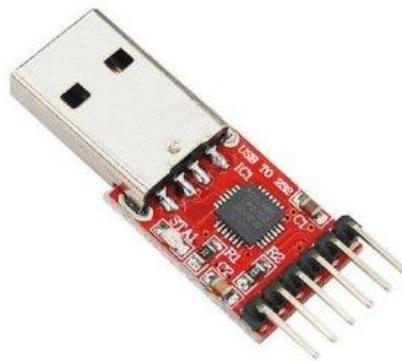


Figura 26. CP2102 Controlador USB – TTL

Fuente: Silicon Laboratories (2013)

Este dispositivo incluye un controlador USB 2.0, un “transceiver” USB, un oscilador, una memoria EPROM o EEPROM y un bus de datos seriales asíncronos (UART) en un tamaño compacto de 5mm x 5mm y no requiere componentes USB externos.

Los controladores para puertos COM virtuales (VCP) provistos por “Silicon Laboratories” permiten al producto aparecer como un puerto COM para ser utilizado por aplicaciones de computadora. (Silicon Laboratories, 2013)

El motivo de su uso en el presente proyecto es que el puerto USB en la actualidad se lo puede encontrar en cualquier computador personal, además el convertidor provee una comunicación directa con el puerto serial del microcontrolador.

2.6.2.1 Comunicación serial TTL

La comunicación serial es un protocolo muy común para comunicación entre dispositivos que por lo general se incluye de manera estándar en cualquier computadora.

El concepto de comunicación serial se puede definir de la siguiente manera: el puerto serial envía y recibe bytes (8 bits) de información un bit a la vez. A pesar de que es más lento que una comunicación en paralelo (8 bits a la vez), este método de comunicación es más sencillo y puede alcanzar mayores distancias.

Las cadenas seriales de bits generadas por los puertos serie de la computadora utilizan una forma simple de codificación. Un bit se transmite durante un determinado periodo de tiempo con un bit "1" representado por un voltaje alto TTL (5 VDC) y un "0" por un voltaje bajo TTL (0 VDC).

2.6.3 Servomotor

El servomotor es un dispositivo que cuenta en su interior con un pequeño motor, un reductor de velocidad y un multiplicador de fuerza, además consta de un circuito que controla el sistema tanto en velocidad como en posición. En la mayoría de casos tiene un ángulo de giro máximo de 180° lo que puede ser modificado para tener un giro libre de 360°.

El control del servomotor se lo realiza a través de un pulso de duración y frecuencia específicas. Todos los servomotores disponen de tres cables, como se muestra en la Figura 27; dos de alimentación (VCC y GND) y un tercero al cual se aplica un tren de pulsos de control con un periodo de 10 ms a 22 ms, que hace que el circuito de control diferencial interno ubique al servomotor en la posición indicada dependiendo del ancho de pulso. (Universidad Técnica Federico Santa María, 2003)

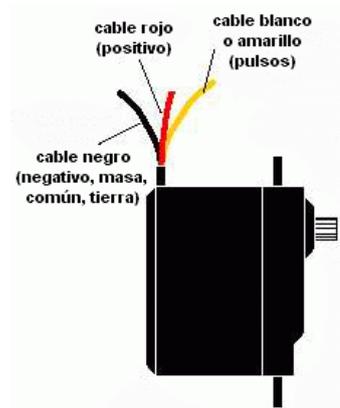


Figura 27. Cables de un servo estándar

Fuente: Robotica PY (2003)

Cabe mencionar que la disposición de los colores de los cables puede variar de acuerdo al fabricante. En la Figura 28 se puede apreciar ejemplos del posicionamiento del servomotor dependiendo del ancho de pulso teniendo las posiciones:

- 0° con ancho de pulso de 0.5 ms
- 90° con ancho de pulso de 1.5 ms
- 180° con ancho de pulso de 2.5 ms

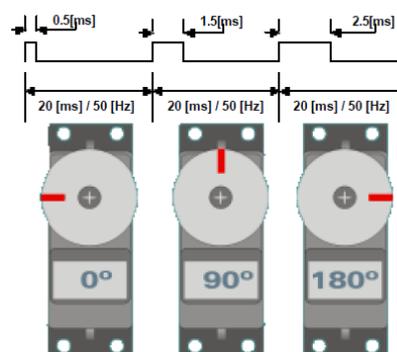


Figura 28. Ejemplos de posicionamiento de un servomotor

Fuente: EcuRed (2014)

2.6.4 Módulo controlador de servomotores

En el presente proyecto se utiliza el controlador SSC-32 “Servo Controller” para el manejo de los servomotores, el cual consta de 32 canales, una resolución de $1\mu\text{s}$ y posee una comunicación bidireccional con comandos de consulta. La razón por la que fue escogido radica en que permite el manejo de 32 servomotores a partir de una misma fuente de alimentación, además la resolución de $1\mu\text{s}$ permite un control de posición más preciso. El prototipo requiere del control de cuarenta servomotores, razón por la cual es necesaria la utilización de dos controladores que actúen de manera simultánea. En la Figura 29 se muestra el hardware del controlador seguido de una explicación de las partes que lo componen.

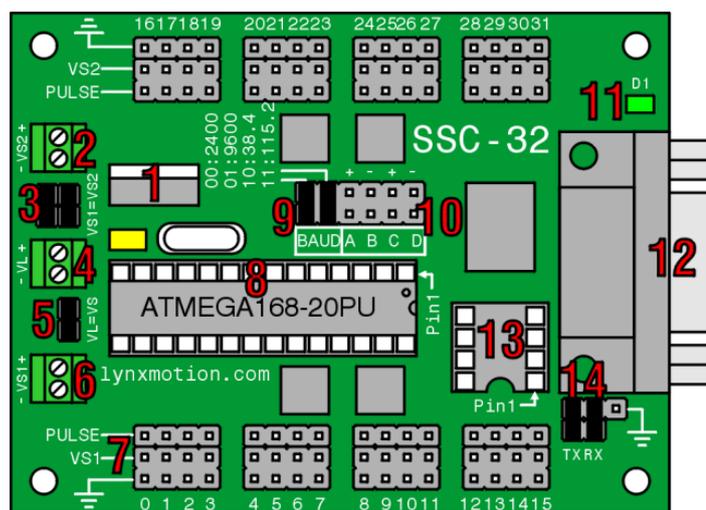


Figura 29. Esquema del hardware del controlador SSC-32

Fuente: Jim Frye (2010)

1. El regulador de baja caída proporciona 5 Vdc con un voltaje de mínimo de 5.5 Vdc. y uno máximo de 9 Vdc.
2. Terminal que da la alimentación a los canales 16 a 31 para los servomotores.
3. Jumpers utilizados para conectar VS1 a VS2. Esta opción se utiliza cuando se requiere alimentar todos los servos desde la misma fuente.

4. Voltaje lógico. Esta entrada puede ser utilizada con una batería de 9V.
5. Jumper que permite dar la alimentación al microcontrolador y soportar la circuitería desde la fuente de alimentación del servo. Se requiere al menos 6V para que funcione correctamente.
6. Terminal que da la alimentación a los canales 0 a 15 para los servomotores.
7. Pines de conexión para los servomotores u otros dispositivos de salida.
8. Microcontrolador Atmel.
9. Entradas que permiten configurar la velocidad de transmisión (baudios).
10. Entradas ABCD que tienen soporte estático y de enclavamiento.
11. Procesador Good LED. Se mantiene encendido cuando existe alimentación.
12. Conector DB9 para la comunicación serial con el computador.
13. Zócalo para una memoria EEPROM de ocho pines.
14. Puerto serial para niveles de voltaje TTL. (Frye, 2010)

2.7 Recursos de software

2.7.1 Java

Java es un lenguaje de programación y una plataforma de computación lanzado por Sun Microsystems en 1995. Este lenguaje es rápido, seguro y confiable, además que su descarga desde Internet es gratuita. Java es una tecnología que se usa para el desarrollo de aplicaciones, es concurrente, basado en clases y orientado a objetos y específicamente diseñado para tener la menor cantidad posible de dependencias en su implementación, es decir que el código que es ejecutado en una plataforma no necesita ser compilado nuevamente para poder ejecutarlo en otra. (Java, 2014)

2.7.2 NetBeans IDE

Es un entorno de desarrollo integrado, gratuito y de código libre. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio como para aplicaciones web y dispositivos móviles dando soporte a las siguientes tecnologías: Java, PHP, groovie, C/C++, HTML5 entre otros, y puede instalarse en varios sistemas operativos como Window, OS X y Linux. (Genbeta, 2014)

2.7.3 CCS Compiler

El compilador CCS fue desarrollado para compilar código en C para los microcontroladores de “Microchip Technology” el cual ofrece soluciones de desarrollo para aplicaciones embebidas utilizando PICs, MCUs y dispositivos dsPIC DSC. Este compilador incluye optimización de alto nivel, una gran cantidad de librerías con funciones integradas y ejemplos listos para ser ejecutados. (Custom Computer Services, 2014)

CAPÍTULO 3

DISEÑO DEL PROTOTIPO

3.1 Diseño electrónico

El sistema electrónico está conformado por los siguientes bloques:

- Periférico de Entrada
- Actuadores
- Unidad de Control
- Alimentación Eléctrica

En la Figura 30 se muestra el diagrama de bloques del prototipo.

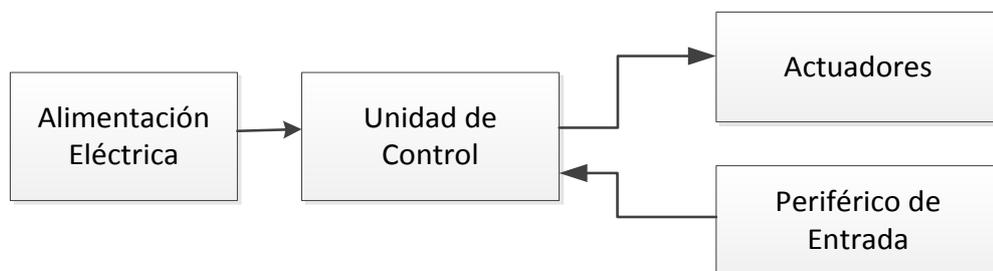


Figura 30. Diagrama de Bloques del prototipo

En el diagrama de bloques de la Figura 30 se puede apreciar las etapas fundamentales del proyecto, las cuales actúan de la siguiente manera:

- Actuadores: Reciben datos procesados por la unidad de control según los cuales despliegan físicamente los caracteres Braille del documento de texto.
- Periférico de entrada: Envía datos a la unidad de control para la navegación dentro del documento de texto.
- Unidad de control: Encargada de la apertura y traducción a Braille del documento de texto, además envía los comandos de control a los actuadores y recibe datos del periférico de entrada.
- Alimentación eléctrica: Provee la energía adecuada a cada uno de los bloques para el correcto funcionamiento del dispositivo.

3.1.1 Actuadores

El prototipo requiere el manejo de 120 pines que permiten la formación de 20 caracteres Braille de manera dinámica. Cada uno de los pines tiene dos estados, cuando están activados el usuario puede palparlos sobre la superficie del prototipo, en cambio al estar desactivados se ocultan de la superficie. En el Cuadro 4 se detallan los actuadores considerados para cumplir el propósito indicado, teniendo en cuenta que deben ser capaces de manipular cada pin en un área de 1.5 x 1.5 mm para cumplir la norma Braille.

Cuadro 4

Cuadro comparativo de los tipos de actuadores

CARACTERÍSTICA	SERVOMOTOR	MOTORES A PASOS	COMPONENTES PIEZOELECTRICOS
TIPO SEÑAL DE CONTROL NECESARIA	TS531A Señal analógica PWM	28BYJ-48 Serie de impulsos eléctricos	Cosson KS-202T68A Dos niveles de voltaje On – Off
VOLTAJE DE ALIMENTACIÓN	+4 VDC a +6 VDC	+5 VDC, +12VDC, +24VDC Unipolar y Bipolar.	200 a 300 VDC
TAMAÑO	19 x 8 x 17.5 [mm]	35x31x12 [mm]	8 x 5 x 0.22 [mm]
MOVIMIENTO	Movimiento circular de 0° a 210°	Movimiento circular, 200 pasos constituyen una vuelta.	Movimiento lineal 0.18 mm a 0.65 mm
CANTIDAD REQUERIDA	40	40	120
PRECIO UNITARIO	US\$ 3.99	US\$ 4.72	US\$ 35
DISPONIBILIDAD EN EL PAÍS	Si	Si	No

Generalmente se utilizan los componentes piezoeléctricos para la fabricación de un dispositivo Braille, pero estos no están disponibles dentro del país y su adquisición implica un problema puesto que los componentes dedicados a este tipo de aplicaciones son desarrollados y vendidos únicamente por empresas que fabrican dispositivos Braille, lo que hace que su precio se eleve mucho más allá de los US\$35 y que solo sean vendidos para propósitos específicos de desarrollo de tecnología. Además se debe

considerar que al utilizar componentes piezoeléctricos se necesita de un actuador por cada uno de los pines mientras que con los motores el sistema puede manejar tres pines a la vez.

Las razones mencionadas hacen que la implementación con motores sea una opción más económica y accesible, además que al existir disponibilidad en el país, disminuye el tiempo de adquisición y por ende el de implementación del proyecto. Para la implementación con motores, ya sean servomotores o motores paso a paso, se requiere cuarenta de estos elementos para el manejo de los 120 pines del dispositivo.

Los servomotores han sido escogidos como la mejor opción para el proyecto sobre los motores paso a paso, pues presentan un rango de movimiento suficiente para una interacción completa entre los piñones y las cremalleras, además presentan un tamaño más reducido a un costo menor.

3.1.2 Periférico de Entrada

Una vez abierto un documento de texto, este se mostrará en Braille en el dispositivo para que el usuario pueda leer el texto veinte caracteres a la vez. El dispositivo ofrece al usuario opciones de navegación dentro del documento, las cuales son seleccionadas mediante un periférico de entrada de datos.

Las cuatro opciones de navegación son detalladas a continuación:

- Avanzar: Solicita al programa el despliegue de la siguiente línea de texto del documento que está siendo leído.
- Retroceder: Solicita al programa el despliegue de la anterior línea de texto del documento que está siendo leído.
- Inicio: Solicita al programa el despliegue desde la primera línea de texto del documento que está siendo leído. Esta línea siempre tendrá el nombre del archivo abierto.

- Fin: Solicita al programa el despliegue desde la última línea de texto del documento que está siendo leído. Esta línea siempre tendrá la frase "Fin del Archivo".

En el Cuadro 5 se detallan los periféricos de entrada considerados para la navegación del texto.

Cuadro 5

Cuadro comparativo de los periféricos de entrada

CARACTERÍSTICA	BOTONES	TECLADO PC
Tamaño	10 x 10 x 4 [mm]	40 x 12 x 1.50 [cm]
Movimiento	Movimiento circular de 0° a 210°	Movimiento circular, 200 pasos constituyen una vuelta.
Cantidad requerida	4	1
Precio unitario	US\$ 0.20	US\$ 7.50
Disponibilidad en el país	Si	Si

Para el correcto funcionamiento del dispositivo es necesaria la entrada de cuatro datos diferentes.

Teniendo en cuenta que el proyecto cuenta con una computadora para la lectura del archivo de texto, no es necesaria la adquisición de otro teclado como periférico de entrada mencionado en el Cuadro 5, en este caso se asignarían cuatro teclas del mismo teclado para que cumplan las funciones de navegación. La diferencia que se debe tomar en cuenta, entre el teclado y los botones, es que los botones se ubican en el dispositivo lo que da mayor comodidad al usuario que solo debe recurrir al uso del ordenador para escoger el archivo de texto que desea leer, en cambio al utilizar el teclado como periférico de entrada el usuario pierde independencia al tener que recurrir al ordenador para cambiar de línea del texto. En este caso, el aspecto

económico no representa una diferencia considerable puesto que el costo de los botones es bajo.

Se escoge a los botones como la mejor opción para la implementación en el prototipo, tomando en cuenta la facilidad de manejo que ofrecen al estar ubicados junto a la línea de texto en Braille y que la persona no vidente puede sentir su activación al presionarlos, lo que les brinda mayor seguridad de haber enviado el comando deseado.

3.1.3 Unidad de Control

La Unidad de Control se encarga del procesamiento de los datos de entrada y de salida enviados al ordenador y a los servomotores, para posicionarlos según corresponda.

Control de Servomotores

El prototipo final debe ser capaz de manejar cuarenta servomotores, por lo que se analiza las siguientes opciones de controladores:

Cuadro 6

Controladores para el control de servomotores

CARACTERÍSTICA	MICROCONTROLADOR	TARJETA ARDUINO (MEGA)	CONTROLADOR DE SERVOMOTOR
Tamaño	52.45 x 14.22 x 4.06 [mm]	101.6 x 53.3 x 10 [mm]	76.20 x 58.42[mm]
Salidas pwm	2	14	32
Cantidad de necesaria para cumplir los requerimientos	20	3	2
Precio unitario	US\$ 11.20	US\$ 33	US\$ 55
Disponibilidad en el país	Si	Si	Si

Entre las opciones presentadas en el Cuadro 6, el microcontrolador no se lo considera una buena opción para el control de varios servomotores de manera simultánea puesto que este consta únicamente de dos salidas PWM necesarias para la ubicación de los motores, en este caso se necesitan veinte microcontroladores para el control de los 40 servomotores como se indica en el Cuadro 6.

La tarjeta Arduino Mega permite el uso de 14 señales PWM para el control de los servomotores, en este caso se necesitan tres tarjetas para el control de los 40 motores. Por otro lado la tarjeta controladora para servomotor tiene 32 salidas de PWM y se necesitan de dos de estas tarjetas para el control de todos los servomotores.

Se optó por la utilización de las dos tarjetas controladoras para servomotores, puesto que al ser un controlador dedicado ofrecen más opciones de configuración para el movimiento de los servomotores, estas pueden definir la velocidad que se demora cada servomotor en alcanzar la posición final y el orden de movimiento de cada servomotor. Además solo se necesitan dos tarjetas para cubrir la necesidad del proyecto, lo que permite que se ocupe menos espacio dentro del prototipo final.

Los controladores para servomotores son sistemas microcontrolados que permiten manejar las posiciones de los servomotores a través de comandos enviados a través de una comunicación serial.

Cada controlador requiere de alimentación externa de 6 VDC para la parte de control, y la corriente necesaria para que los 32 servomotores puedan moverse.

Comunicación PC - Controlador

Para la comunicación entre la computadora y el controlador del dispositivo electrónico se consideraron cuatro posibilidades diferentes, las cuales son detalladas en el Cuadro 7.

Cuadro 7

Cuadro comparativo de tipos de comunicación entre computador y controlador

CARACTERÍSTICA	COMUNICACIÓN USB DIRECTA	CONVERTIDOR RS-232 A TTL	CONVERTIDOR USB A SERIAL TTL	MÓDULO BLUETOOTH
Tipo	Cable USB tipo A.	Max 232 + Cable USB – DB9	CP2102	HC-06
Velocidad de la comunicación	-Low speed 1.5Mbps -Full speed 12Mbps -High speed 480 Mbps -Super speed 5 Gbps	Según norma 232C: -300 bps -600 bps -1200 bps -2400 bps -4800 bps -9600 bps -19200 bps	-Full speed USB 2.0 12 Mbps.	Ajustable' -1200 bps -2400 bps -4800 bps -9600 bps -19200 bps -38400 bps -57600 bps -Máxima 1382400 bps
Niveles de voltaje	+ 5 VDC - 0 VDC	+ 3VDC a + 15VDC - 3VDC a - 15VDC	+ 5 VDC - 0 VDC	+ 3.3VDC a 5 VDC
Necesidades adicionales	Controlador que soporte comunicación USB.	Circuito externo convertidor a niveles TTL. Conector RS232 en la computadora o convertidor USB a RS232.	Controlador que soporte comunicación serial niveles TTL.	Computador con módulo Bluetooth integrado.
Precio	US\$ 1.50	US\$ 17	US\$ 10	US\$ 18
Disponibilidad en el país	Si	Si	Si	Si

Al comparar los cuatro tipos diferentes de comunicación se puede notar que la comunicación por convertidor RS-232 a TTL es la menos recomendable puesto que se necesita de un circuito externo adicional para su funcionamiento, además los niveles de voltaje en los que opera requieren de una fuente eléctrica adicional, por estas razones esta opción es descartada.

El módulo Bluetooth es una buena opción para establecer una conexión inalámbrica con el computador, brindando así mayor comodidad al usuario. La gran desventaja que esto presenta es que requiere que la computadora cuente con un módulo bluetooth integrado, o a su vez que se conecte un adaptador bluetooth para que se realice la conexión con él dispositivo. El

proyecto no requiere que la conexión sea inalámbrica por lo que se ha descartado esta opción para no imponer un requerimiento de hardware adicional al usuario.

Las opciones de USB directo y de comunicación serial a través de un convertidor USB a serial TTL, presentan una implementación sin inconvenientes a través de una interfaz USB disponible en todos los ordenadores.

El inconveniente que puede presentar en la implementación la comunicación por USB directo, es que las opciones de controladores que soportan USB son reducidas. Además, en el caso de programación con un microcontrolador, la comunicación serial permite la utilización de interrupciones lo cual es beneficioso porque asegura la recepción de los datos por parte del microcontrolador.

El convertidor USB a serial TTL es la mejor opción para la implementación. Este es un componente electrónico de tamaño reducido que se encarga de transformar la comunicación serial del controlador, la cual se da a niveles TTL, a una comunicación serial tipo USB usada por la computadora. Por otro lado crea en el ordenador un puerto Com virtual que puede ser accedido por las aplicaciones seriales o por un terminal de comunicación serial.

Controlador

El controlador es la parte central del procesamiento del dispositivo. Este se encarga de comunicarse con el software del ordenador para recibir y procesar los datos según los cuales controla la posición de los servomotores.

Para la selección del controlador se debe tomar en cuenta las características de los componentes seleccionados anteriormente que se deben conectar al controlador, estos se detallan en el Cuadro 8.

Cuadro 8

Componentes con conexión con el controlador

COMPONENTE	CANTIDAD	TIPO DE COMUNICACIÓN CON EL CONTROLADOR
Servomotor	40	No se conecta.
Botones	4	Entradas digitales.
Tarjeta controladora para servomotores	2	Comunicación serial con niveles TTL.
Convertidor serial a usb ttl	1	Comunicación serial con niveles TTL.

Los componentes especificados en el Cuadro 8 generan los siguientes requerimientos que debe cumplir el controlador:

- Manejo de cuatro entradas digitales.
- Tres puertos para comunicación serial de manera simultánea.

En el Cuadro 9 se comparan dispositivos de control capaces de cumplir con los requerimientos mencionados.

Cuadro 9

Cuadro comparativo de controladores para el prototipo

CARACTERÍSTICA	MICROCONTROLADOR	TARJETA ARDUINO	PC	TARJETA FPGA	TARJETA NI DAQ
Tipo	PIC 18F2550	Arduino MEGA	Procesador Intel G1610 2.6ghz 320gb 2gb	ZYBO Zynq™-7000 (Z-7010 + Dual UART IP Core D2692	NI myRIO-1900 (FPGA Xilinx Z-7010)
N° de puertos seriales	3	4	0	2	2
N° de entradas y salidas digitales	24	54	0	8	40
Voltaje de alimentación	4.5 – 5.5 VDC	6-20 VDC	110 VAC	3.3 VDC	6 – 16 VDC
Dimensiones	1.03 x 1.78 [cm]	10.15 x 5.33 [cm]	14.00 x 28.55 [cm]	10.00 x 7.00 [cm]	13.66 x 88.6 [cm]
Costo	US\$ 11.20	US\$ 33	US\$ 310.00	US\$ 189.00 + US\$30.00	US\$ 650
Disponibilidad en el país	Si	Si	Si	No	No

Entre las opciones mostradas en el Cuadro 9, la computadora es la opción que se debe descartar inmediatamente pues no cumple los requerimientos necesarios para la comunicación serial con los dos controladores de servomotores, por esta razón es necesario utilizar un controlador que se comunique con el ordenador y envíe de manera serial los comandos de control a los controladores.

El microcontrolador PIC18f2550 es la opción con menor costo para la implementación, su desventaja al compararlo con los demás controladores es que necesita del convertidor USB a serial TTL para su conexión con la computadora, mientras que los FPGA y la tarjeta Arduino tienen integrado este tipo de convertidor en su Hardware, lo que les permite conectarse al ordenador mediante un cable USB de manera directa. A pesar de la implementación de este convertidor, el costo total del microcontrolador incluyendo el convertidor USB a serial TTL es de US\$ 21.20 que sigue siendo un costo mucho menor que el de las tarjetas de adquisición de datos, y ligeramente menor a la tarjeta Arduino.

La tarjeta DAQ de National Instruments tipo NI myRio 1900 es un dispositivo de diseño embebido basado en una FPGA Xilinx Z7010. Esta tarjeta cuenta con tecnología de entrada/salida reconfigurable en tres conectores distintos, además tiene compatibilidad WiFi, procesador ARM en tiempo real dual-core, cuenta con 8 entradas analógicas, 4 salidas analógicas, 32 líneas de entrada/salida digitales, 2 puertos de comunicación serial, Leds, pulsadores, acelerómetro interno y una conexión USB. Las características descritas indican que la tarjeta de adquisición de datos cumple con los requerimientos de comunicación serial necesarios para el control de los servomotores, sin embargo el procesamiento que esta va a manejar es bajo, y no existe la necesidad de usar la mayoría de características que el NI myRio ofrece, por lo que el precio que tiene de US\$ 650.00 es demasiado elevado en comparación al trabajo que debe realizar. Esta alternativa se descarta puesto que existen otras opciones que se ajustan de mejor manera a las necesidades del proyecto.

El FPGA Z-7010 de la marca Xilinx es el FPGA que está implementado dentro del sistema embebido de la tarjeta NI myRio. Al ser utilizado de manera independiente, no provee todas las características que el NI myRio ofrece, sin embargo cuenta con la mayoría de características necesarias para el funcionamiento del proyecto a un costo mucho menor que la tarjeta de National Instruments. El único requerimiento que no cumple el FPGA es la disponibilidad de dos puertos para la comunicación serial para lo cual es necesario adquirir el componente D2692 el cual es un complemento para el FPGA llamado “Dual UART IP Core” que le permite contar con dos puertos seriales y utilizarlos de manera simultánea. El costo final del FPGA con el núcleo D2692 es de US\$ 219.00, un precio mucho más bajo que el NI myRio.

La tarjeta Arduino Mega es una opción que cubre todos los requerimientos del proyecto a un costo bajo, esta se basa en el microcontrolador ATmega1280, cuatro puertos seriales (UART) y cuenta con 54 pines de entrada y salida digitales. A diferencia del microcontrolador PIC18f2550, este se conecta a través de un cable USB directamente a la computadora.

A partir de las opciones mostradas en el Cuadro 9 se concluye lo siguiente de cada opción. La computadora se debe descartar por no cumplir los requerimientos del proyecto. Las FPGAs presentan soluciones similares que cuentan con una conexión USB hacia la computadora, y dos puertos de comunicación serial hacia los controladores. Debido a su costo elevado ninguna de estas dos opciones puede ser implementada, además estas tarjetas no están disponibles en el país lo cual incrementa su precio y aumenta el tiempo de implementación del proyecto. Por último se determina que los sistemas microcontrolados ofrecen una solución que se apega de mejor manera a las necesidades del proyecto a un costo bajo. La tarjeta Arduino Mega brinda una conexión USB con el ordenador y cuatro puertos seriales de los cuales solo se necesitan dos. También cuenta con entradas análogas que no se utilizan en el proyecto, y se requiere cuatro entradas digitales de las 54 que son provistas. Su costo de US\$ 33.00 es mayor al costo de la implementación con el PIC18f2550 (US\$ 21.20).

A pesar de que las prestaciones de la tarjeta Arduino y el microcontrolador PIC sean muy similares, se escoge al microcontrolador como la mejor opción para la implementación en el proyecto puesto que permite crear un circuito dedicado para el control del dispositivo sin desperdiciar recursos, como pasaría con la tarjeta Arduino, además el circuito ocupa un espacio reducido dentro de la caja del proyecto y tiene un consumo menor de energía.

El Microcontrolador

El microcontrolador PIC 18f2550 se encarga de gestionar la comunicación entre el computador y el dispositivo.

Este microcontrolador utiliza un puerto serial para la conexión con el ordenador, del cual recibe la información sobre la posición que los servomotores deben tomar para formar los caracteres Braille. La información es procesada y convertida en valores para definir el ciclo de trabajo de una señal PWM, dichos valores son enviados a los controladores para la generación de señales PWM con ciclo de trabajo variable para posicionar a los servomotores. Para esto el microcontrolador trabaja con tres puertos seriales de manera simultánea, uno para la comunicación con el convertidor USB a serial TTL que a su vez se conecta con la computadora a través del puerto USB, y dos puertos para manejar cada controlador para servomotores.

El microcontrolador también recibe las señales generadas por los botones y envía a la computadora la información sobre qué acción de navegación el usuario quiere realizar, de manera que el software interpreta esta información y determina que función de navegación se debe realizar con el texto (avanzar a la siguiente línea, retroceder a la línea anterior, inicio o fin del documento).

3.1.4 Alimentación Eléctrica

Los requerimientos de energía eléctrica del prototipo están detallados a continuación en el Cuadro 10.

Cuadro 10

Voltajes necesarios para la alimentación del prototipo

COMPONENTE	CANTIDAD	VOLTAJE REQUERIDO
Servomotor	40	4 VDC – 6VDC
Botones	4	Sin alimentación
Tarjeta controladora para servomotores	2	5 VDC
Convertidor serial a usb ttl	1	5 VDC
Microcontrolador	1	5 VDC

Los actuadores requieren de una alimentación de 6 VDC la cual debe ser provista a la etapa de potencia de los controladores para servomotores. Este voltaje es generado por una fuente externa con la potencia necesaria para que los 40 servomotores puedan funcionar al mismo tiempo.

La fuente se encarga de recibir la corriente alterna tomada de la red eléctrica (110V/60Hz) y transformarla en corriente continua con un voltaje de salida de 6 VDC.

El convertidor serial a USB TTL necesita una alimentación de 5 VDC la cual la obtiene del puerto USB del computador, además entre los pines de salida brinda 5 VDC con los cuales se alimenta el microcontrolador PIC y la etapa de control de las tarjetas controladoras para servomotores.

Después de definir qué tipos de elementos se utilizarán en la implementación, en la Figura 31 se muestran estos elementos dentro de cada etapa del diagrama de bloques.

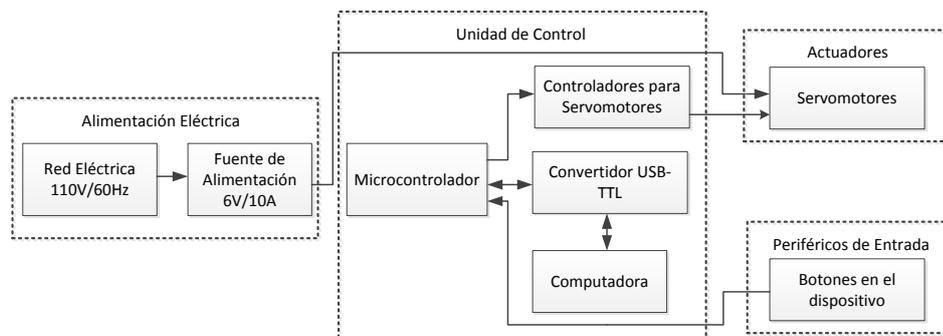


Figura 31. Diagrama de Bloques con tipos de elementos del prototipo

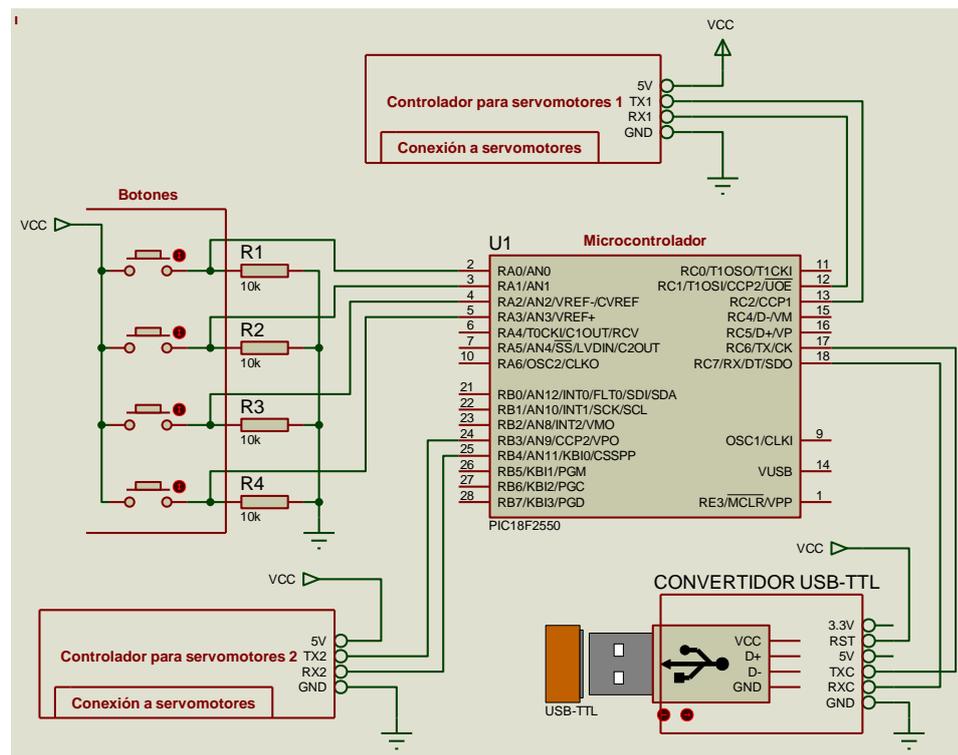


Figura 32. Diseño electrónico (circuito esquemático)

Las conexiones entre los elementos del prototipo se muestran en la Figura 32, donde los botones se conectan al puerto A del microcontrolador, el puerto UART se conecta al convertidor USB a serial TTL que a su vez está conectado a un puerto USB del ordenador, y los controladores para servomotores se conectan a dos puertos seriales generados por software, el primero de ellos generado por los pines 12 y 13 del microcontrolador pertenecientes al puerto B, y el segundo puerto serial por los pines 24 y 25 pertenecientes al puerto C.

3.2 Diseño de software

El software del proyecto se divide en dos partes, la programación del microcontrolador y la programación en Java. A continuación se describe el funcionamiento general de los programas y la manera en que interactúan entre ellos, además se describe la arquitectura utilizada para la programación y las principales etapas que conforman cada programa.

3.2.1 Funcionamiento General del Software

El primer proceso que se realiza al iniciar el programa es el de comunicación, en este punto se busca los puertos COM habilitados y se los muestra como una opción para el usuario. Posterior a haber escogido un puerto COM se procede a escoger el archivo de texto o PDF que se requiere abrir, al cual se le añade al inicio del documento el nombre del mismo seguido por su extensión y al final de este la frase “Fin del Archivo”.

Si el archivo escogido ha sido abierto anteriormente, este se abrirá desde la última posición en que se detuvo la lectura, caso contrario el documento empezará desde la primera línea que tiene el nombre del archivo y continuará con el texto del documento.

Después de escoger el archivo, el programa traduce todo el documento a Braille de computador y se separa el texto en líneas de 20 caracteres considerando que no se dividan las palabras. La razón por la que se necesita separar el texto es que el dispositivo puede desplegar físicamente hasta 20 caracteres de manera simultánea, limitante que se determinó en el alcance del proyecto para evitar que el tamaño del prototipo final sea excesivo.

Una vez separadas las líneas del texto traducido, se procede a codificar las posiciones para los servomotores con el fin de formar cadenas de caracteres que se envían al microcontrolador. Se envían datos al dispositivo con las posiciones de servomotor seguidas por la letra “Y” que ayuda al microcontrolador a confirmar que ha recibido las posiciones de los veinte caracteres a desplegarse. El microcontrolador a su vez envía comandos al computador para navegar en el texto, estos comandos permiten avanzar o retroceder a la línea de texto siguiente o posterior según corresponda, así como también ir al inicio o al final del documento.

Después de finalizar la aplicación, el programa envía un comando para que el dispositivo se ubique en la posición inicial, y guarda la posición final de los documentos leídos para su continuar la lectura la próxima vez que sean abiertos.

3.2.2 Programación en Java

Para la programación en Java se utiliza el entorno de desarrollo Netbeans, considerando una arquitectura de software de tipo modelo-vista-controlador, de esta manera se establece los siguientes paquetes conteniendo las clases necesarias para el funcionamiento del programa.

- Modelo
- Vista
- Controlador

En la Figura 33 se muestra el diagrama de casos de uso, donde se observa la forma en que el usuario opera con el sistema.

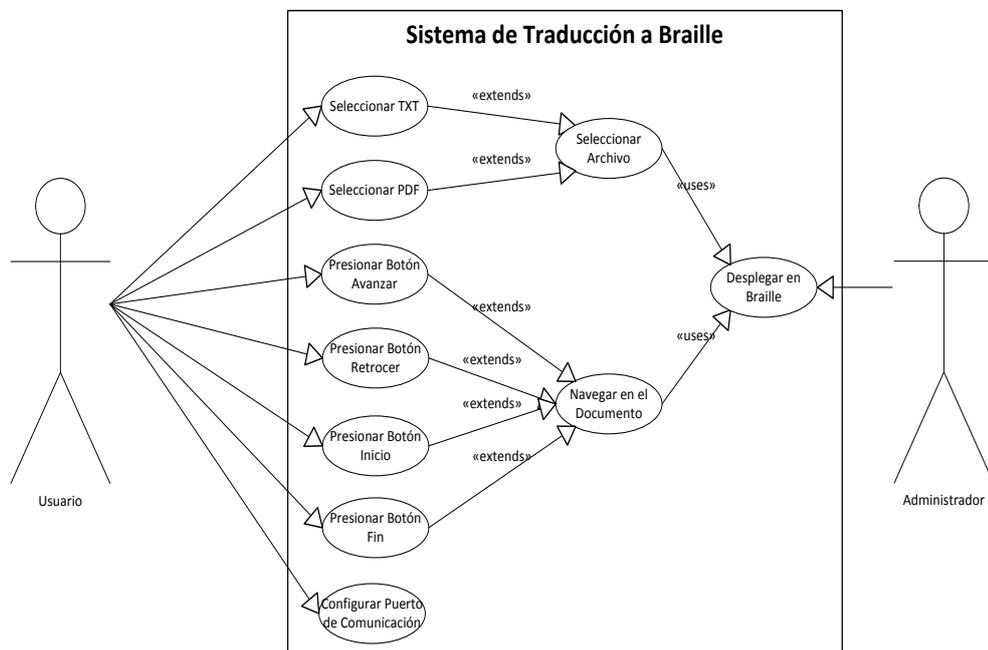


Figura 33. Diagrama de Caso de Uso

3.2.2.1 Paquete Modelo

Este paquete incluye las siguientes clases:

- ConvertirABraille

- GruposDe20
- HiloComunicacion
- LeerArchivosGuardados
- LeerPDF
- LeerTXT
- PosicionDelServo

ConvertirABraille

Esta clase contiene dos métodos, el método constructor de la clase y el método para convertir un texto a Braille el cual se llama traducirabraille(String texto_in).

El método para traducir un texto a Braille recibe como parámetro un dato de tipo String que es una cadena de caracteres con el texto escogido para que sea traducido a braille según las reglas del Sistema Braille en castellano. Como resultado devuelve una cadena de caracteres con el texto en Braille de Computador (Anexo 1), el cual realiza una equivalencia entre los seis puntos del signo generador en Braille considerándolos como un número binario y el carácter en código ASCII. En la Figura 34 se muestra el diagrama UML de la clase, y en la Figura 35, Figura 36, Figura 37, Figura 38 y Figura 39 se muestra el diagrama de flujo de la clase.



Figura 34. UML Clase ConvertirABraille

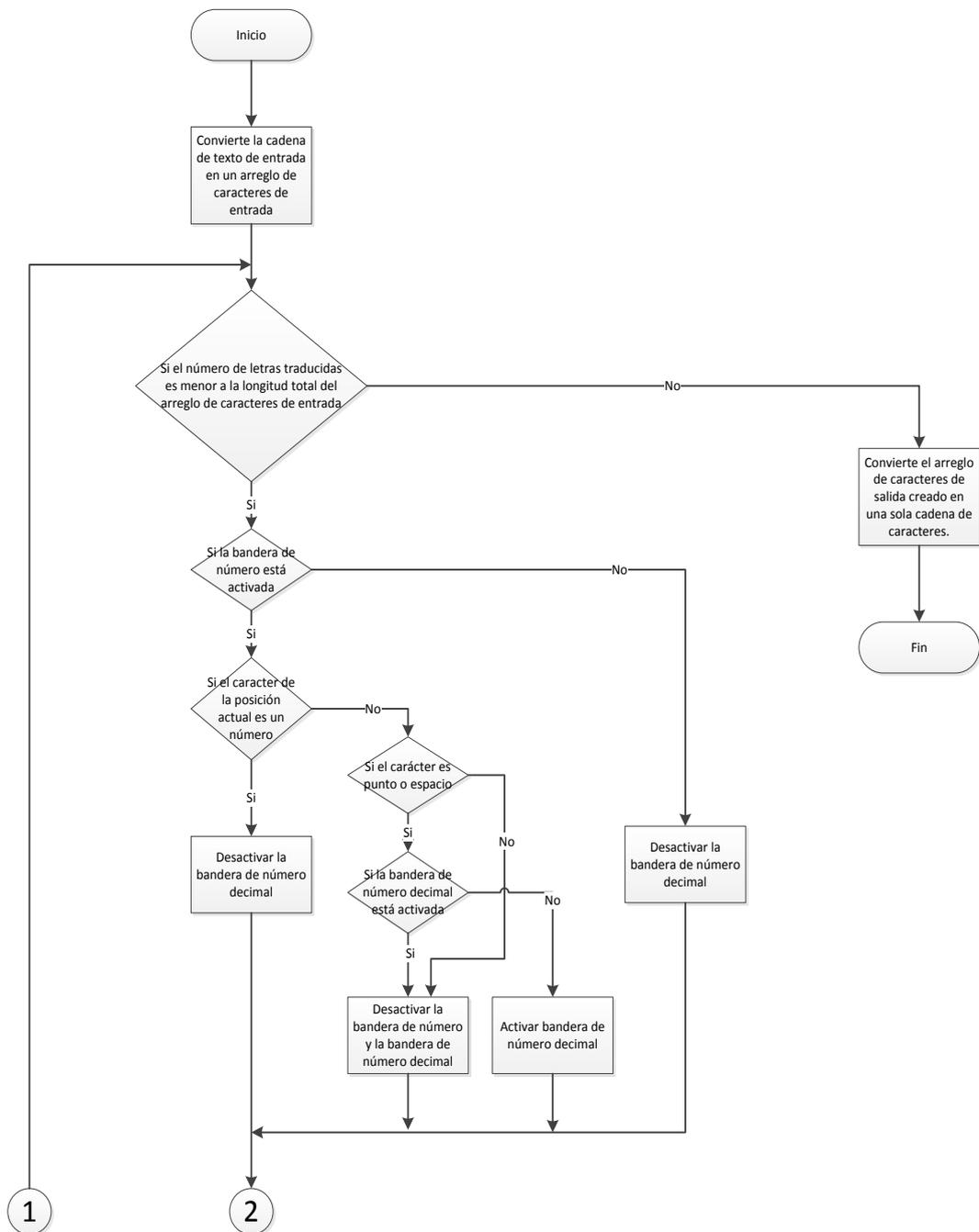


Figura 35. Diagrama de flujo de la clase ConvertirABraille, parte 1/5

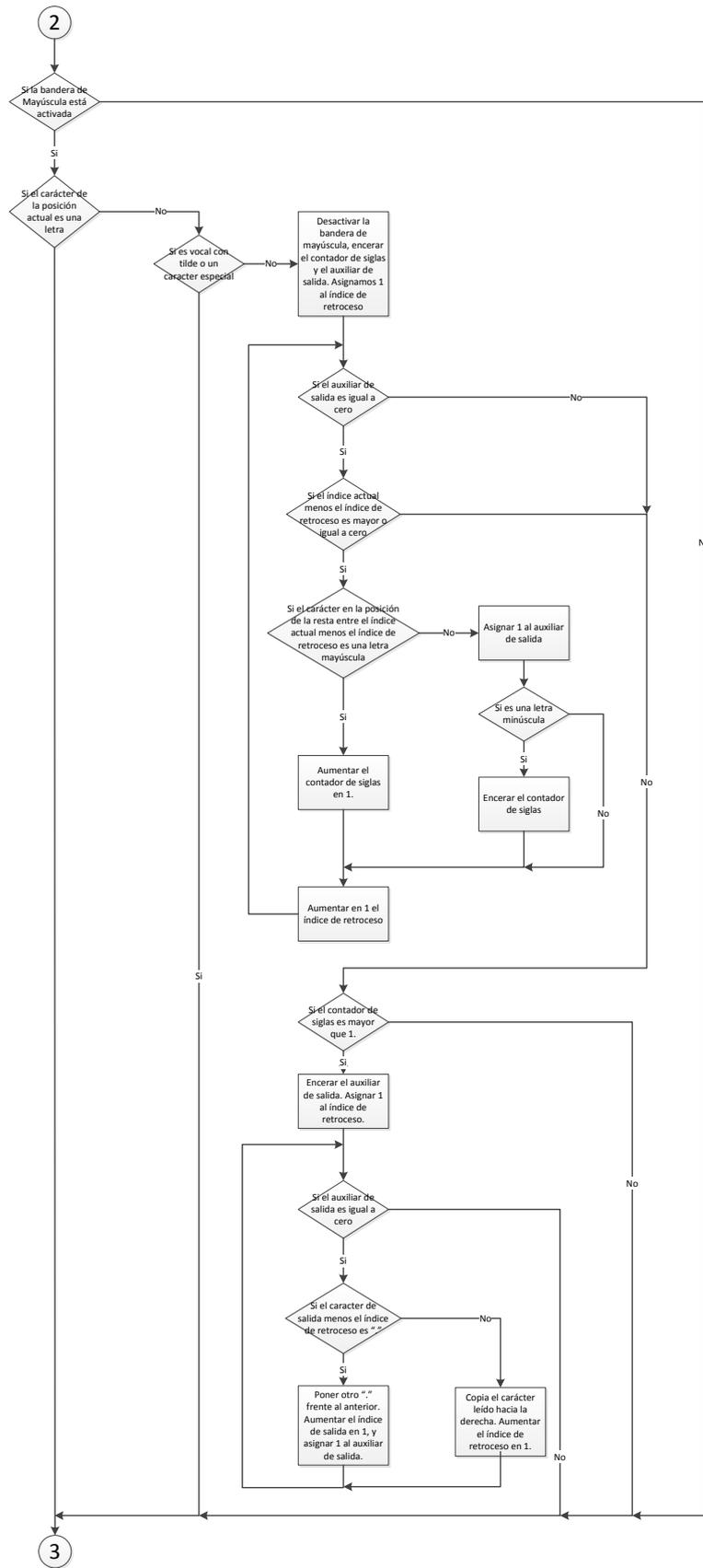


Figura 36. Diagrama de flujo de la clase ConvertirABraille, parte 2/5

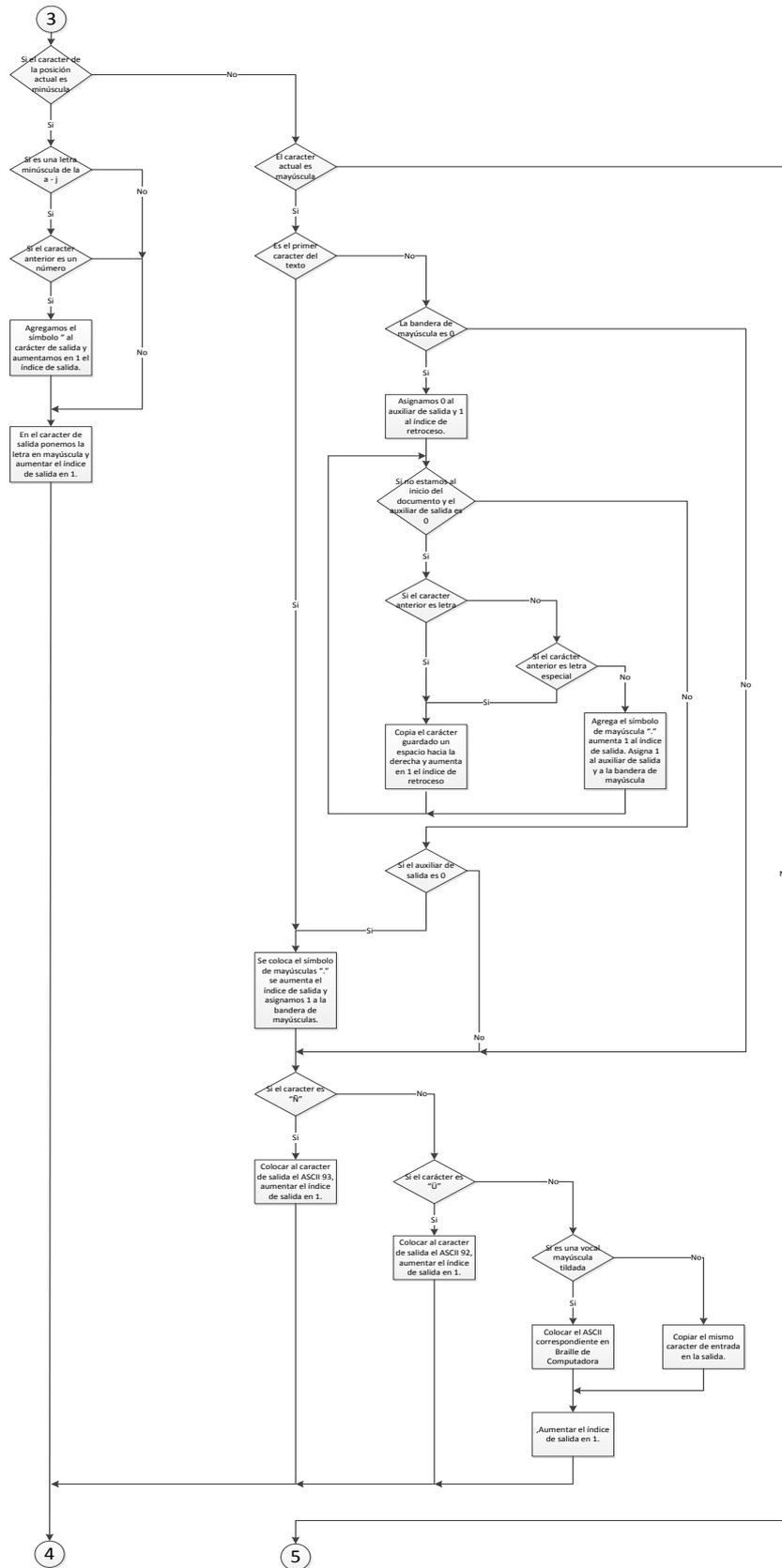


Figura 37. Diagrama de flujo de la clase ConvertirABraille, parte 3/5

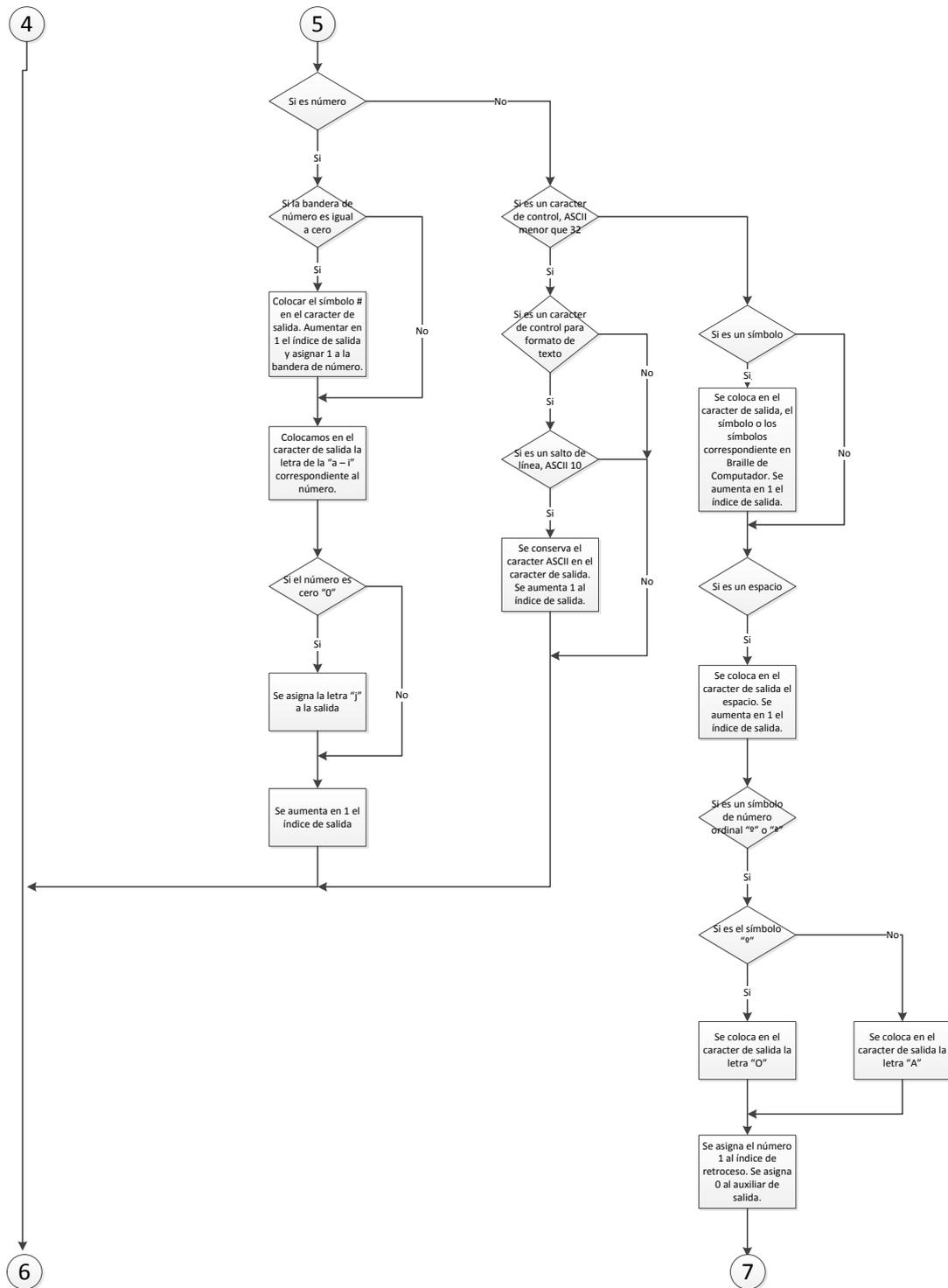


Figura 38. Diagrama de flujo de la clase ConvertirABraille, parte 4/5

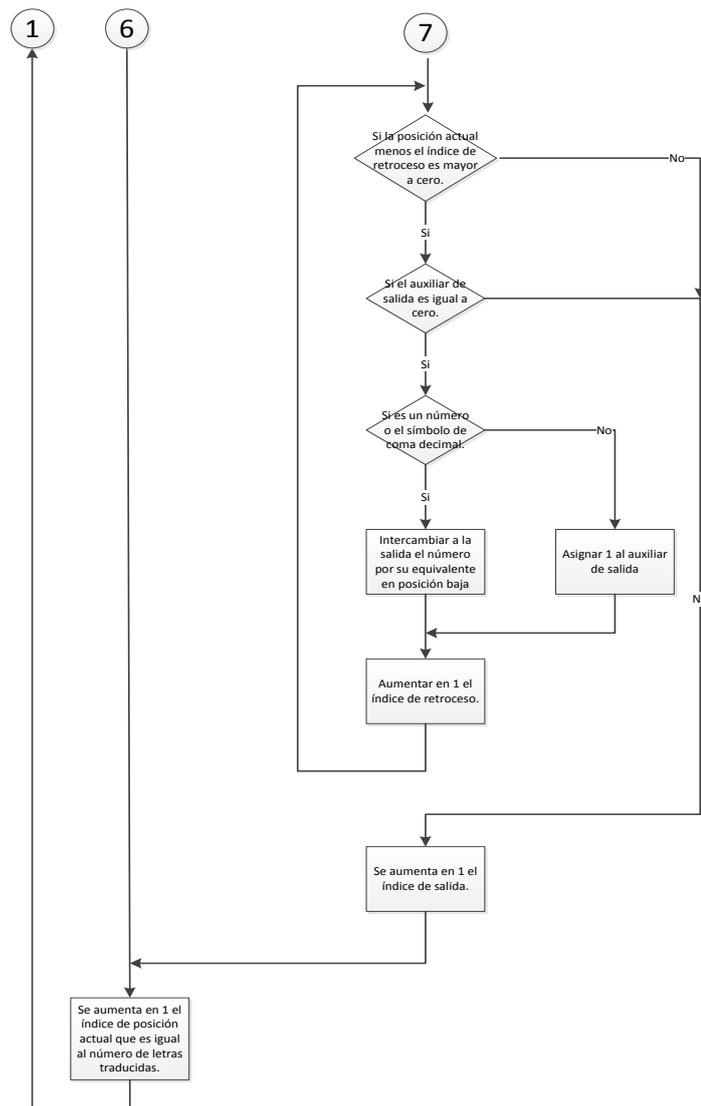


Figura 39. Diagrama de flujo de la clase ConvertirABraille, parte 5/5

GruposDe20

Esta clase permite la agrupación de texto en grupos de 20 caracteres. Dentro de esta clase se encuentran dos métodos, el método constructor de la clase y el método AGruposDe20Caracteres (String texto_in) que recibe una cadena de caracteres con el texto en Braille de Computador, y lo divide en cadenas de 20 caracteres sin cortar palabras y eliminando espacios entre líneas. Como resultado devuelve un arreglo de cadenas de 20 caracteres que contienen el texto del archivo escogido en Braille de Computador. En la Figura

40 se muestra el diagrama UML de la clase y en la Figura 41, Figura 42 y Figura 43 se muestra el diagrama de flujo.



Figura 40. UML clase GruposDe20

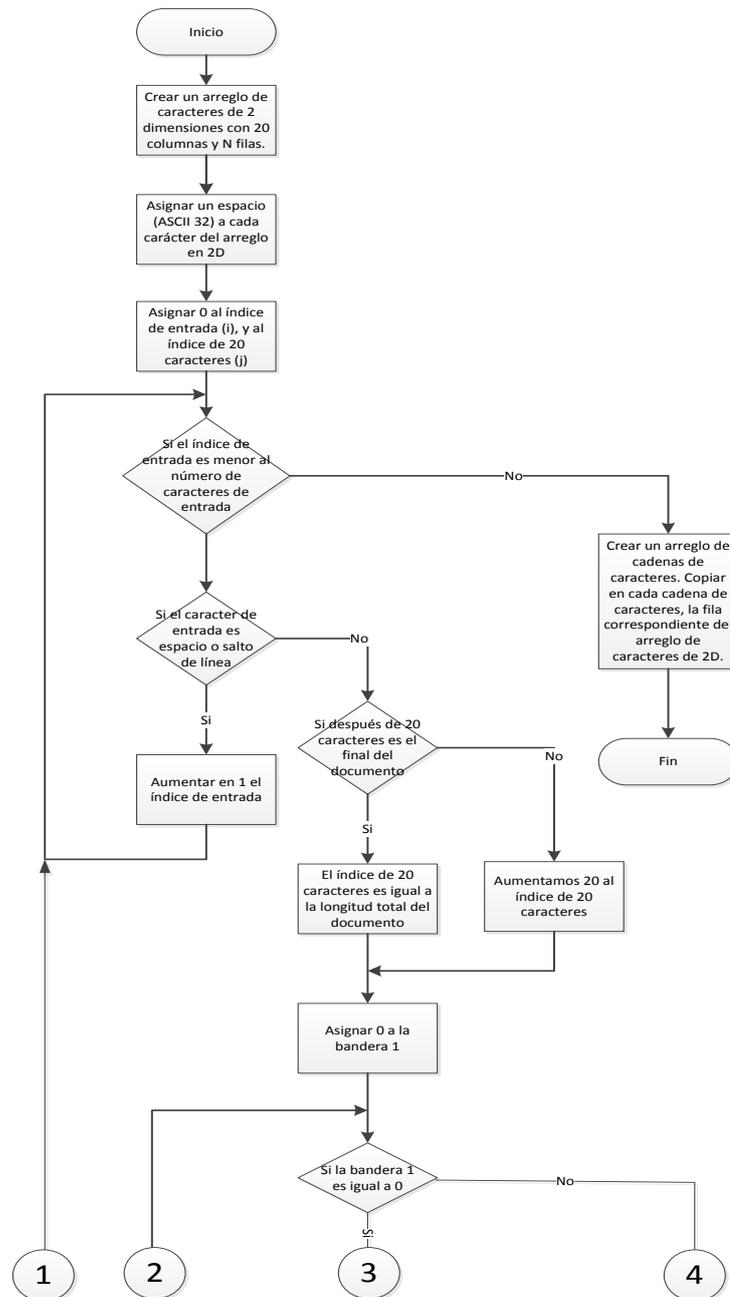


Figura 41. Diagrama de Flujo de la clase GruposDe20, parte 1/3

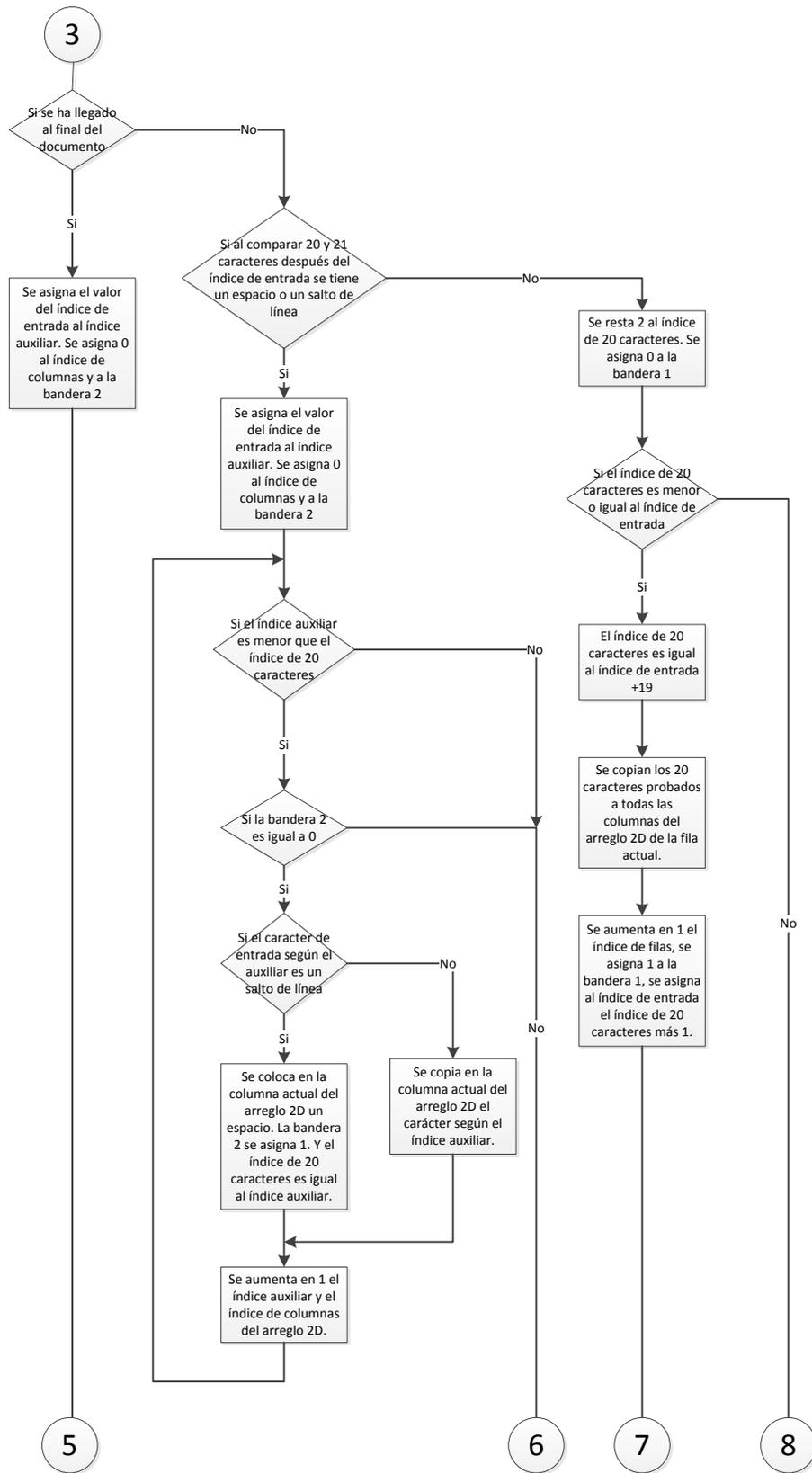


Figura 42. Diagrama de Flujo de la clase GruposDe20, parte 2/3

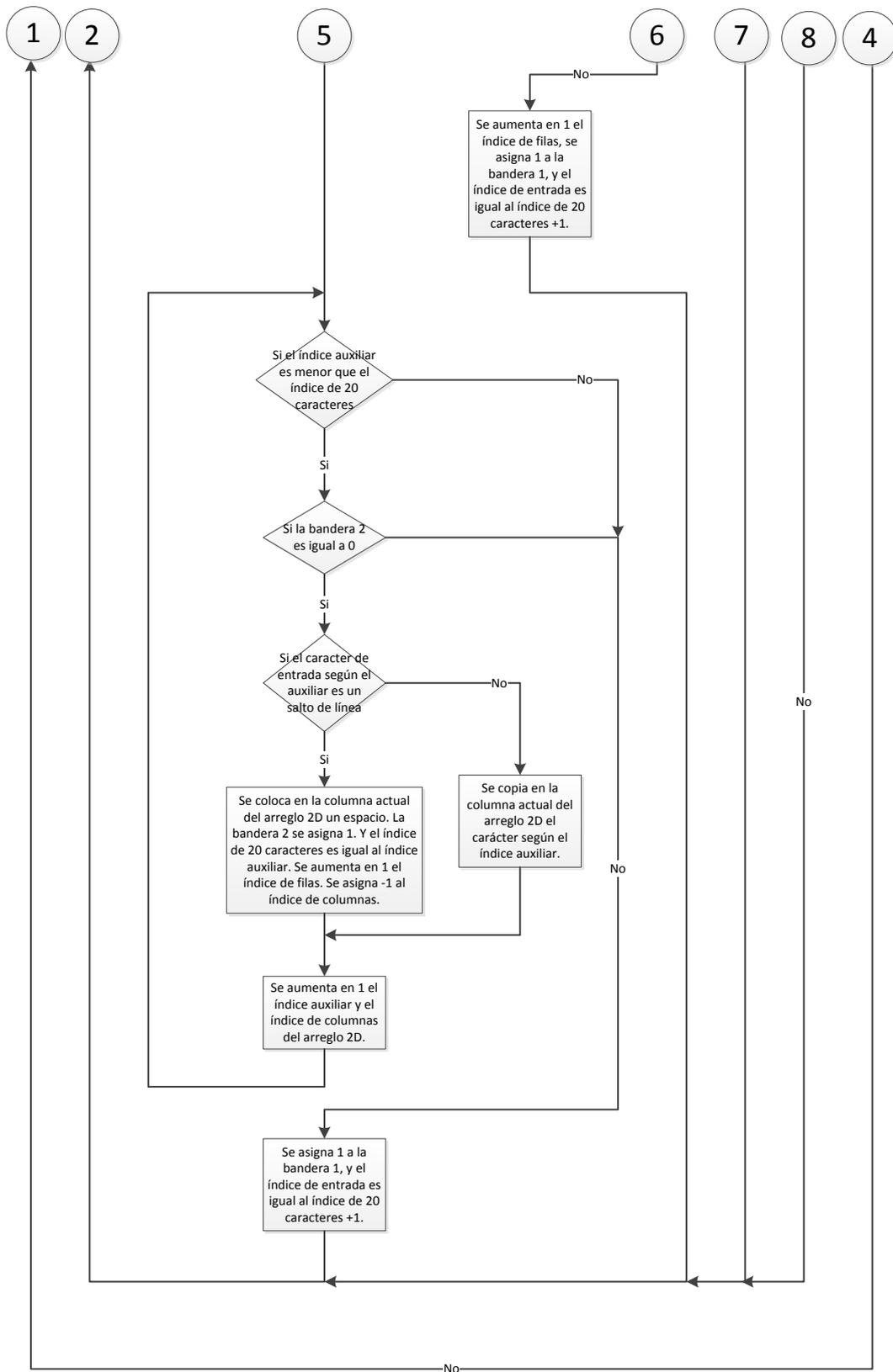


Figura 43. Diagrama de Flujo de la clase GruposDe20, parte 3/3

HiloComunicacion

Esta clase que se encarga de la comunicación del programa con el dispositivo Braille. Esta clase cuenta con los siguientes métodos:

- HiloComunicación(JTextField, File): Constructor de la clase que se encarga de iniciar la comunicación entre el software y el dispositivo, además es el encargado de disponer los puertos de comunicación libres para que el usuario escoja el puerto perteneciente al dispositivo, a partir de esto guarda el puerto escogido como puerto por defecto.
- cambiarCom(): Método que elimina el archivo guardado con el puerto de Comunicación usado por defecto, al hacer esto permite que se configure un nuevo puerto de comunicación al iniciar nuevamente el programa.
- Enviar(): Método Enviar que se encarga de enviar al dispositivo la cadena de caracteres correspondiente al archivo traducido a Braille aumentando la letra “Y” al final de la trama de cuarenta caracteres.
- fin(): Método que verifica si la comunicación con el dispositivo está activa para proceder a cerrar el puerto de comunicación.
- Reset(): Método Reset que se encarga de enviar al dispositivo una cadena de caracteres para que los pines tomen suposición inicial antes de finalizar el programa.
- run(): Método run que está en ejecución siempre con el objetivo de mantener activa la comunicación con el dispositivo para el intercambio de información.
- verificarCom(): Método que verifica si existe una comunicación activa, en tal caso consulta el puerto que se está utilizando para dicha comunicación y lo muestra al usuario.

En el diagrama de flujo de la Figura 45 se muestra el funcionamiento de esta clase; en primer lugar se ejecuta el proceso del constructor del hilo de comunicación, en el cual se determina los puertos COM habilitados del computador así como también verifica si ya existe un puerto COM almacenado

por defecto. En caso de no encontrar un puerto por defecto se despliega en un cuadro de diálogo los puertos COM habilitados para que el usuario seleccione el puerto perteneciente al dispositivo. Una vez establecido el puerto, este se guarda como el puerto COM por defecto, de esta manera cuando se vuelva a abrir el programa ya no será necesario volver a seleccionar un puerto de manera manual, en este caso el programa automáticamente trabaja con el puerto establecido en un principio.

Después de que el puerto de comunicación ha sido determinado, se procede a comprobar si es posible establecer comunicación con el dispositivo, en este caso el usuario puede trabajar con normalidad y el computador se puede comunicar con el dispositivo, caso contrario el programa entra en modo “sin conexión” donde se puede abrir archivos pero no existe la comunicación con el dispositivo. Cabe mencionar que el modo de sin conexión se puede seleccionar manualmente el momento en que se presentan las opciones de puertos de comunicación. En la Figura 44 se muestra el diagrama UML.

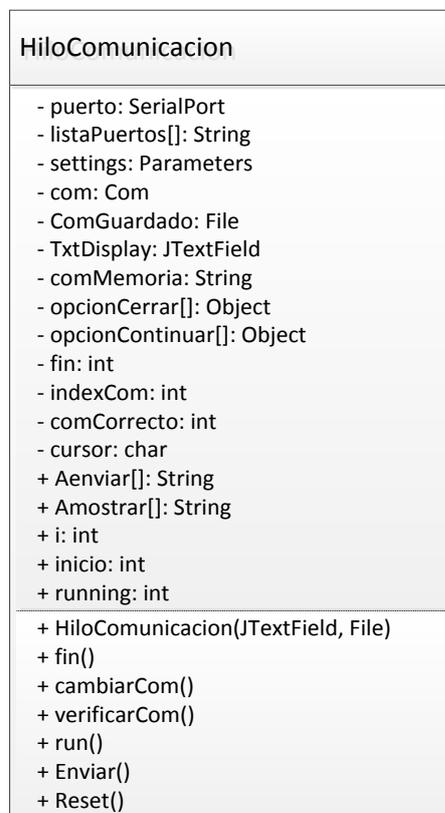


Figura 44. UML de la clase HiloComunicacion

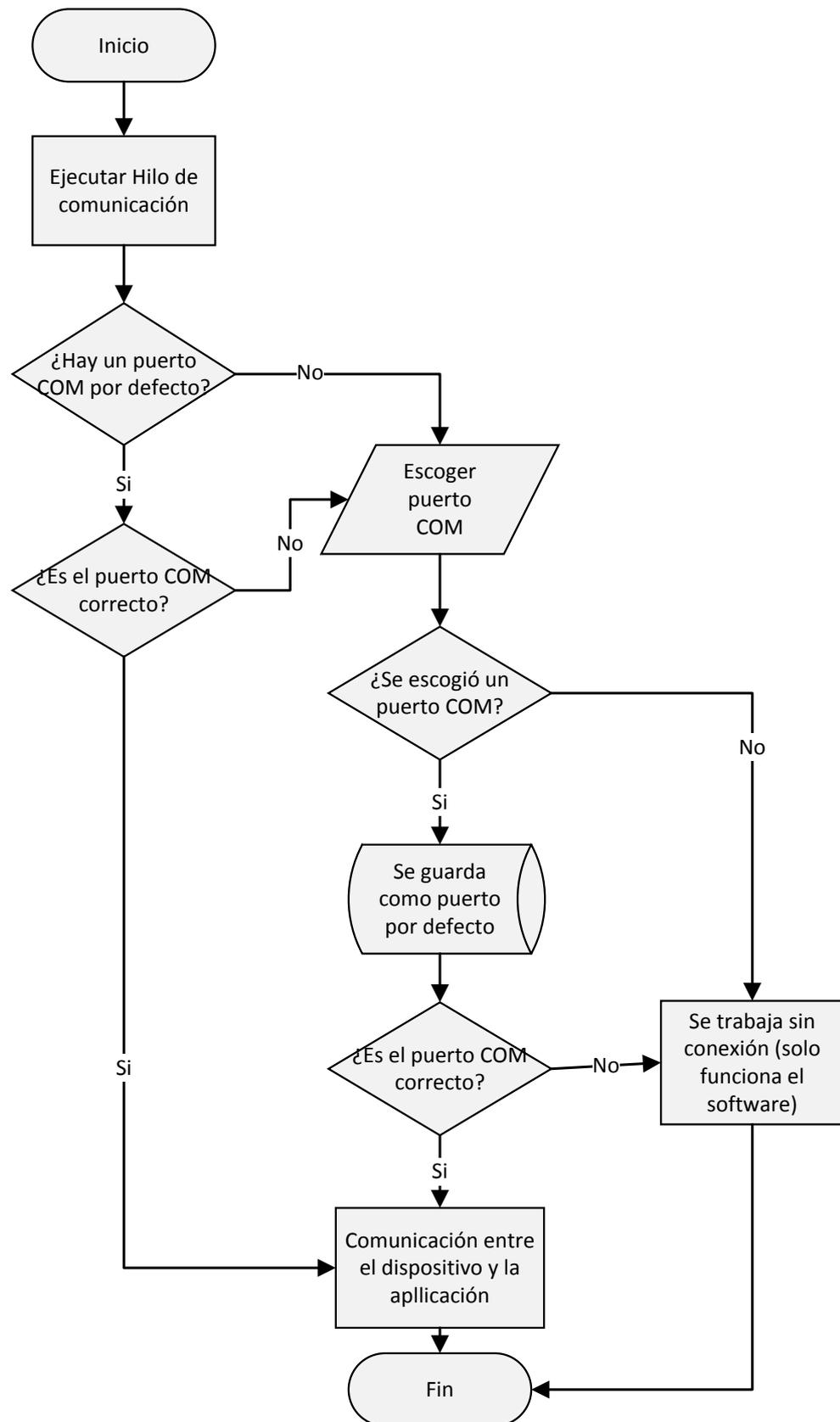


Figura 45. Diagrama de flujo de la clase HiloComunicacion

LeerArchivosGuardados

Esta clase permite almacenar y recuperar la información de los archivos abiertos como también la posición final en la lectura. En la Figura 46 se muestra el diagrama UML de la clase.



Figura 46. UML de la clase LeerArchivosGuardados

La clase contiene los siguientes métodos:

- LeerArchivosGuardados(): Constructor de la clase.
- ArchivosGuardados(String): Método que recibe la dirección del archivo en donde se ha guardado la información sobre archivos leídos con sus respectivas posiciones de lectura. Retorna un arreglo de cadenas de caracteres con la información mencionada.
- GuardarNombreArchivo(String[], String, File, int): Método que permite guardar un nuevo archivo con la información de los archivos leídos anteriormente y sus posiciones finales.

LeerPDF

Esta clase permite abrir un documento en formato .PDF. En esta clase se encuentran dos métodos, el método LeerPDF() que es el constructor de la clase y el método pdfToText(String filePathName), el cual recibe la dirección del archivo con extensión .PDF y devuelve en una cadena de caracteres el texto del archivo sin formato. En la Figura 47 se muestra el diagrama UML de la clase.

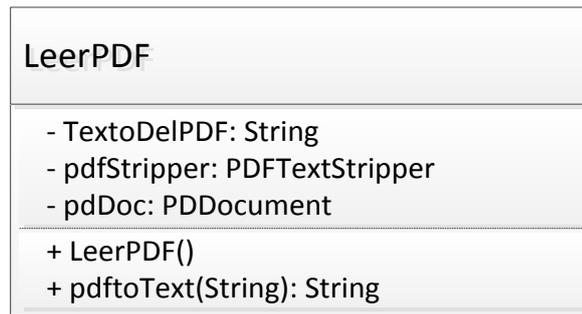


Figura 47. UML de la clase LeerPDF

LeerTXT

Esta clase permite abrir un documento de texto en formato .TXT. La clase contiene dos métodos, el primer método es LeerTXT() el cuál es el constructor de la clase, el segundo método es txttoText(String), el cual recibe la dirección del archivo tipo TXT escogido y devuelve en una cadena de caracteres el texto del archivo sin formato. En la Figura 48 se muestra el diagrama UML de la clase.

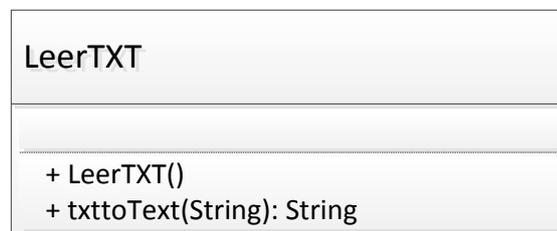


Figura 48. UML de la clase LeerTXT

PosicionDelServo

Esta clase permite determinar la posición del Servomotor según la letra en Braille.

La clase contiene dos métodos, el método constructor PosicionDelServo() y el método APosicionDelServo(String[]) que recibe un arreglo de cadenas de veinte caracteres cada una, las cuales contienen el texto del documento traducido a Braille de Computador y separado cada veinte caracteres.

Esta clase transforma cada caracter ASCII en dos letras entre la A y la H (8 posiciones). En el dispositivo cada letra Braille es representada por dos servomotores posicionados dependiendo de la letra que representan, es por eso que este método retorna un arreglo de cadenas de cuarenta caracteres.

En el Cuadro 11 se puede observar los caracteres ASCII con su equivalente pareja de letras de las ocho posiciones de servomotor, en la Figura 49 se muestra el diagrama UML y en la Figura 50 el de flujo.

Cuadro 11

Cuadro con los caracteres ASCII y la posición de los servomotores.

VALOR ASCII	CARÁCTER ASCII	POSICIÓN DEL SERVOMOTOR	
		Izquierdo	Derecho
32	Espacio	A	A
33	!	G	D
34	"	A	C
35	#	H	F
36	\$	E	D
37	%	B	D
38	&	F	D
39	'	H	A
40	(F	G
41)	G	F
42	*	B	H
43	+	H	D
44	,	A	H
45	-	H	H
46	.	A	D
47	/	H	B
48	0	H	G
49	1	C	A
50	2	G	A
51	3	C	C
52	4	C	G
53	5	C	H
54	6	G	C
55	7	G	G
56	8	G	H
57	9	H	C
58	:	B	G
59	;	A	G
60	<	E	H
61	=	F	F
62	>	H	E
63	?	B	F

Continua 

64	@	A	B
65	A	B	A
66	B	E	A
67	C	B	B
68	D	B	E
69	E	B	C
70	F	E	B
71	G	E	E
72	H	E	C
73	I	C	B
74	J	C	E
75	K	D	A
76	L	F	A
77	M	D	B
78	N	D	E
79	O	D	C
80	P	F	B
81	Q	F	E
82	R	F	C
83	S	G	B
84	T	G	E
85	U	D	H
86	V	F	H
87	W	C	F
88	X	D	D
89	Y	D	F
90	Z	D	G
91	[C	D
92	\	E	G
93]	E	F
94	^	A	E
95	_	A	F

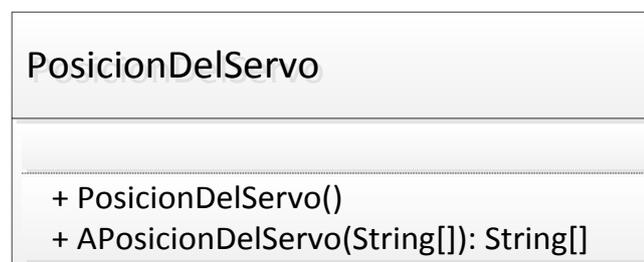


Figura 49. UML de la clase PosicionDelServo

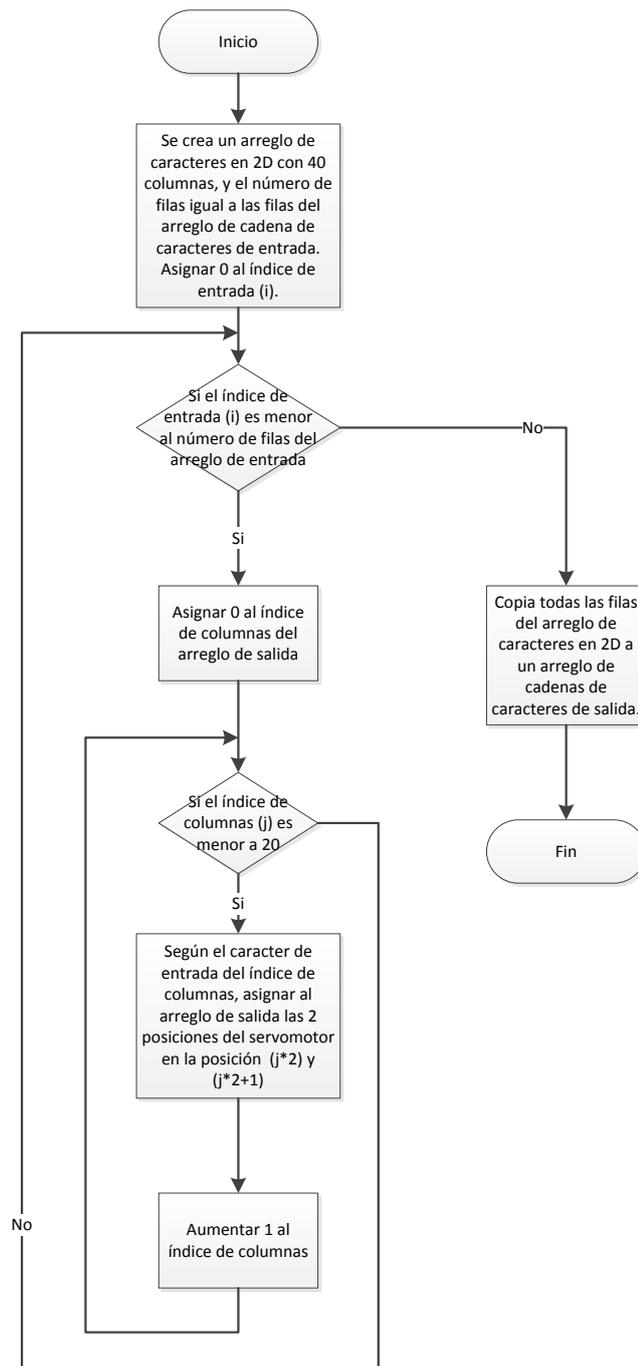


Figura 50. Diagrama de flujo de la clase PosicionDelServo

3.2.2.2 Paquete Vista

Este paquete incluye las siguientes clases:

- VentanaAcercaDe

- VentanaComputerBraille
- VentanaPrincipal

VentanaAcercaDe

Esta ventana muestra información sobre el programa, la misma que está ordenada como se indica en el diagrama esquemático de la ventana en la Figura 52 y su implementación se puede observar en la Figura 53.

La clase VentanaAcercaDe implementa los siguientes métodos:

- main(String[]): Crea y despliega la ventana.
- initComponents(): Inicializa los componentes dentro de la ventana.
- formWindowClosing(WindowEvent): Cierra la ventana creada.
- formMouseClicked(MouseEvent): Cierra la ventana al hacer click dentro de la misma.
- jTextArea1MouseClicked(MouseEvent): Cierra la ventana al hacer click dentro del cuadro de texto de la misma.
- VentanaAcercaDe(): Constructor de la clase.

En la Figura 51 se muestra el diagrama UML de la clase.



Figura 51. UML de la clase VentanaAcercaDe

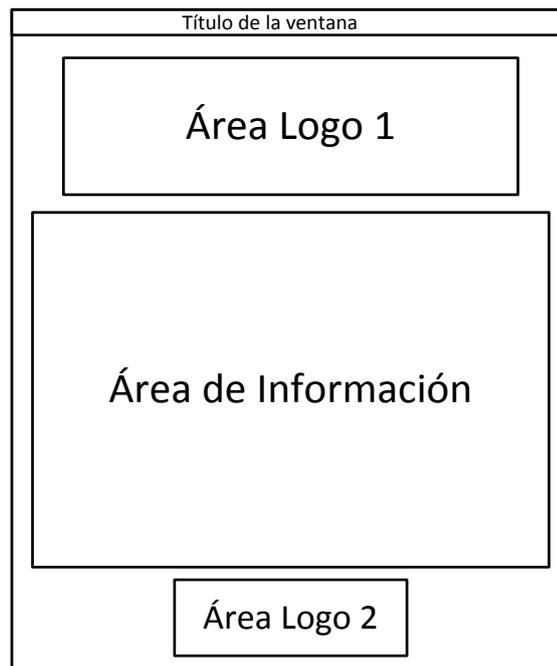


Figura 52 Esquema de la Ventana "Acerca De..."

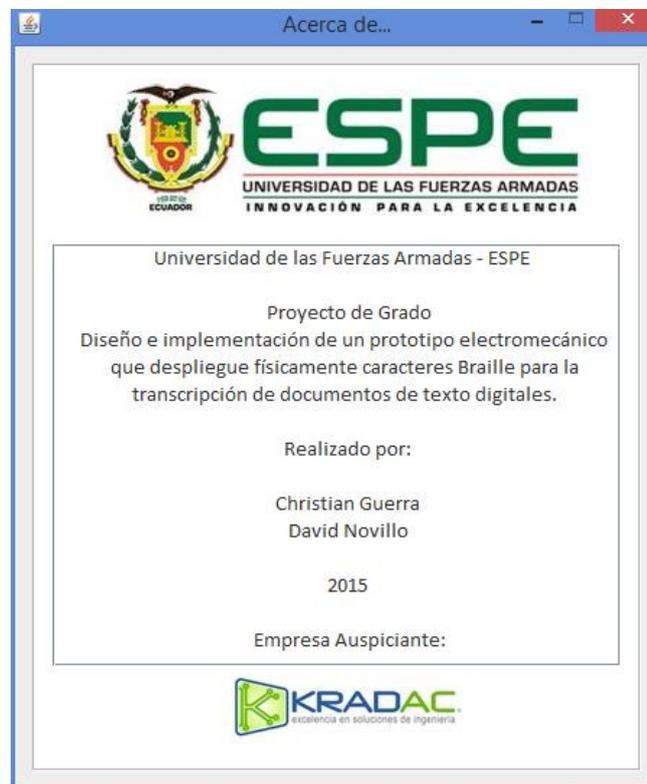


Figura 53. Implementación de la Ventana "Acerca de..."

VentanaComputerBraille

Esta ventana muestra la tabla del Braille de Computador organizada como se muestra en el esquema de la Figura 54 y su implementación en la Figura 55.

La clase VentanaComputerBraille implementa los siguientes métodos:

- `main(String[])`: Crea y despliega la ventana.
- `initComponents()`: Inicializa los componentes dentro de la ventana.
- `VentanaComputerBraille()`: Constructor de la clase.

En la Figura 56 se muestra el diagrama UML de la clase.



Figura 54 Esquema de la Ventana del Braille de Computador

Punto 6	Punto 5	Punto 4	Punto 3	Punto2	Punto1	Caracter	ASCII en decimal
0	0	0	0	0	0	ESPACIO	32
0	0	0	0	0	1	A	65
0	0	0	0	1	0	1	49
0	0	0	0	1	1	B	66
0	0	0	1	0	0	'	39
0	0	0	1	0	1	K	75
0	0	0	1	1	0	2	50
0	0	0	1	1	1	L	76
0	0	1	0	0	0	@	64
0	0	1	0	0	1	C	67
0	0	1	0	1	0	I	73
0	0	1	0	1	1	F	70
0	0	1	1	0	0	/	47
0	0	1	1	0	1	M	77
0	0	1	1	1	0	S	83
0	0	1	1	1	1	P	80
0	1	0	0	0	0	"	34
0	1	0	0	0	1	E	69
0	1	0	0	1	0	3	51
0	1	0	0	1	1	H	72
0	1	0	1	0	0	9	57
0	1	0	1	0	1	O	79
0	1	0	1	1	0	6	54
0	1	0	1	1	1	R	82
0	1	1	0	0	0	^	94
0	1	1	0	0	1	D	68
0	1	1	0	1	0	J	74
0	1	1	0	1	1	G	71
0	1	1	1	0	0	>	62
0	1	1	1	0	1	N	78
0	1	1	1	1	0	T	84
0	1	1	1	1	1	Q	81
1	0	0	0	0	0	,	44

Figura 55. Implementación de la Ventana del Braille de Computador

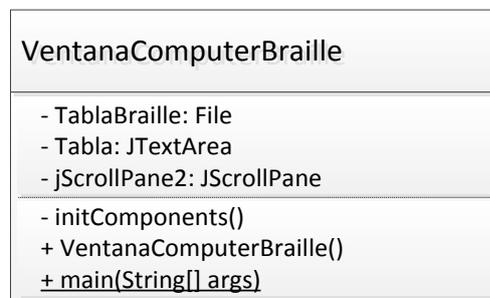


Figura 56. UML de la clase VentanaComputerBraille

VentanaPrincipal

Esta ventana es la interfaz entre el usuario y el computador para la selección del archivo que se requiera mostrar en el dispositivo Braille. En la Figura 57 se observa la organización de la ventana, y en la Figura 58 como está implementada. Después de escoger un texto para mostrar, la ventana

cambia para que el área del texto sea más grande y que bajo el botón de Abrir se muestre la línea, en Braille de Computador, que está siendo mostrada en el dispositivo, esta disposición de la ventana se la observa en la Figura 59, y su implementación en la Figura 60.

En esta clase también se realiza el proceso para mostrar un archivo de texto en el dispositivo Braille, para esto son implementados los siguientes métodos:

- `main(String[])`: Crea y despliega la ventana.
- `initComponents()`: Inicializa los componentes dentro de la ventana.
- `VentanaPrincipal()`: Constructor que inicializa las variables e inicia la comunicación con el dispositivo.
- `formWindowClosing(WindowEvent)`: Método que se ejecuta cuando la ventana se cierra y llama al método `CerrarPrograma`.
- `menuOpenActionPerformed(ActionEvent)`: Opción Abrir en el menú Archivo.
- `menuSalirActionPerformed(ActionEvent)`: Opción Salir en el menú Archivo.
- `menuConfigurarPuertoActionPerformed(ActionEvent)`: Opción Configurar puerto en el menú Configuración, elimina el puerto de comunicación guardado para que la siguiente vez que se abra el programa se deba configurar un nuevo puerto.
- `menuVerificarActionPerformed(ActionEvent)`: Opción Verificar Puerto en el menú Configuración, permite ver el puerto de comunicación actualmente usado en el programa.
- `menuAcercaDeActionPerformed(ActionEvent)`: Opción Acerca De en el menú Ayuda con la información del programa.
- `OpenActionPerformed(ActionEvent)`: Botón Abrir en la interfaz de usuario para elegir un nuevo archivo.
- `menuComputerBrailleActionPerformed(ActionEvent)`: Opción para abrir la tabla de Braille de Computador en el menú Ayuda.

- `menuManualUsuarioActionPerformed(ActionEvent)`: Opción para abrir el Manual de Usuario del programa.
- `AbrirArchivo()`: Método `AbrirArchivo` que permite escoger un nuevo archivo a ser mostrado en el dispositivo Braille.
- `CerrarPrograma()`: Método que ejecuta un proceso para el cierre del programa, dejando el dispositivo inicializado y guarda los archivos usados.

En la Figura 61 se muestra el diagrama UML de la clase.

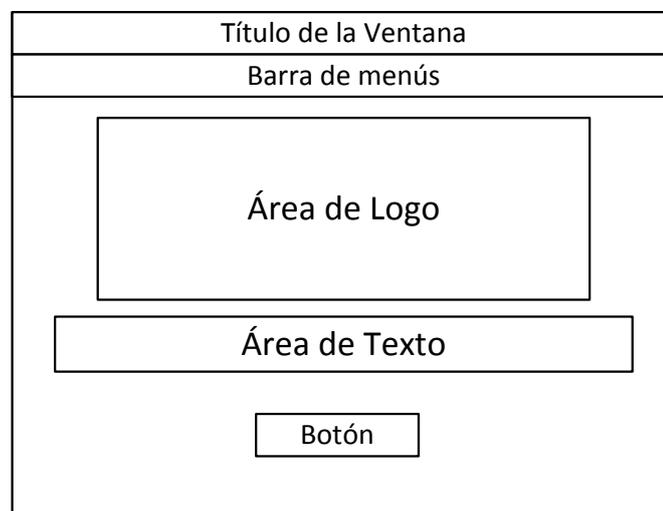


Figura 57. Esquema de la Ventana Principal



Figura 58. Implementación de la Ventana Principal

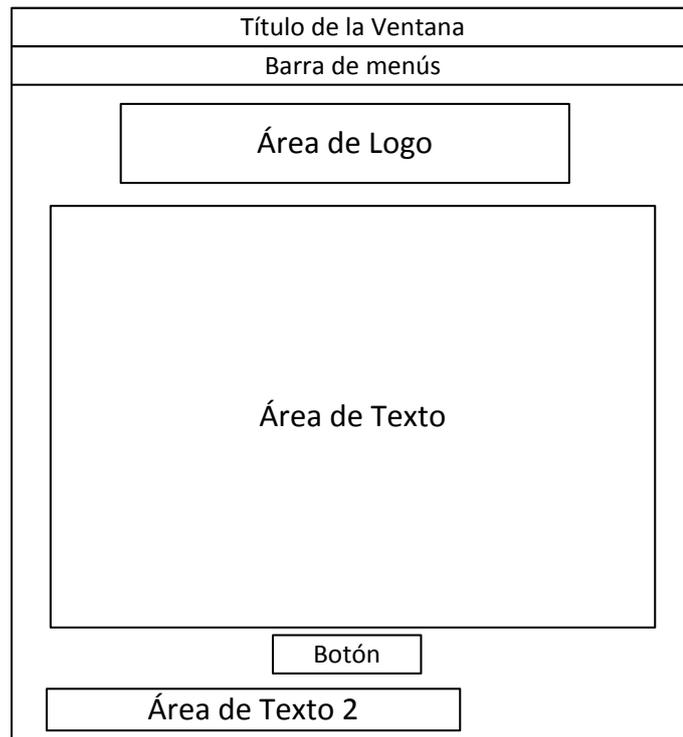


Figura 59. Esquema de la Ventana Principal al abrir un archivo.

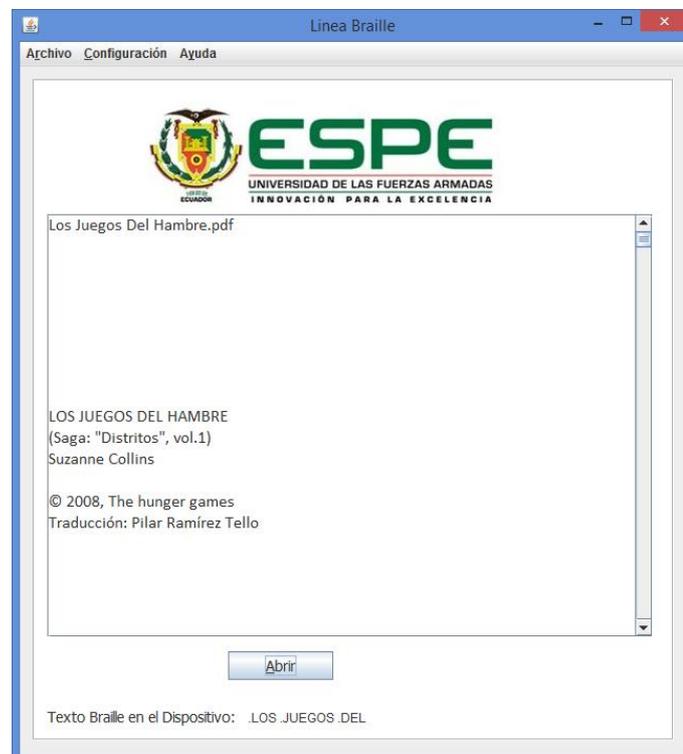


Figura 60. Implementación de la Ventana Principal al abrir un archivo.

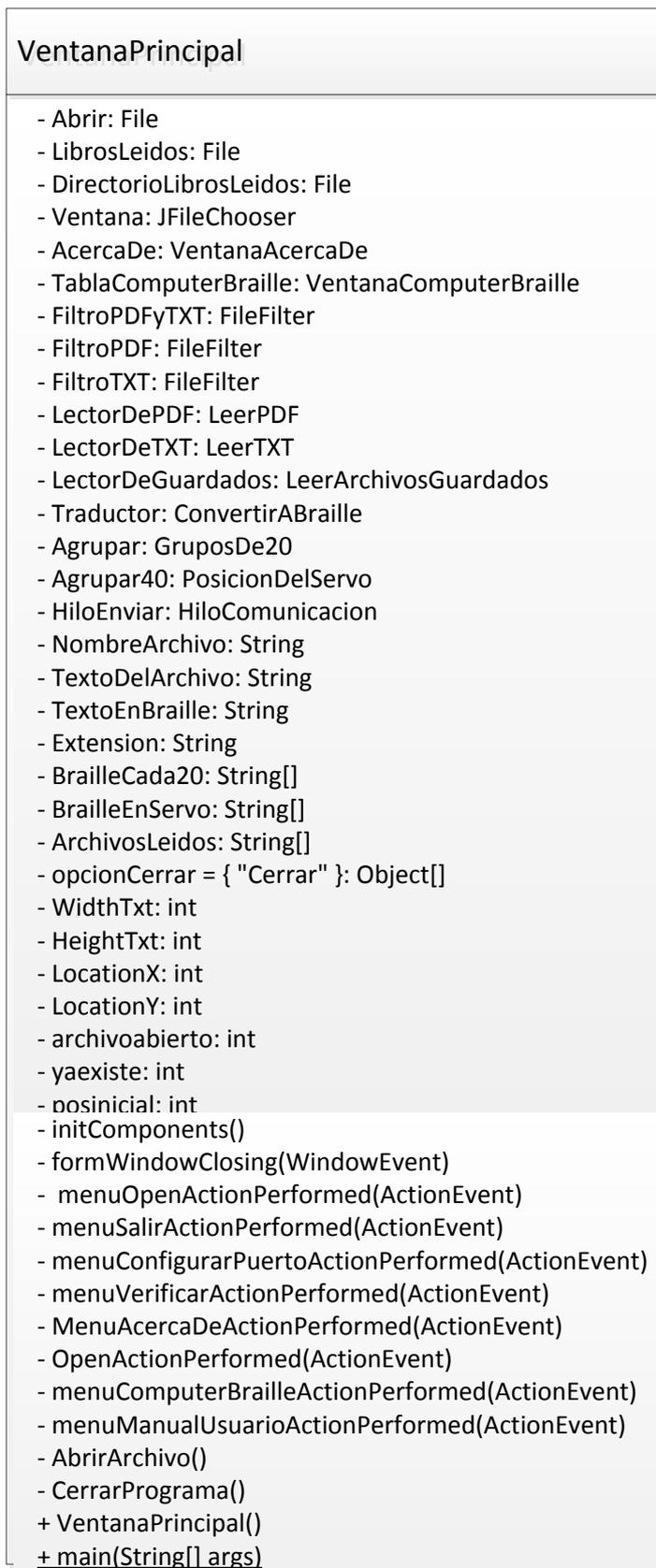


Figura 61. UML de la clase VentanaPrincipal

3.2.2.3 Paquete Controlador

Este paquete incluye la clase principal del programa, la cual es la que inicia la ejecución del programa.

ProgramaPrincipal

Esta es la clase principal del proyecto en Java. Al abrir el proyecto se ejecuta el método main de la clase, dentro del cual se crea una instancia de la clase VentanaPrincipal y se muestra la interfaz principal del programa. En la Figura 62 se muestra el diagrama UML de la clase.

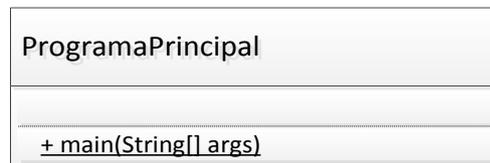


Figura 62. UML de la clase Principal.

3.2.2.4 Librerías

En el proyecto se utiliza dos librerías externas, la librería Apache PDFBox para el manejo de los documentos en formato .PDF y la librería Giovynet Driver para la comunicación serial entre el dispositivo y el computador.

3.2.2.4.1 Librería Apache PDFBox

La librería Apache PDFBox es una herramienta Java de código abierto para trabajar con documentos en formato PDF. Esta librería está disponible para su uso y distribución de manera gratuita desde la página web: <http://pdfbox.apache.org>

PDF es el acrónimo de Portable Document Format (Formato de Documento Portátil), término que identifica una modalidad para el almacenamiento de archivos digitales. Este formato se ha convertido en un

estándar de hecho para la producción de documentos académicos, publicaciones y notas de prensa. (Definicion.de, 2014)

La librería permite la creación de nuevos documentos PDF y la manipulación de documentos existentes con la habilidad de extraer el contenido de los documentos. Además provee varias utilidades ejecutables desde línea de comandos, entre las funciones que presenta la librería están las siguientes:

- Extracción de Texto: Extrae el texto dentro de un PDF para su uso en otras aplicaciones.
- Fusión y División: Une varios PDFs en uno solo, o separa un PDF en varios.
- Llenado de Formas: Extrae los datos de formas de una forma en PDF o llena una forma en PDF.
- Validación de PDF/A: Valida PDFs con el estándar ISO PDF/A.
- Impresión de PDF: Imprime un archivo PDF con una impresora que soporte el API de impresión de Java.
- Conversión de PDF a Imagen: Convierte un PDF a imágenes.
- Creación de PDF: Crea un PDF desde cero.
- Firma de PDF: Firma un documento en PDF.

En el desarrollo del proyecto se requiere la utilización de la herramienta de extracción de texto para su uso dentro de la aplicación. Esta herramienta está compuesta por varias clases proporcionadas por la librería de las cuales se implementan dos de estas clases, PDDocument y PDFTextStripper.

PDDocument

Esta clase es la representación en memoria de un documento PDF. Al crear un objeto de esta clase se crea un nuevo documento PDF sin páginas, y a través de los métodos proporcionados por la clase se puede modificar el documento o cargar un documento ya existente en el objeto.

El método load es el encargado de cargar un archivo PDF existente en el objeto de tipo PDDocument. Para el funcionamiento de este método es necesario proporcionar el URL con la ubicación del archivo de tipo PDF; como resultado el objeto PDDocument contiene en memoria el PDF escogido.

PDFTextStripper

Esta clase toma un documento PDF y separa todo el texto ignorando formatos del archivo, para esto el usuario deberá verificar que se tenga los permisos correspondientes para extraer el texto del documento.

Con la creación del objeto de esta clase se puede acceder al método getText, el cual recibe un objeto de la clase PDDocument que contiene el documento PDF escogido con anterioridad y retorna un String con el texto plano del archivo PDF para ser tratado en el programa. Es importante recalcar que el PDF no debe estar encriptado al ser tratado por este método.

3.2.2.4.2 Librería Giovynet Driver

La librería Giovynet Driver es un sistema que permite el uso de Java para crear aplicaciones que puedan comunicar circuitos externos con el computador. Esta librería proporciona instrucciones para manejar los dispositivos GIV8DAQ y GIV4R, además soporta el envío y recepción de caracteres ASCII a través del puerto RS-232. Esta librería está disponible para su uso y distribución de manera gratuita desde la página web: <http://www.giovynet.com/>

Para el desarrollo del proyecto se ha elegido la comunicación usando el puerto RS-232, para lo cual se utiliza las siguientes clases:

Clase SerialPort

La clase Serial Port pertenece al paquete giovynet.nativelink de la librería Giovynet, y se encarga de enlazar los métodos para el manejo de la

comunicación serial. Al crear un objeto de esta clase se tiene acceso a los métodos para la identificación de puertos seriales.

El método `getFreeSerialPort` es utilizado en la aplicación para obtener los puertos seriales libres que se tiene en el computador para posteriormente escoger uno de ellos para la comunicación.

Enum Baud

Es un tipo especial de dato que enumera varias constantes que representan la tasa de Baudios para la comunicación serial. Este dato es requerido para la comunicación del programa.

Clase Parameters

Esta clase permite crear objetos que contengan los parámetros para la comunicación serial. En el desarrollo de la aplicación se utilizan los métodos `setBaudRate` y `setPort`.

El método `setBaudRate` recibe un dato de tipo `Baud` con la velocidad en baudios de la comunicación serial.

El método `setPort` recibe un `String` con el nombre del puerto que se ha escogido para la comunicación, por defecto se asigna el puerto `COM1`.

Clase Com

Esta clase representa un puerto serial. Al crear un objeto de esta clase es necesario enviar los parámetros para la comunicación serial, esto se hace enviando al constructor un objeto de la clase `Parameters` el cual define el puerto que se utilizará y la velocidad de comunicación.

A partir de la creación del objeto de la clase se abre la comunicación y se puede utilizar los métodos de la clase para el envío y recepción de datos. La

aplicación desarrollada utiliza los métodos `sendString` y `receiveSingleChar` para la comunicación con el microcontrolador.

El método `sendString` recibe un `String` del programa y lo envía al dispositivo. Este método ha sido escogido para la implementación en la aplicación puesto que antes de enviar los datos se cuenta con un `String` como resultado final de la conversión a Braille y la codificación en posiciones de servomotor.

El método `receiveSingleChar` recibe a través del puerto serial un solo carácter ASCII, el cuál es enviado como resultado del método para su uso en la aplicación. Este método ha sido escogido para recibir instrucciones simples del dispositivo para la navegación dentro del documento abierto.

Por último, al terminar de usar la comunicación serial es necesario llamar al método `close`, el cual cierra el puerto que se creó y lo deja libre para el uso en otra ocasión. Si este método no es usado, el puerto quedará abierto a pesar de que se haya cerrado el programa, y no podrá ser usado por ninguna aplicación sin que se lo haya cerrado previamente.

3.2.3 Programación del Microcontrolador

CCS-Compiler es el entorno de desarrollo escogido para realizar el programa para el microcontrolador PIC18F2550, en el encabezado del programa se establecen las funciones que se utilizan en la programación del microcontrolador. Las funciones a utilizar son las siguientes:

- Tres puertos de comunicación serial.
- Cuatro entradas digitales para los botones que maneja el usuario.

Para la comunicación con el computador se maneja un buffer de 41 caracteres donde se almacenan los bytes que envía el computador. Cada byte representa una posición para un servomotor en específico y además recibe un carácter adicional de verificación al final de cada trama.

3.2.3.1 Descripción del programa principal

El programa principal recibe una trama de 41 caracteres enviada por el computador, cada caracter representa una posición de un servomotor y es almacenado en una posición del buffer. Una vez almacenadas las posiciones se procede a enviar la información a los dos controladores de servomotores.

El valor almacenado en la posición 0 del buffer corresponde al canal 1 del primer controlador, se asignan los valores restantes del buffer de manera ascendente hasta la posición 19, correspondiente al canal 20. Para el segundo controlador la asignación inicia en la posición 20 del buffer correspondiente al canal 1 de este controlador, y se finaliza al llegar a la posición 39 correspondiente al canal 20.

El caracter almacenado en la posición 40 del buffer permite verificar que se ha recibido una trama completa. Este caracter es representado por la letra Y.

Los caracteres recibidos son letras en mayúscula entre la A y la H, los cuales representan ocho posiciones del servomotor. Según estas letras recibidas, se envían las tramas correspondientes a cada controlador con las posiciones que los servomotores deben tomar para formar la letra correspondiente en Braille.

Para la navegación en el documento existen cuatro botones que envían señales al microcontrolador. En el programa principal se leen estas señales como entradas digitales y según el botón presionado se envía de manera serial al computador una letra en minúscula que representa las acciones: avanzar (‘a’), retroceder (‘b’), inicio (‘c’) y fin (‘d’).

En la Figura 63 y Figura 64 se muestran el diagrama de flujo del funcionamiento del programa principal, y el proceso realizado para convertir un dato en una posición de servomotor, respectivamente.

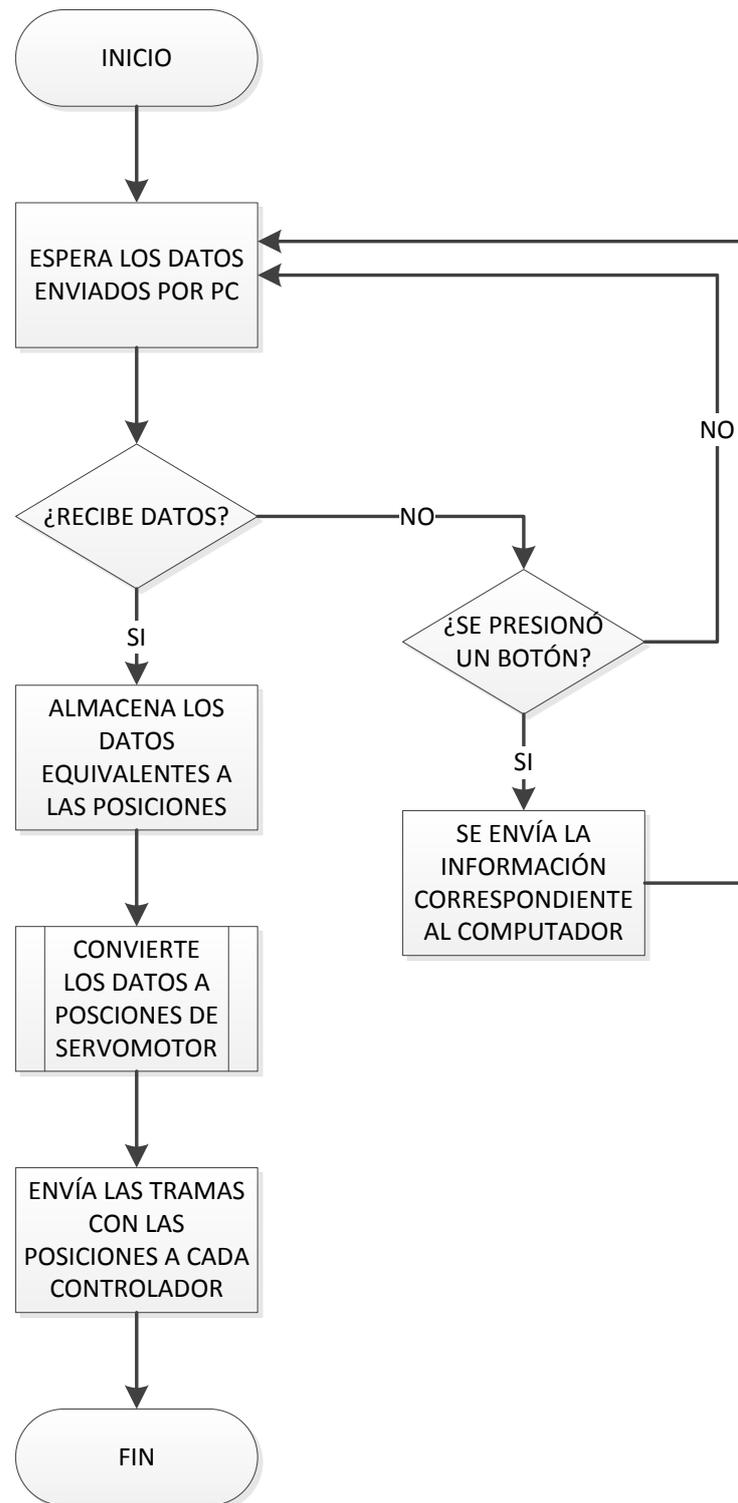


Figura 63. Diagrama de flujo del programa del microcontrolador

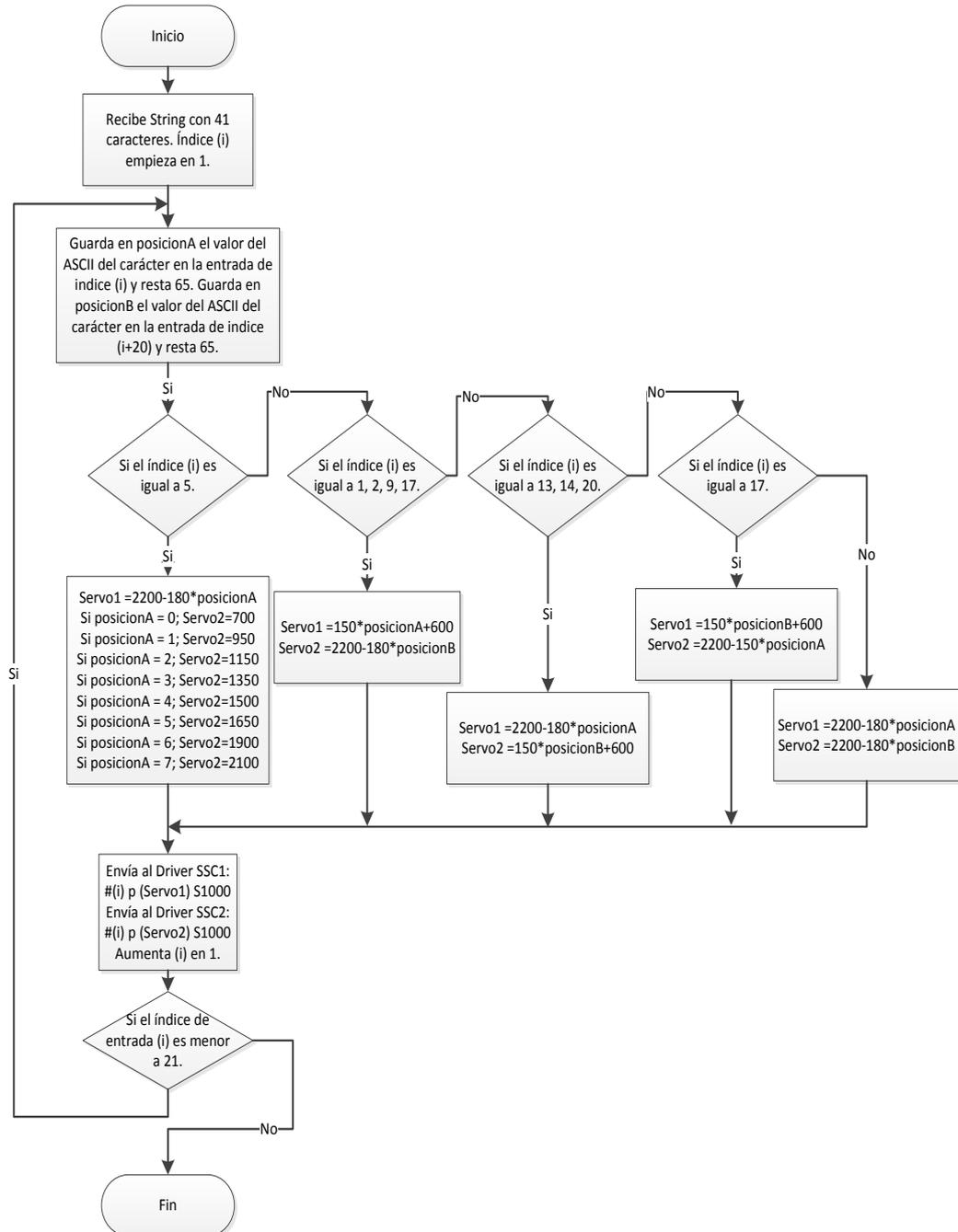


Figura 64. Diagrama de Flujo del proceso para convertir un dato en una posición de servomotor.

3.3 Diseño mecánico

El diseño de la parte mecánica se compone de las siguientes piezas:

- Celdas Braille
- Bases para los servomotores
- Piñones sujetos a los servomotores
- Cremalleras con adaptación para el levantamiento de pines
- Pines para formar los caracteres Braille
- Caja del prototipo

3.3.1 Celdas Braille

El diseño de las veinte celdas Braille se realizó de manera que todas las celdas conformen una sola pieza, como se observa en la Figura 65.

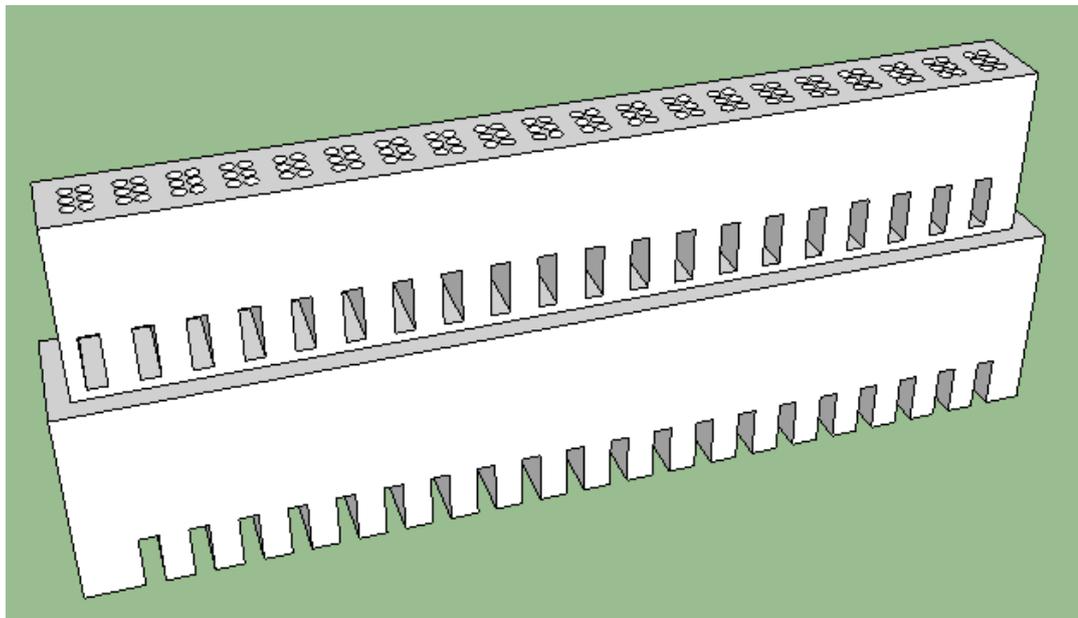


Figura 65. Celdas Braille

El diseño está realizado de modo que los servomotores sean dispuestos en dos niveles, uno superior y uno inferior de manera que se aproveche el espacio y el dispositivo no tenga una gran dimensión.

Las dimensiones de las celdas (principalmente la altura) están establecidas de acuerdo a las dimensiones de los servomotores que se utiliza (19 mm x 8 mm x 17.5 mm). En la Figura 66, Figura 67 y Figura 68 se muestran las dimensiones en milímetros de la pieza que conforma las celdas Braille.

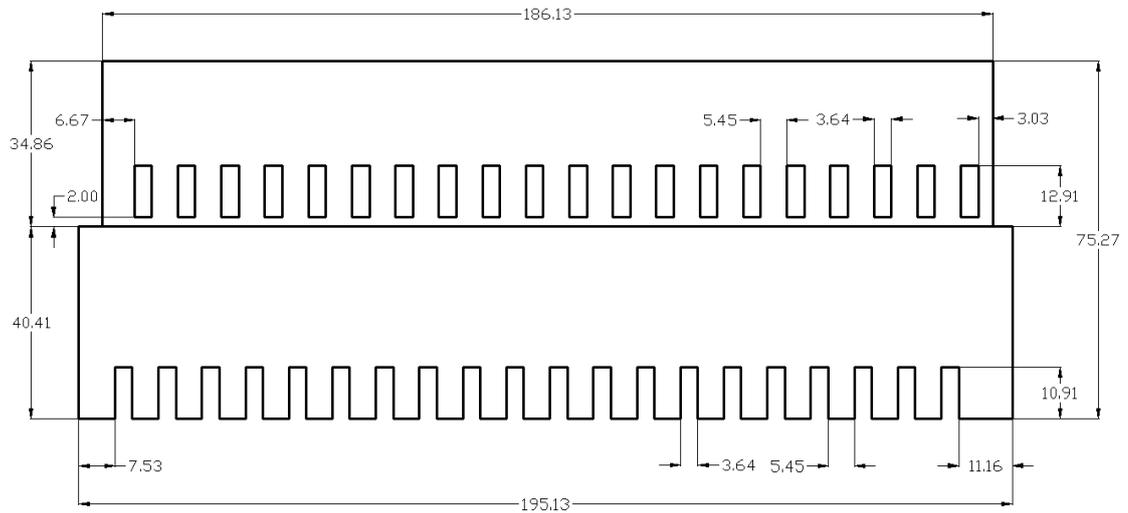


Figura 66. Vista frontal de las celdas (dimensiones en milímetros)

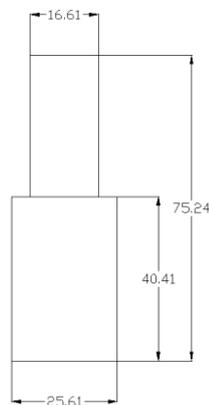


Figura 67. Vista lateral de las celdas (dimensiones en milímetros)

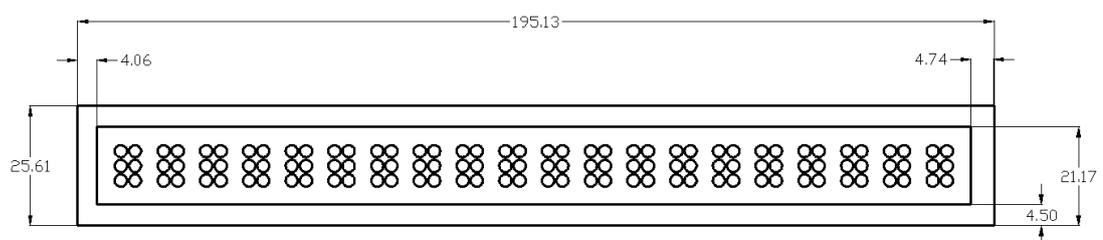


Figura 68. Vista superior de las celdas (dimensiones en milímetros)

En el diseño de las celdas Braille se tomó en cuenta la máxima tolerancia permitida para las dimensiones de las celdas, de acuerdo a los estándares establecidos por las Guías de la Comisión Braille Española, pero debido a las limitaciones que existen en cuanto al desarrollo de la pieza mediante impresión en tres dimensiones y la rigidez del material (plástico ABS), algunas dimensiones fueron ligeramente modificadas más allá de la tolerancia establecida, sin embargo al ser un cambio mínimo no representa dificultad para el usuario al momento de leer el texto. En la Figura 69 se observa las dimensiones en milímetros con las que fue diseñada la celda Braille.

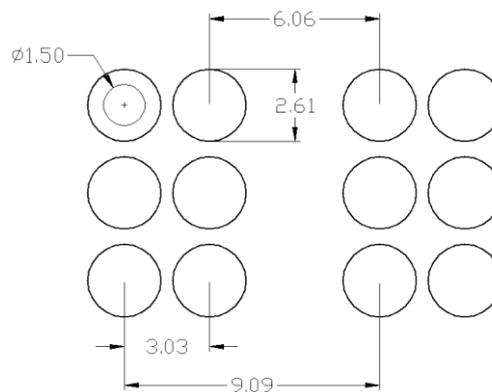


Figura 69. Dimensiones de la celda en milímetros

Como se observa en la Figura 69, el punto se mantiene dentro del estándar mientras que las demás dimensiones están con una diferencia entre 0.5 mm y 1.5 mm de la máxima medida establecida. Al ser una variación minúscula el texto sigue siendo legible para el usuario.

3.3.2 Bases para los servomotores

Las bases para los servomotores son superficies donde se ubican los servomotores de manera que no intervengan en el movimiento de las cremalleras, tanto en el nivel superior de los servomotores como en el inferior como se observa en la Figura 70.

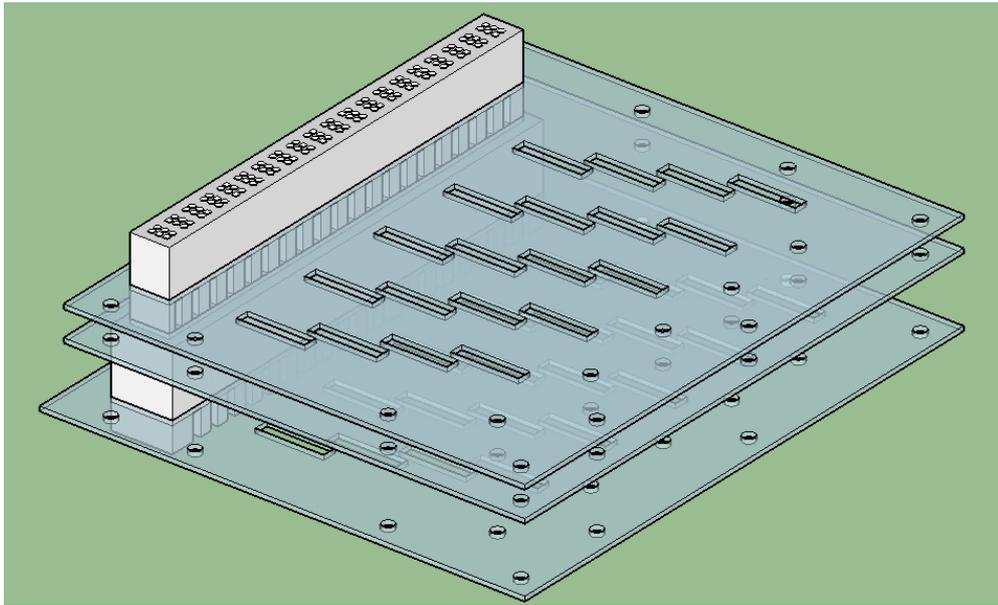


Figura 70. Bases para servomotores

Estas bases tiene 2 mm de espesor y sus dimensiones se muestran en la Figura 71, Figura 72 y Figura 73. Los espacios rectangulares son utilizados para que el piñón sobresalga y se acople a la cremallera.

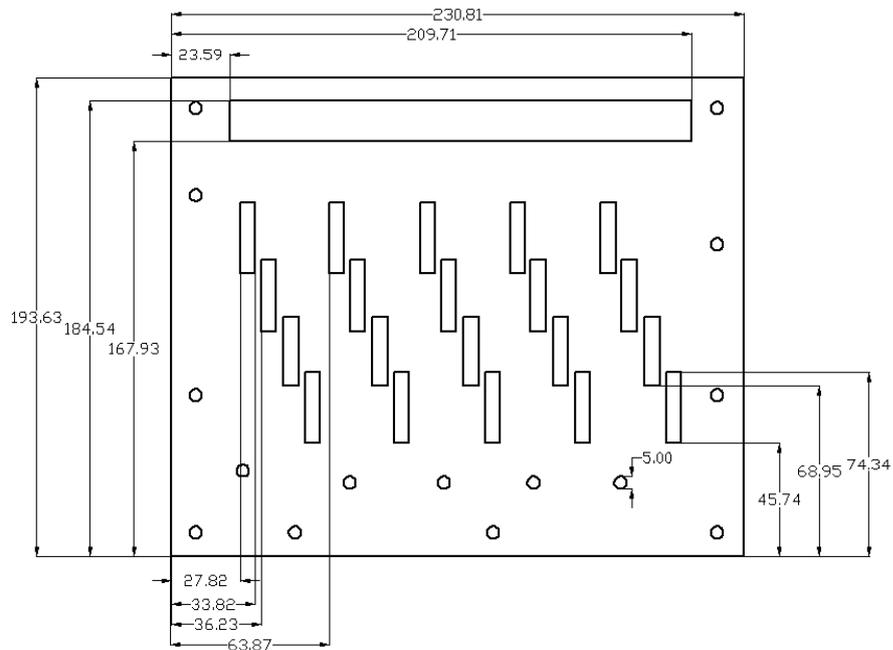


Figura 71. Base superior de servomotores (dimensiones en milímetros)

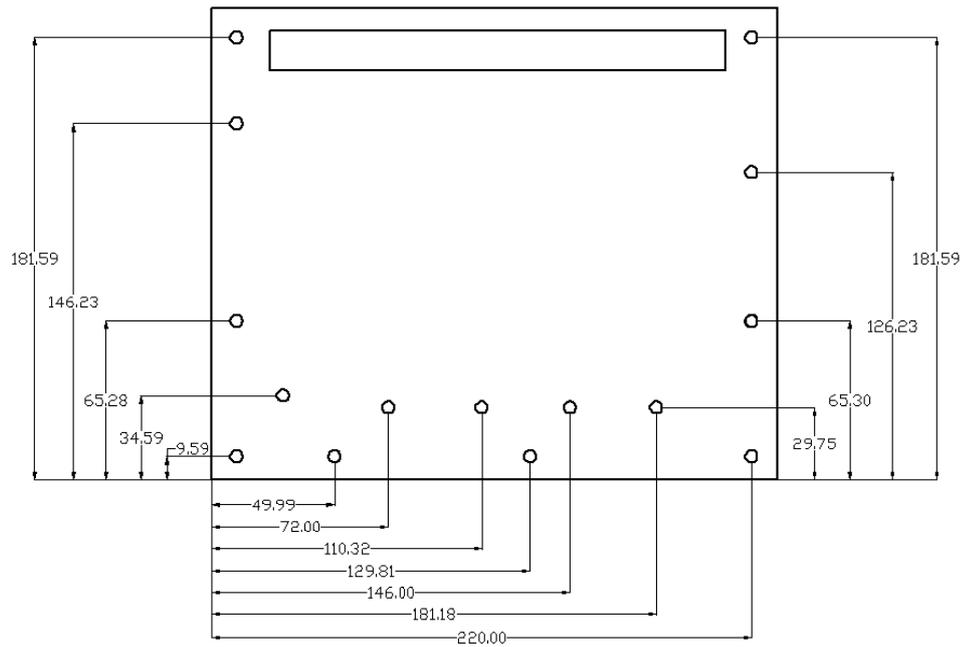


Figura 72. Segunda base superior (dimensiones en milímetros)

La base de la Figura 72 es la superficie que soporta las cremalleras del nivel superior.

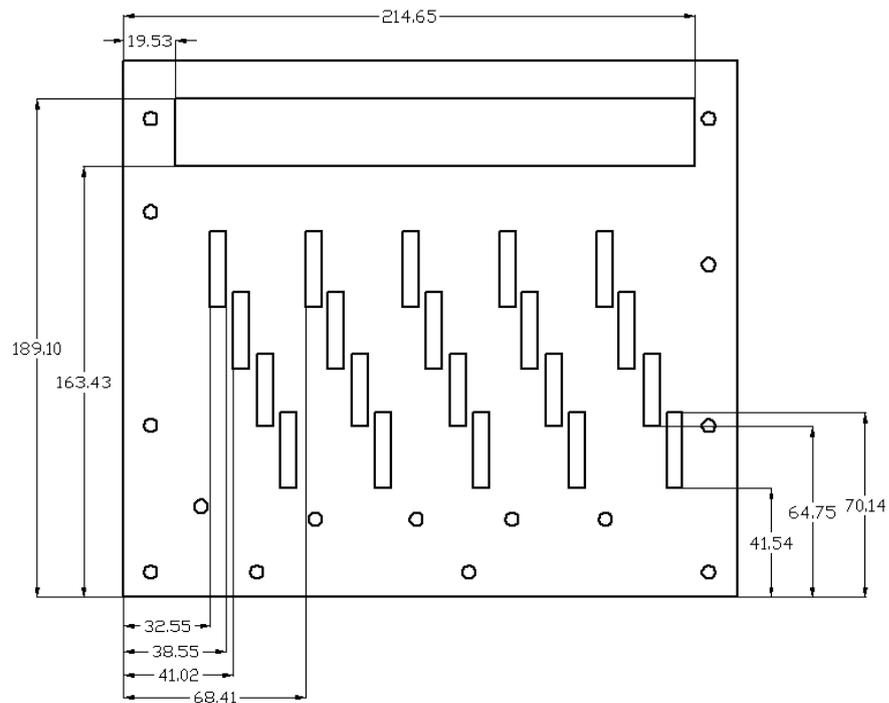


Figura 73. Base inferior de servomotores (dimensiones en milímetros)

El perímetro de todas las bases es igual. Las cremalleras del nivel inferior están ubicadas en la tapa inferior de la caja del dispositivo.

3.3.3 Piñones sujetos a los servomotores

Los piñones que se sujetan a los servomotores permiten el movimiento de las cremalleras.

Considerando que la separación entre la base para servomotores y la base para las cremalleras es de 10.8 mm, y que el engranaje del servomotor se encuentra a 4.5 mm de su base, como se observa en la Figura 74 el piñón debe tener un diámetro mínimo de 13 mm para sobresalir de la base.

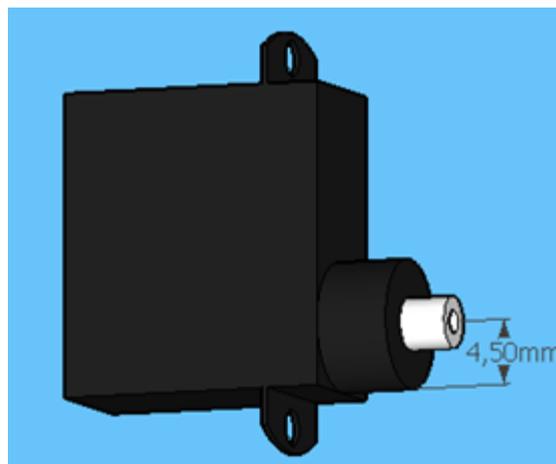


Figura 74. Distancia entre la base del servomotor y su engranaje

Tomando estas consideraciones, se determina un diámetro que cubra la mitad del espacio entre las bases de los servomotores.

La mitad del espacio es de 5.4 mm, sumando a esta distancia la distancia que se muestra en la Figura 74 y los 2 mm de espesor de la base, da un total de 11.9 mm, por lo que se determina un radio de esta magnitud, como se observa en la Figura 75.

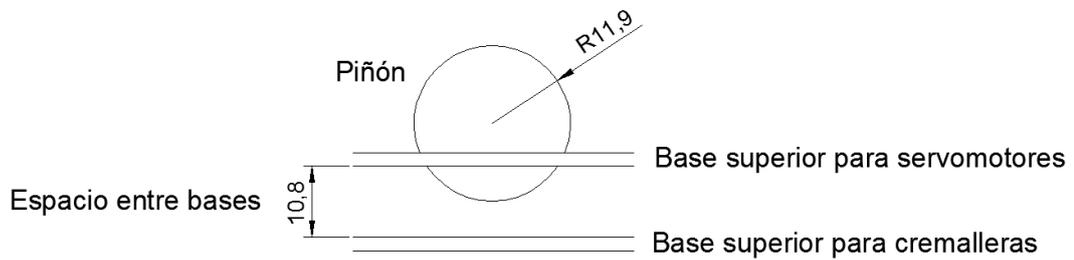


Figura 75. Ubicación del piñón entre las bases para servomotores (dimensiones en milímetros)

Una vez establecido el diámetro del piñón, se determina la altura de los dientes, para lo cual se escoge una distancia cercana a la mitad del espacio entre bases que se observa en la Figura 75, en este caso es de 4.68 mm.

Se establece que la rueda dentada tenga 21 dientes. El perímetro de la circunferencia del piñón es:

$$P = 2\pi R \quad (1)$$

$$P = 2 \cdot \pi \cdot 11.9 \text{ mm} \quad (2)$$

$$P = 74.77 \text{ mm} \quad (3)$$

Con un perímetro de 74.77 mm y 21 dientes, la longitud del arco entre dientes del piñón es de 3.56 mm, como se observa en la Figura 76.

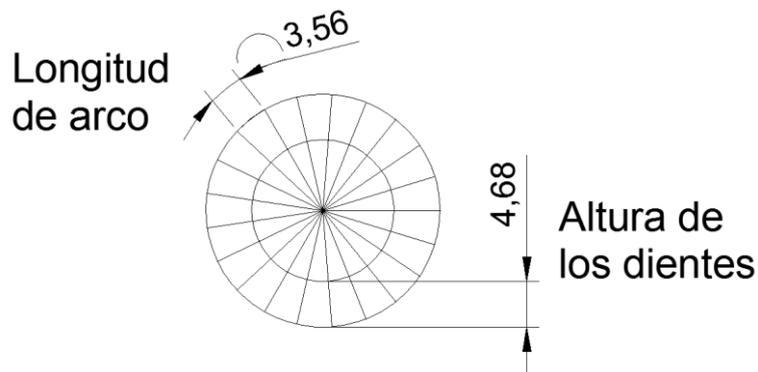


Figura 76. Diseño de la rueda dentada (dimensiones en milímetros)

Siendo el diámetro del piñón de 23.8 mm, se calcula el ángulo necesario para formar un arco cuya longitud sea de 3.56 mm para un radio de 11.9 mm.

$$s = R \cdot \theta \quad (4)$$

Donde s es la longitud del arco, R es el radio del círculo y θ es el ángulo a encontrar en radianes.

$$\theta = \frac{s}{R} \quad (5)$$

$$\theta = \frac{3.56 \text{ mm}}{11.9 \text{ mm}} \quad (6)$$

$$\theta = 0.277 \text{ rad} = 17.14^\circ$$

Por lo tanto se requiere formar un ángulo de 17.14° para tener una longitud de arco de 3.56 mm. En la Figura 77 se muestra el diseño de los dientes del piñón a partir de los cálculos realizados.

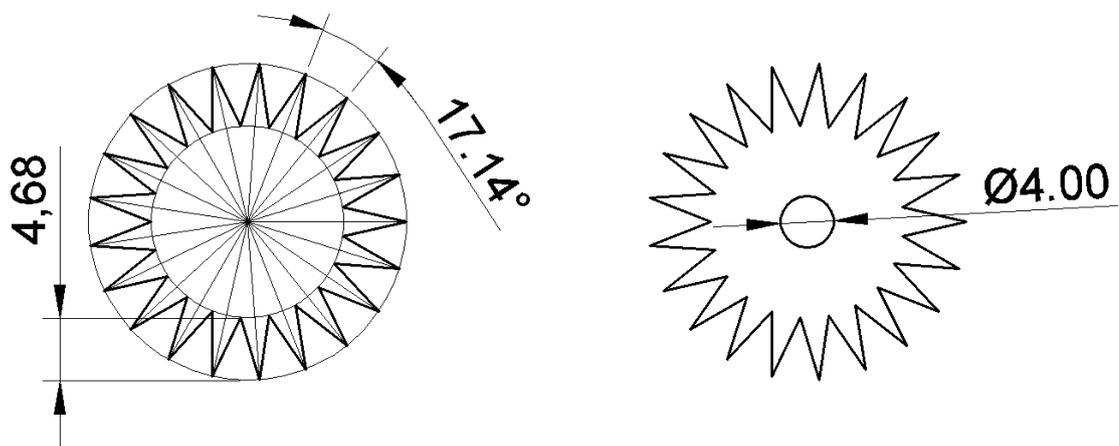


Figura 77. Diseño del piñón (dimensiones en milímetros)

El diámetro de 4 mm que se observa en el centro del piñón en la Figura 77 es equivalente al diámetro del engranaje del servomotor, de manera que permita acoplar el piñón al servomotor, como se observa en la Figura 78.

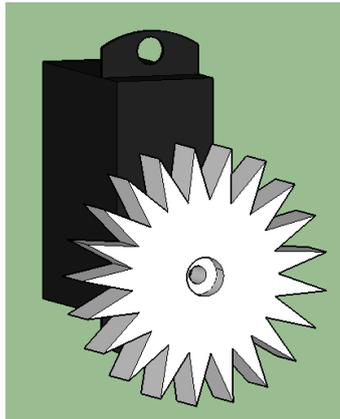


Figura 78. Acople del piñón al servomotor

El grosor de la rueda dentada se establece según el espacio que existe en las bases de servomotores. Siendo el ancho de los espacios en las bases de 6 mm, la rueda dentada tiene un grosor de 4.91 mm, de esta manera no existe fricción u obstrucción en el movimiento del piñón. En la Figura 79 se observa cómo se ubica el piñón en la base.

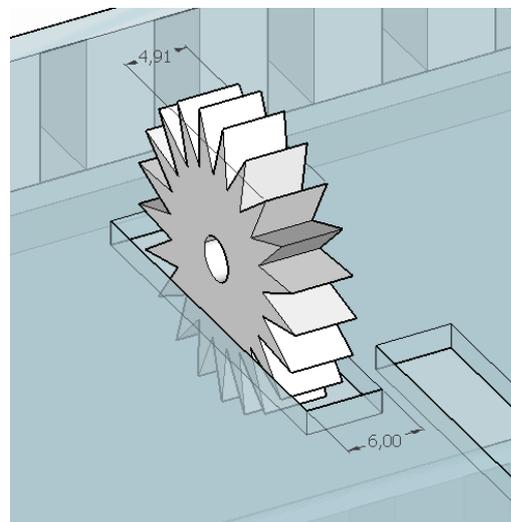


Figura 79. Ubicación del piñón en la base (dimensiones en milímetros)

Este piñón es adaptado al servomotor para controlar el desplazamiento de las cremalleras en las celdas Braille. En la Figura 80 se observa la ubicación del servomotor acoplado al piñón, sobre una de las bases.

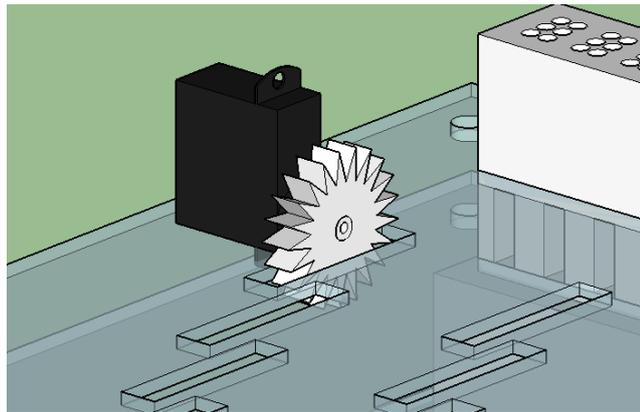


Figura 80. Ubicación del servomotor en una de las bases

3.3.4 Cremalleras con adaptación para levantar pines

El diseño de las cremalleras se muestra en la Figura 81. La adaptación especial está diseñada de manera que el mecanismo que levanta los pines funcione de manera similar a la de una lleva en un cerradura.

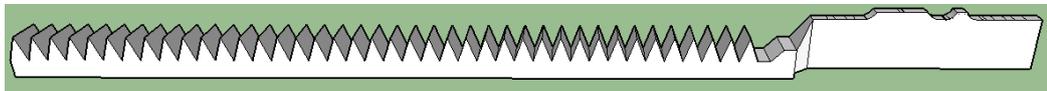


Figura 81. Cremallera con adaptación para levantamiento de pines

La cremallera está diseñada de manera que permita el levantamiento de un grupo de tres pines, formando las ocho posibles combinaciones con la menor cantidad de movimientos. Las ocho combinaciones posibles se muestran en la Figura 82, los puntos negros representan los pines levantados.

○	●	○	○
○	○	●	○
○	○	○	●
●	●	○	●
●	○	●	●
○	●	●	●

Figura 82. Ocho combinaciones posibles para cada fila de tres pines

La adaptación es insertada en los orificios de la pieza que conforma las celdas Braille y al igual que en una llave, el relieve de la adaptación permite levantar los pines de manera que se pueda desplegar los distintos caracteres Braille. El mecanismo se puede observar en la Figura 83.

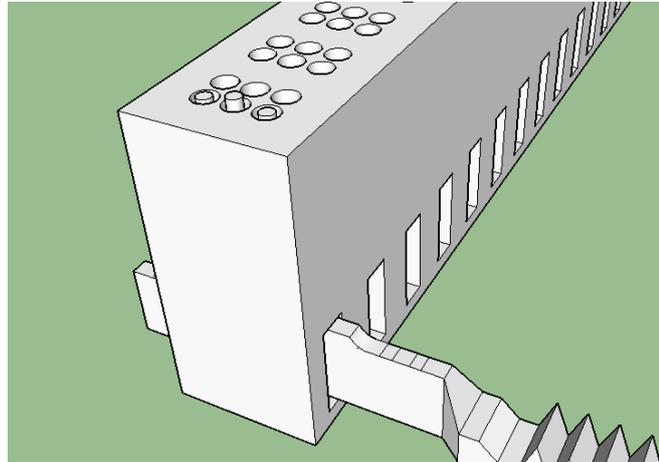
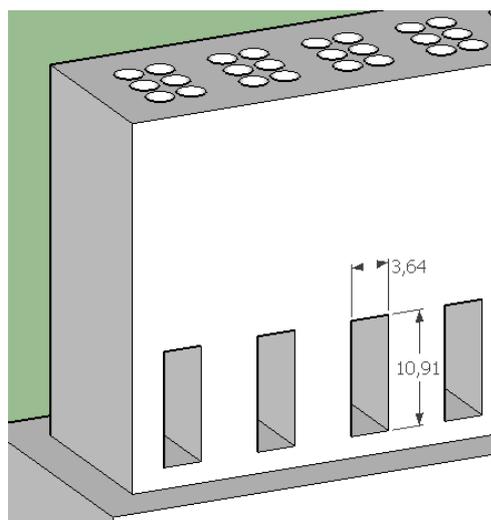


Figura 83. Mecanismo para el levantamiento de pines

Considerando que el espacio donde debe ingresar la punta de la cremallera es de 10.91 mm x 3.64 mm, como se observa en la Figura 84, las dimensiones de la punta de la cremallera se establecieron de manera que exista fricción únicamente en la parte inferior del espacio.



**Figura 84. Espacio donde ingresa la punta de la cremallera
(dimensiones en milímetros)**

Las dimensiones de la punta de la cremallera se establecieron tomando en cuenta que los pines deben elevarse 1 mm, que el ancho de los pines es de 1.5 mm, y que la separación entre pines es de 3.15 mm. El diseño de la punta de la cremallera se muestra en la Figura 85.

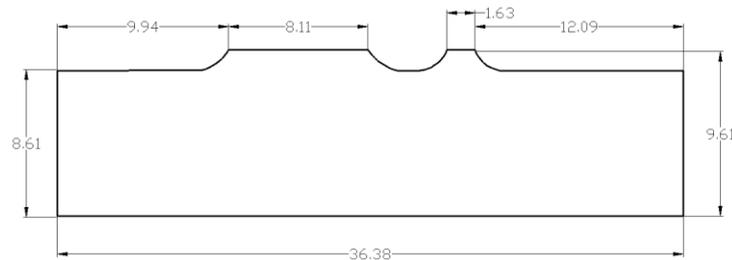


Figura 85. Punta de la cremallera (dimensiones en milímetros)

Como se observa en la Figura 85 y Figura 86, los relieves tienen una altura de 1 mm, además posee curvas a manera de rampa para permitir que los pines se deslicen fácilmente. Las dimensiones establecidas se aplican para las cremalleras de la parte superior, para la parte inferior los extremos son más extensos ya que la parte inferior de las celdas son más anchas. Los espacios de los relieves se mantienen igual así como la altura. En la Figura 86 se muestra el diseño de la punta de las cremalleras de la parte inferior.

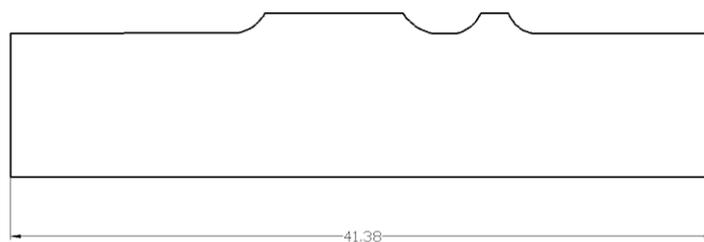


Figura 86. Punta de cremalleras inferiores (dimensiones en milímetros)

Las dimensiones de los relieves son suficientes para mantener los pines levantados, considerando el relieve más corto, el cual posee una longitud de 1.63 mm, ese espacio es suficiente para un pin de 1.5 mm de diámetro. El grosor de la punta de la cremallera es de 2.44 mm, de manera que no tenga

fricción con el orificio donde ingresa. Al momento de deslizarse al interior de las celdas Braille va levantando los pines, en la Figura 87 se muestra un corte a una celda Braille de manera que se puede apreciar la interacción entre los pines y la cremallera.

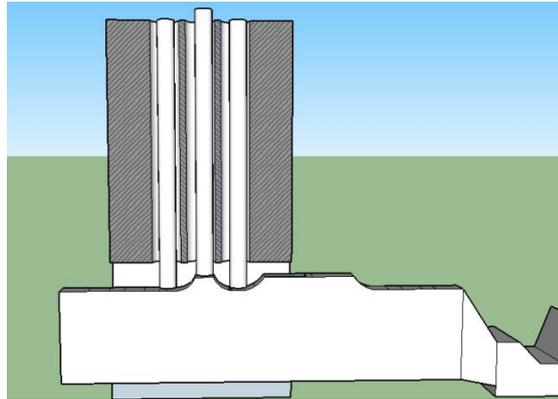


Figura 87. Interacción entre pines y cremallera

La cremallera debe cubrir el espacio suficiente en las bases para que se acople al piñón, como se muestra en la Figura 88.

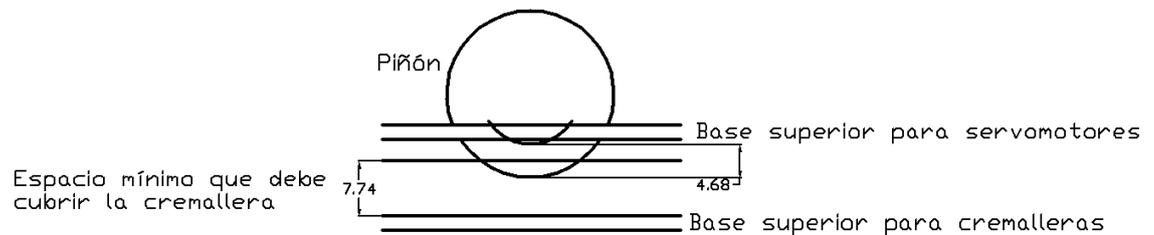


Figura 88. Consideraciones para el diseño de dientes de la cremallera

El diseño de los dientes de la cremallera tiene las mismas especificaciones de los dientes del piñón, de manera que se acoplen. La distancia entre picos es de 3.56 mm y la altura de los dientes es de 4.68 mm.

En la Figura 89 se muestra el diseño de los dientes de la cremallera.

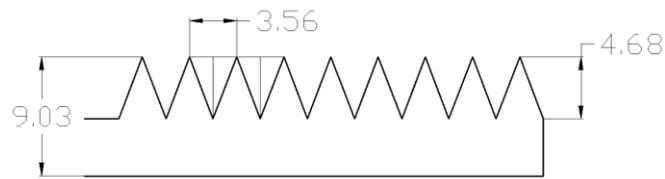


Figura 89. Diseño de los dientes de la cremallera (dimensiones en milímetros)

Como se observa en la Figura 89, la altura total de la cremallera es de 9.03 mm, una distancia mayor al espacio mínimo que debe cubrir, de esta forma se pueden acoplar la cremallera y el piñón. La altura de los dientes es la misma que la del piñón así como la distancia entre ellos. La Figura 90 muestra la unión de la punta de la cremallera con los dientes.

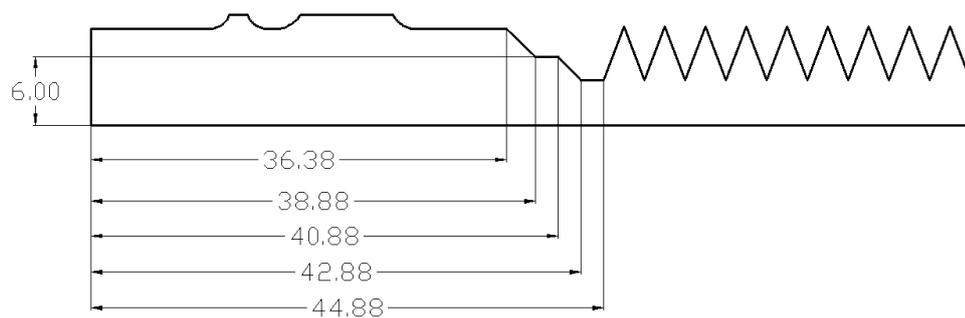


Figura 90. Diseño final de la cremallera (dimensiones en milímetros)

La longitud de las cremalleras es distinto según el servo que las controla, por lo que tienen cuatro longitudes diferentes, tanto las cremalleras superiores como las inferiores.

En total son 40 cremalleras que permiten el movimiento de todos los pines de las celdas Braille. En la Figura 91 y Figura 92 se observa los diseños de las cremalleras.

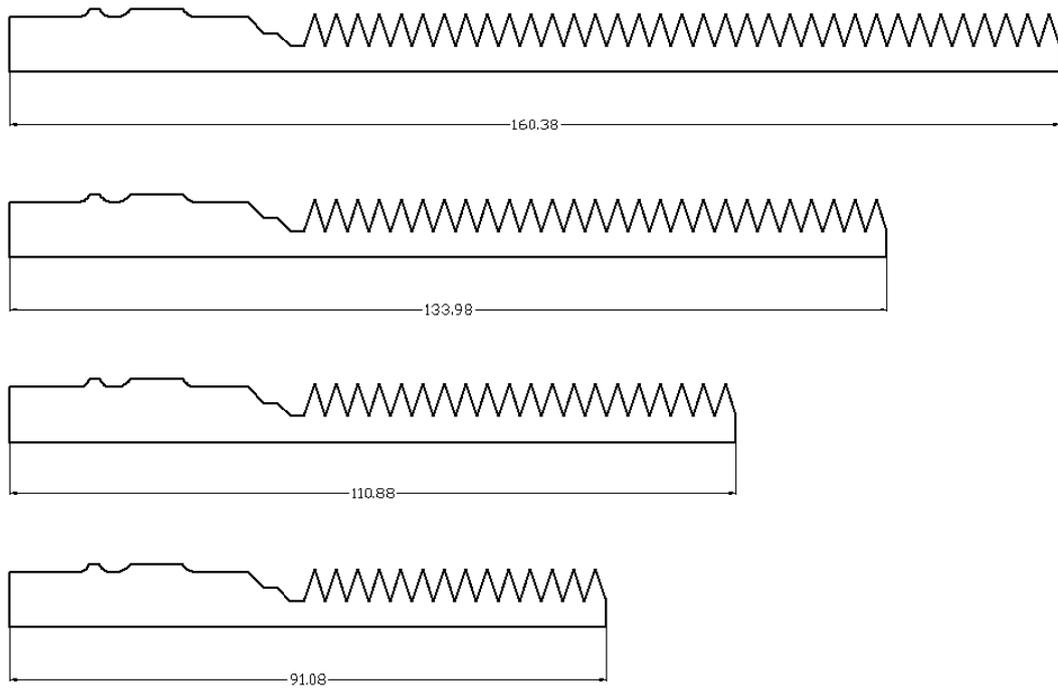


Figura 91. Diseño de cremalleras superiores (dimensiones en milímetros)

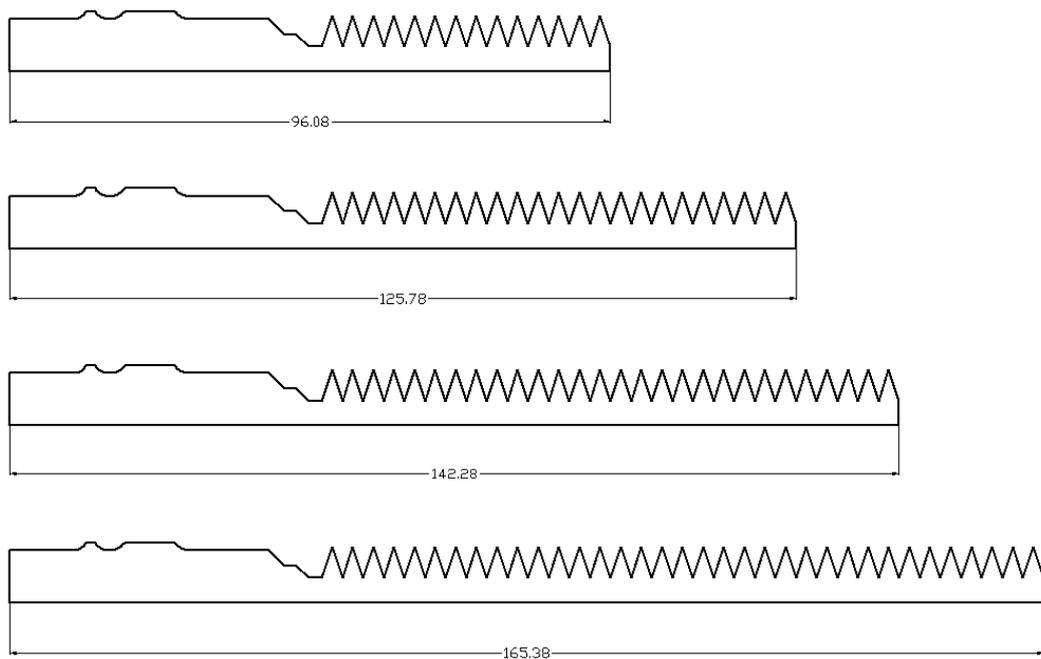


Figura 92. Diseño de cremalleras inferiores (dimensiones en milímetros)

El grosor de los dientes de la cremallera es de 5.94 mm, 1.03 mm más ancho que el piñón para asegurar el acople entre cremallera y piñón.

En la Figura 93 y Figura 94 se observa la vista superior y lateral del diseño de una cremallera.



Figura 93. Vista superior del diseño de la cremallera (dimensiones en milímetros)



Figura 94. Vista lateral del diseño de la cremallera

El desplazamiento de la cremallera se realiza mediante el piñón accionado por un servomotor, como se observa en la Figura 95. La posición del servomotor establece la distancia que debe moverse la cremallera para levantar los pines necesarios; de tal manera que se puedan desplegar físicamente los caracteres Braille.

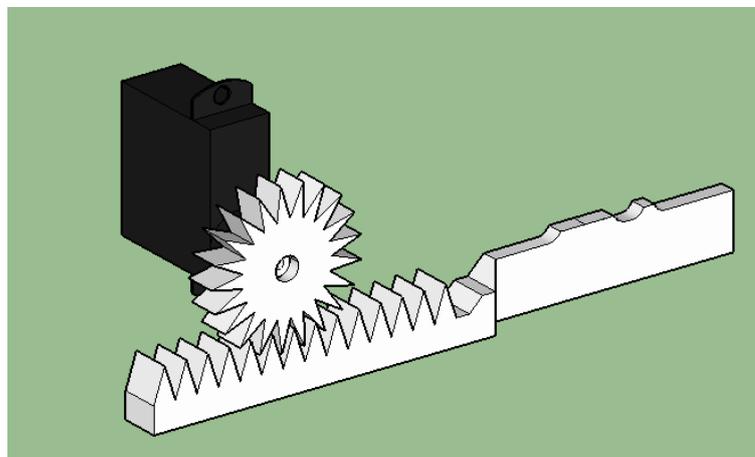


Figura 95. Interacción entre el piñón y la cremallera

3.3.5 Pines para formar los caracteres Braille

Los pines para formar los caracteres Braille son cilindros delgados que se insertan en los orificios de la parte superior de las celdas Braille. Tienen un espesor de 1.5 mm y una longitud de 25 mm para las filas de la derecha y 68 mm para las filas de la izquierda. En la Figura 96 se muestra el diseño de los pines.

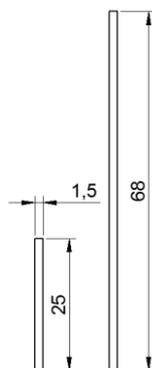


Figura 96. Diseño de los pines (dimensiones en milímetros)

El número total de pines es de 120 (60 de 25 mm y 60 de 68 mm) ya que existen 20 celdas Braille formadas por seis pines, tres pines de 25 mm y tres de 68 mm. En la Figura 97 se observa cómo se ubican los pines en la celda Braille.

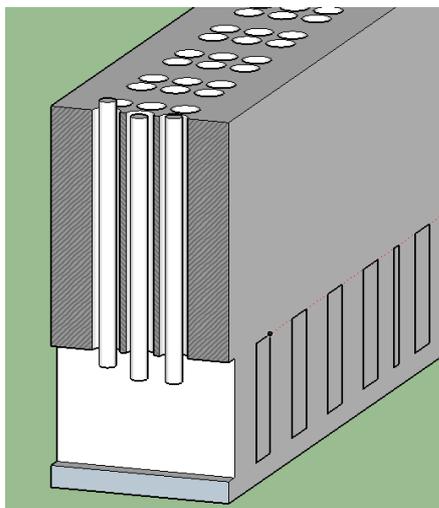


Figura 97. Ubicación de los pines en las celdas Braille

3.3.6 Caja del prototipo

La caja del prototipo contiene todos los elementos mecánicos ensamblados y la parte electrónica que comprende los controladores de los servomotores y la placa para la comunicación con la computadora.

En la Figura 98 se muestra la caja del prototipo ensamblada.

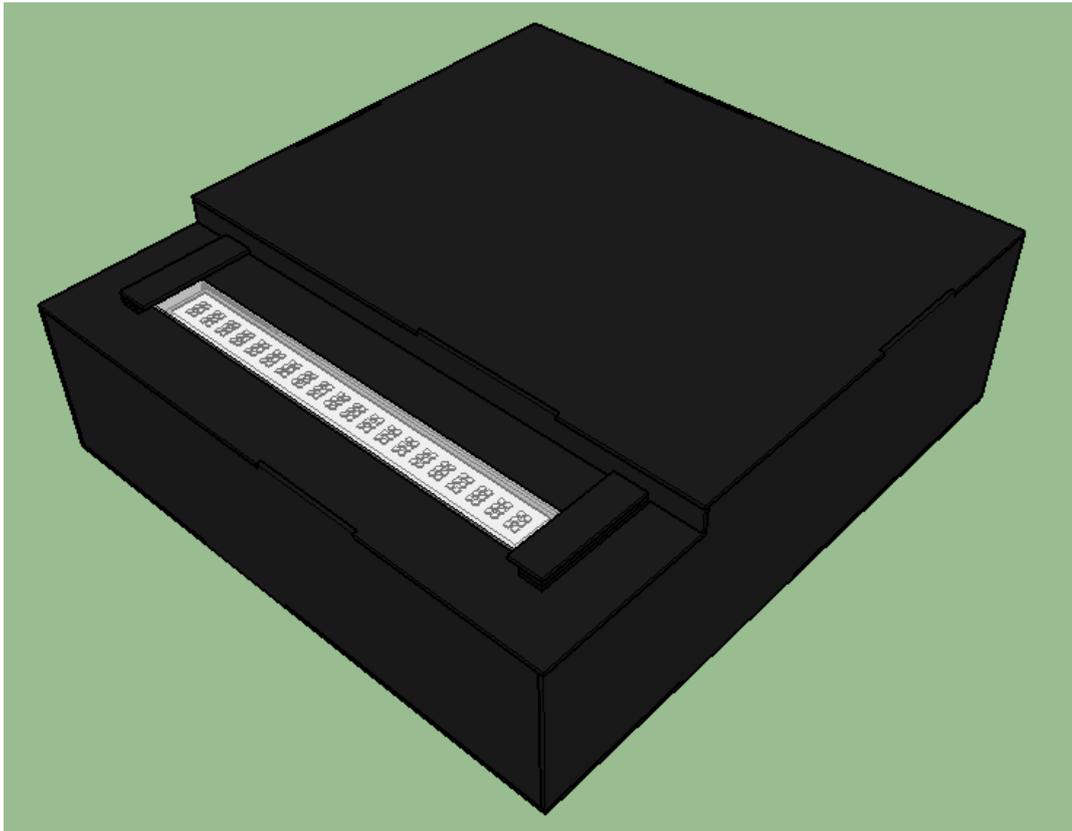


Figura 98. Caja del prototipo

Las dimensiones de las distintas piezas que conforman la caja se muestran en la Figura 99, Figura 100, Figura 101, Figura 102, Figura 103, Figura 104, Figura 105, Figura 106, Figura 107 y Figura 108.

Todas las piezas tienen un grosor de 2 milímetros y están diseñadas con pestañas que permiten que las piezas encajen con mayor precisión y rigidez.

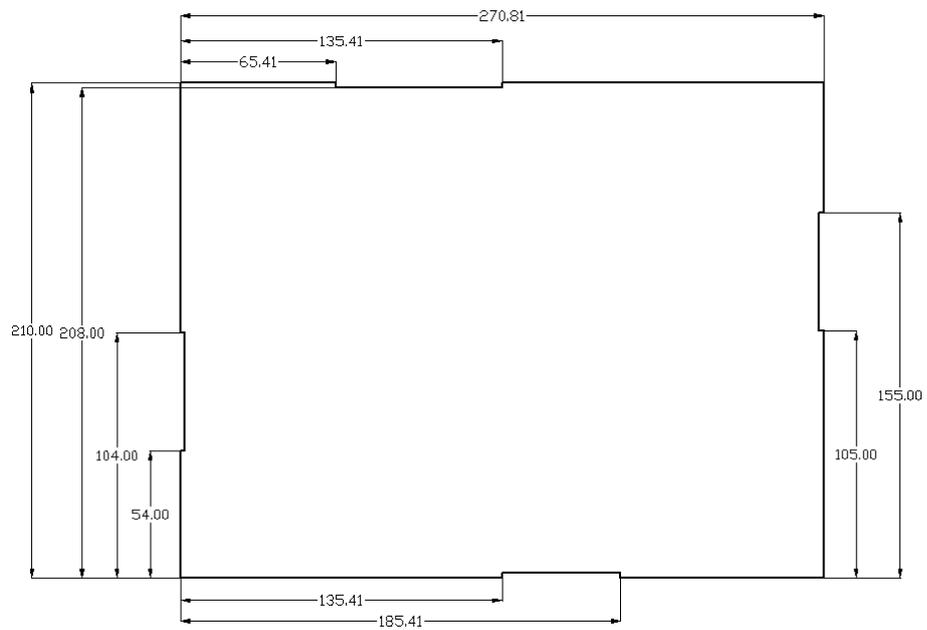


Figura 99. Tapa superior principal (dimensiones en milímetros)

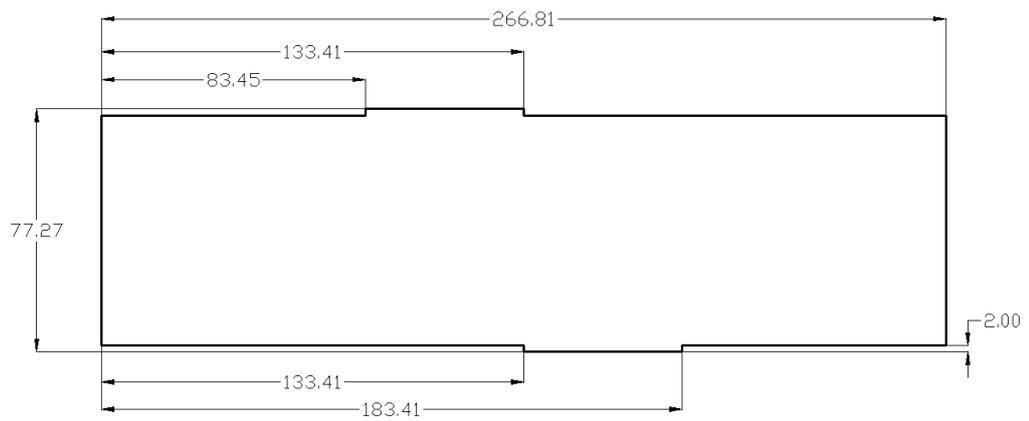


Figura 100. Tapa frontal (dimensiones en milímetros)

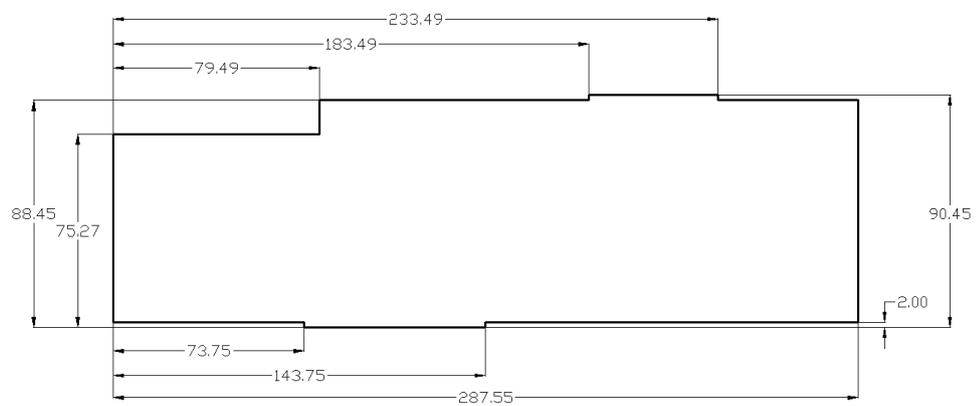


Figura 101. Tapa lateral izquierda (dimensiones en milímetros)

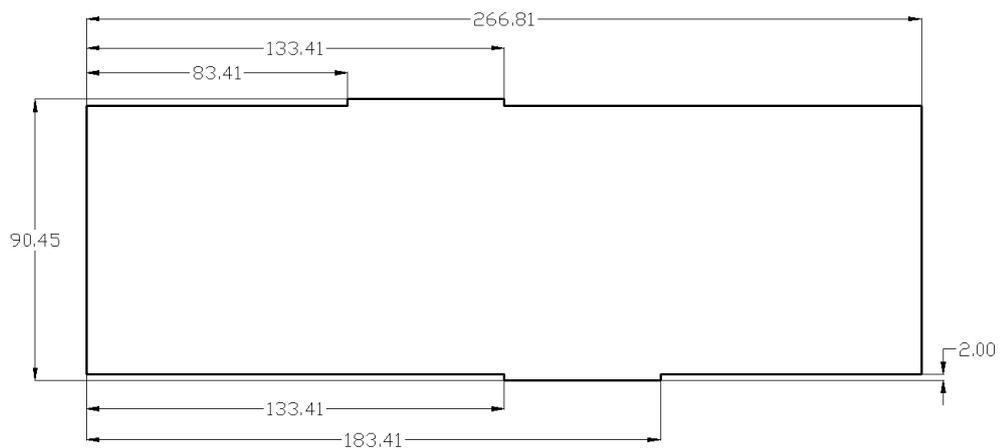


Figura 102. Tapa lateral derecha (dimensiones en milímetros)

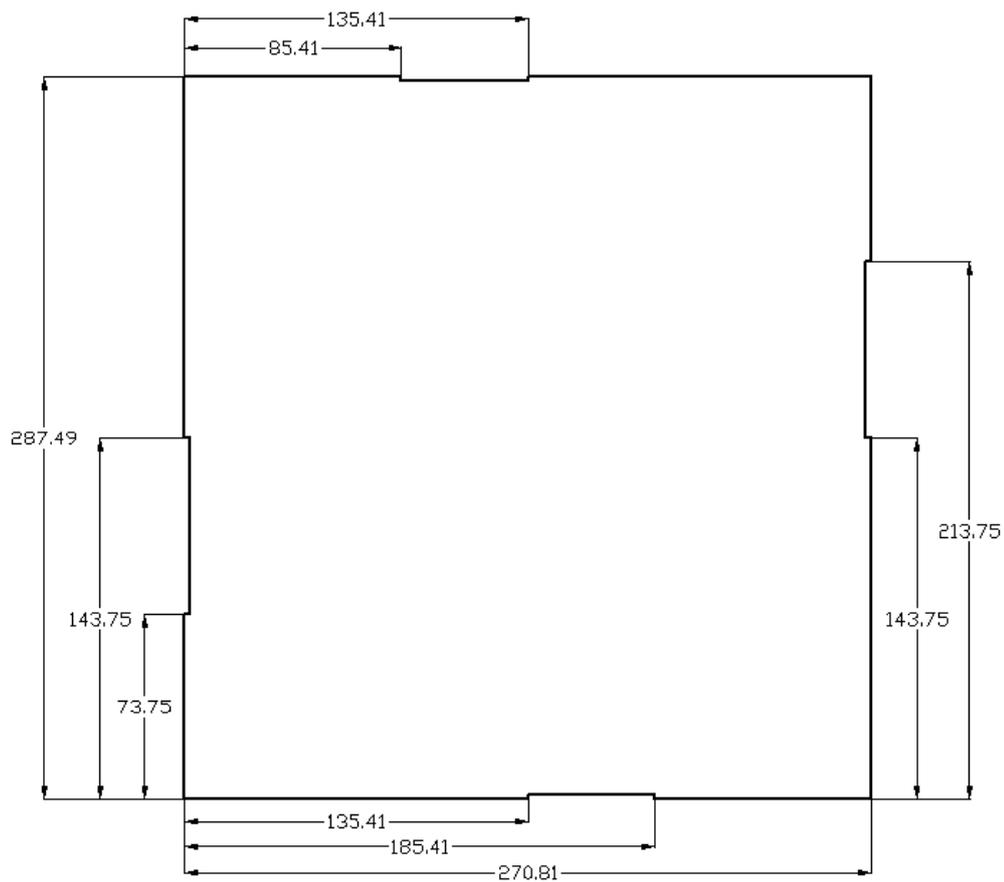


Figura 103. Tapa inferior (dimensiones en milímetros)

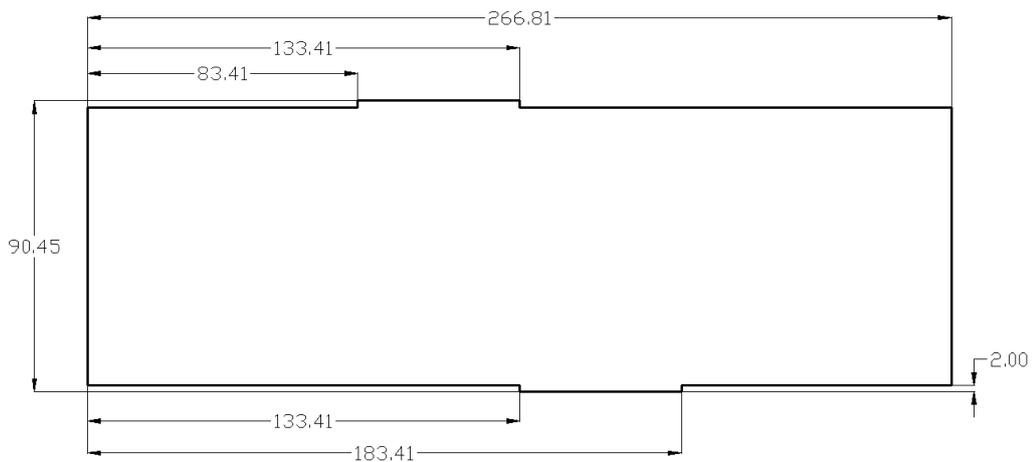


Figura 104. Tapa posterior (dimensiones en milímetros)

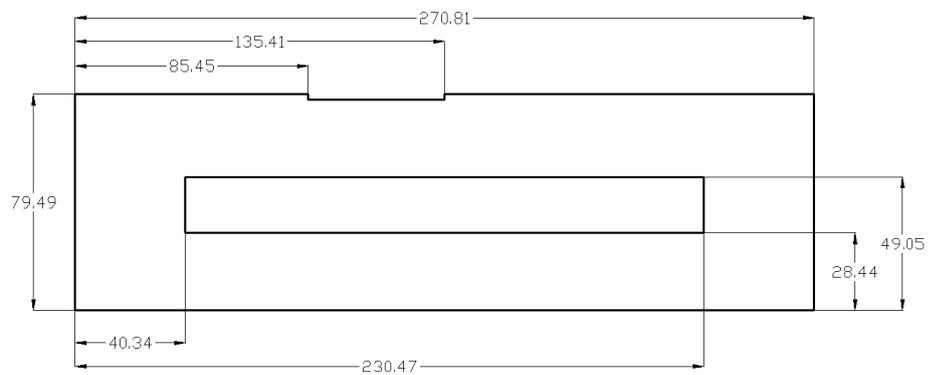


Figura 105. Segunda tapa superior (dimensiones en milímetros)

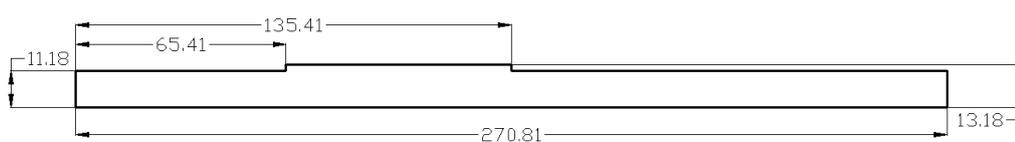


Figura 106. Segunda tapa frontal (dimensiones en milímetros)



Figura 107. Tapa protectora para las celdas Braille (dimensiones en milímetros)

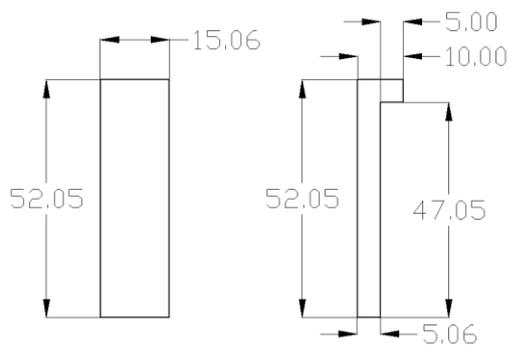


Figura 108. Soportes para la tapa protectora (dimensiones en milímetros)

El ensamblaje de las piezas de la caja se muestra en la Figura 109.

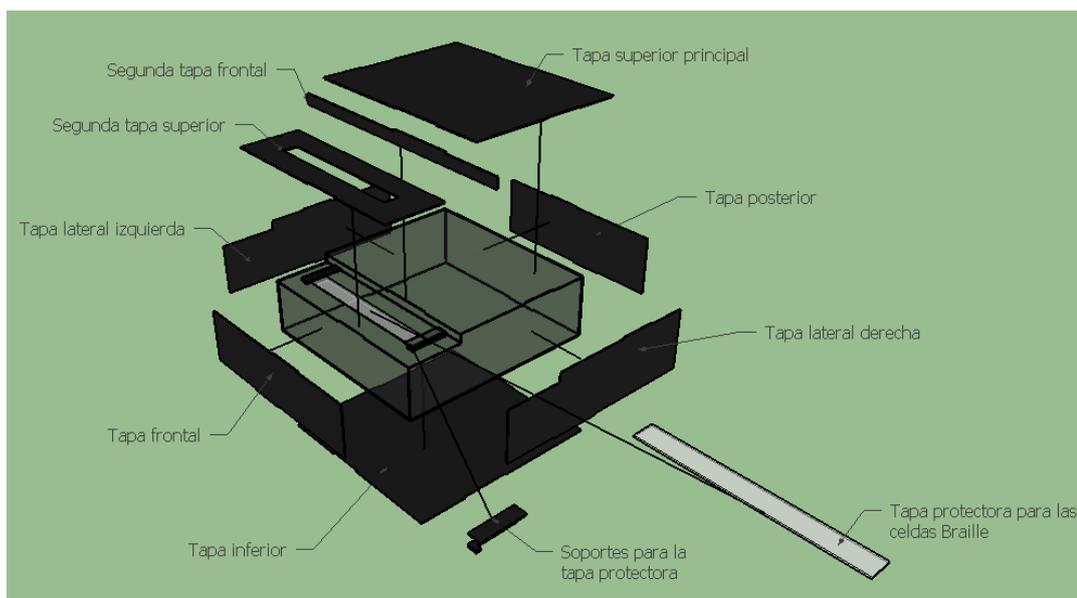


Figura 109. Ensamblaje de las piezas de la caja

En la Figura 110 se muestra como se ensamblan las piezas que permiten que la tapa protectora de las celdas Braille se pueda mover para que el usuario pueda leer.

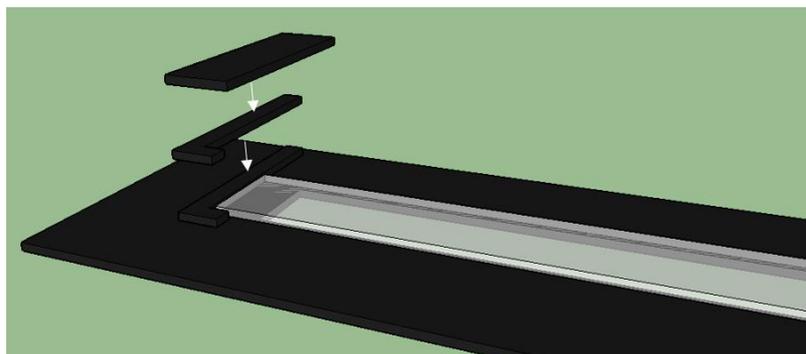


Figura 110. Ensamblaje de los soportes para la tapa protectora de las celdas Braille

3.3.7 Ubicación de los servomotores

La ubicación de los servomotores está distribuida de manera que las cremalleras puedan desplazarse sin inconveniente. En la Figura 111 se muestra la distribución de los servomotores en el dispositivo.

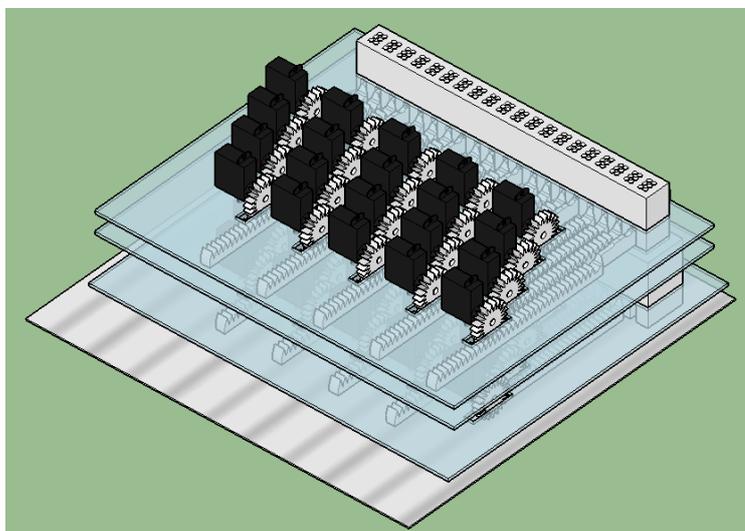


Figura 111. Distribución de servomotores en el dispositivo

Los servomotores están distribuidos de manera que no representen un obstáculo para los demás, es por eso que están ubicados a manera de escalones en grupos de cuatro, como se observa en la Figura 112 y Figura 113.

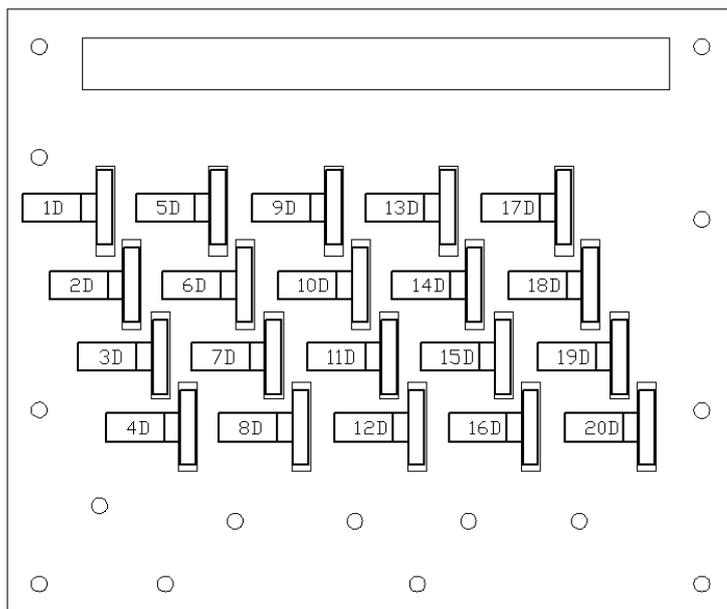


Figura 112. Ubicación de los servomotores en la base superior

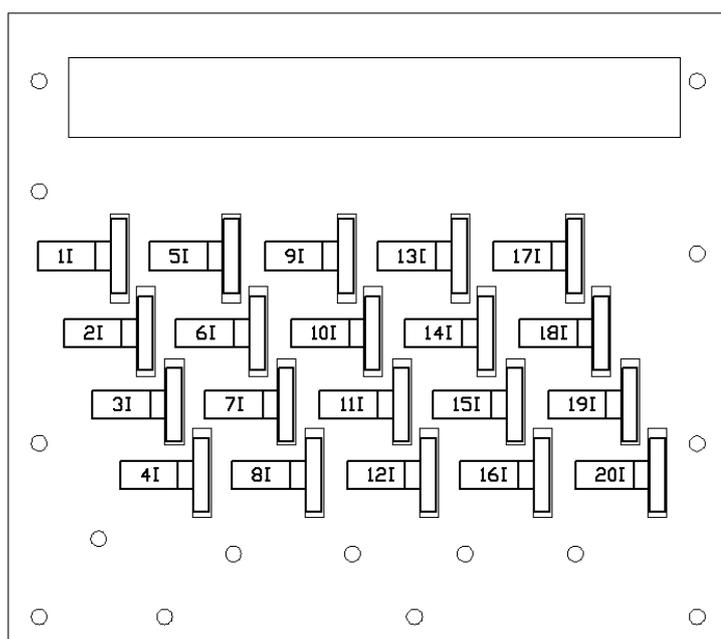


Figura 113. Ubicación de los servomotores en la base inferior

Como se observa en la Figura 112 y Figura 113, la numeración de los servomotores es del 1 al 20 acompañados de una letra, D o I, la D significa derecha, es decir los motores del nivel superior mueven los pines de las filas de la derecha en las celdas Braille, y la letra I está asignada a los motores del

nivel inferior los cuales mueven los pines de las filas de la izquierda. En la Figura 114 se muestran los dos niveles de servomotores del dispositivo

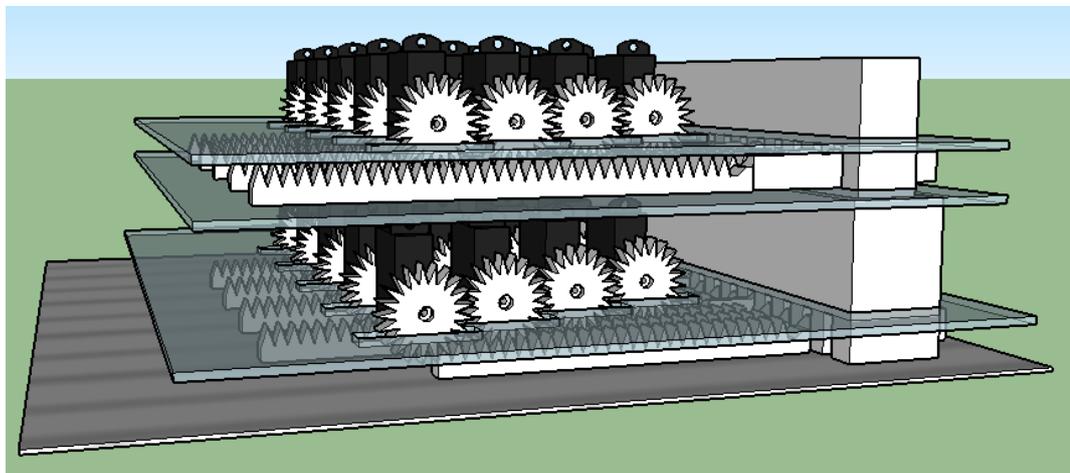


Figura 114. Ubicación de servomotores en dos niveles

En la Figura 115 se muestra como las bases permite el movimiento de las cremalleras sin que los motores representen un obstáculo.

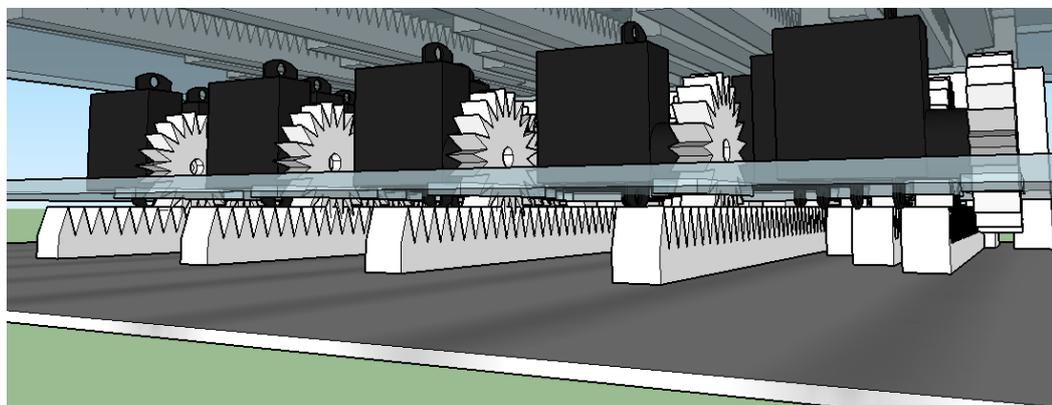


Figura 115. Motores montados en base inferior

3.3.8 Proceso para desplegar caracteres Braille

El movimiento de los servomotores depende de la posición de la adaptación de la cremallera, puesto que la adaptación posee elevaciones que permiten levantar los pines para desplegar los caracteres Braille. Las posiciones fueron consideradas de manera que se ocupe la menor cantidad de movimientos y que permitan representar todos los caracteres braille.

El sistema está dividido por filas de tres puntos, donde cada cremallera tiene el control del movimiento de tres pines, lo que representa el uso de 40 cremalleras para controlar los 120 pines que conforman las celdas Braille. Los grupos de tres puntos pueden formar ocho combinaciones las cuales se representan como se muestra en la Figura 116. Cada pareja de cremalleras permite desplegar los distintos caracteres Braille.

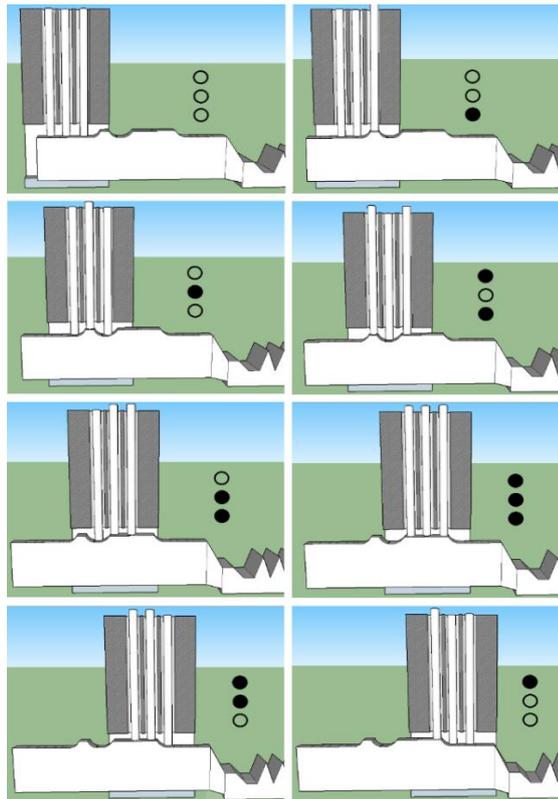


Figura 116. Combinaciones de pines posibles y los movimientos de la cremallera

La densidad del plástico ABS es de 1.05 g/cm^3 , por lo que no representa un problema en cuanto a la interacción entre las piezas, es decir no hay peligro de rupturas o torsiones de las piezas.

Debido a que las piezas citadas anteriormente tienen contacto entre sí, se debe considerar el coeficiente de fricción de este tipo de plástico para calcular la fuerza necesaria para que se puedan mover todas las piezas en conjunto. El coeficiente de fricción del plástico ABS es de 0.48. A continuación se

muestran los cálculos de fuerza para que se puedan realizar los movimientos de las piezas.

Para el cálculo de fuerza se considera los elementos más pesados, los cuales son la cremallera más larga del nivel inferior y el pin más largo.

- Masa de la cremallera: 8.04 g
- Masa del pin: 0.98 g

En la Figura 117 se muestra el movimiento que tiene que realizar la cremallera sobre la superficie de las celdas Braille.

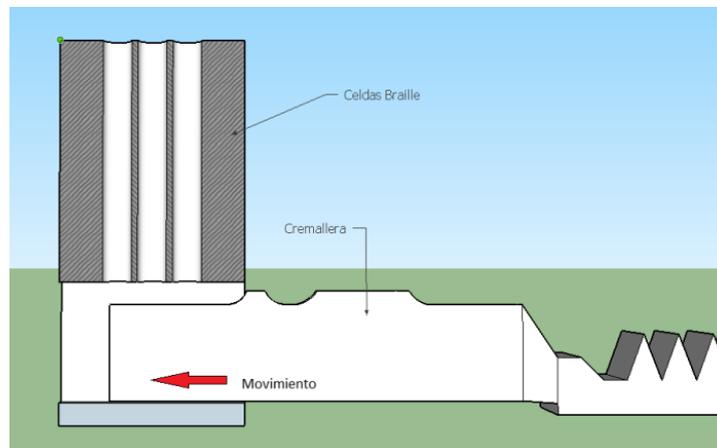


Figura 117. Movimiento de la cremallera

En la Figura 118 se muestra el diagrama de cuerpo libre de la cremallera.

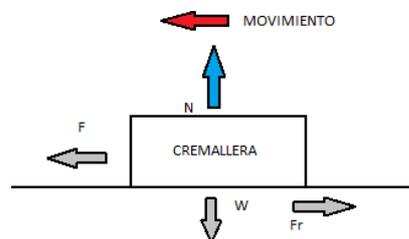


Figura 118. Diagrama de cuerpo libre de la cremallera

Se calcula la fuerza F mínima que permita el movimiento de la cremallera.

$$W = m \cdot g \quad (7)$$

$$W = 0.00804Kg \cdot 9.81m/s^2 \quad (8)$$

$$W = 78.87 \times 10^{-3} N$$

Teniendo el peso W se puede calcular la normal N

$$N = 78.87 \times 10^{-3} N$$

Conociendo que el coeficiente de rozamiento del material es de 0.48, se calcula la fuerza de rozamiento Fr .

$$Fr = \mu \cdot N \quad (9)$$

$$Fr = 0.48 \cdot 78.87 \times 10^{-3} N \quad (10)$$

$$Fr = 37.85 \times 10^{-3} N$$

Por lo tanto, la fuerza F debe ser mayor a $39.43 \times 10^{-3} N$ para poder mover las cremalleras.

Utilizando el mismo DCL de la Figura 118 se calcula la fuerza necesaria para mover las cremalleras en las bases acrílicas, tomando en cuenta que el coeficiente de fricción en este caso es de 0.53.

$$Fr = \mu \cdot N \quad (11)$$

$$Fr = 0.53 \cdot 78.87 \times 10^{-3} N \quad (12)$$

$$Fr = 41.8 \times 10^{-3} N$$

Lo que indica que la fuerza que se necesita para mover las cremalleras debe ser mayor a $41.8 \times 10^{-3} N$.

En la Figura 119 se muestra el movimiento que deben realizar los pines en el sistema.

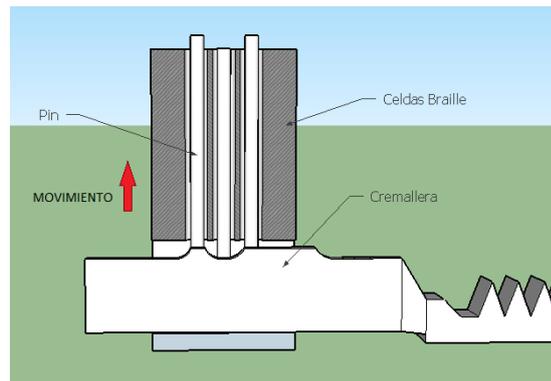


Figura 119. Movimiento de los pines

En la Figura 120 se muestra el diagrama de cuerpo libre de un pin. Se consideró una inclinación de 45° en las rampas que permiten elevar los pines en las celdas Braille.

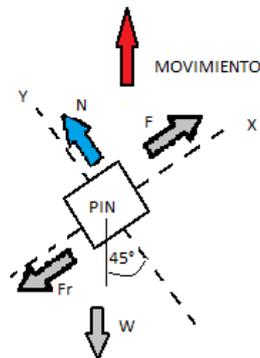


Figura 120. Diagrama de cuerpo libre de un pin

Considerando el coeficiente de fricción de 0.48 se realiza los siguientes cálculos.

$$F - Wx - Fr = 0 \quad (13)$$

$$\Sigma Fy = 0 \quad (14)$$

$$N - W \cos 45^\circ = 0 \quad (15)$$

$$N = W \cos 45^\circ \quad (16)$$

$$N = 0.98 \times 10^{-3} Kg \cdot 9.81 m/s^2 \cdot \cos 45^\circ \quad (17)$$

$$N = 6.798 \times 10^{-3} N$$

Reemplazando los valores en la ecuación (13) se tiene:

$$F = 0.98 \times 10^{-3} Kg \cdot 9.81 m/s^2 \cdot \text{sen } 45^\circ + 0.48 \cdot 6.798 \times 10^{-3} N \quad (18)$$

$$F = 10.061 \times 10^{-3} N$$

Esa fuerza F es paralela al plano inclinado donde se mueve el pin, por lo tanto la componente en el eje X de esa fuerza es la que se necesita vencer para que la cremallera pueda mover el pin.

$$F_x = 10.061 \times 10^{-3} N \cdot \cos 45^\circ \quad (19)$$

$$F_x = 7.11 \times 10^{-3} N$$

Considerando que en los cálculos anteriores la mayor fuerza obtenida fue de $41.8 \times 10^{-3} N$, la fuerza que se necesita para mover los pines mediante las cremalleras es la suma de dicha fuerza F más la fuerza Fx. La fuerza total a vencer es de $48.91 \times 10^{-3} N$.

CAPÍTULO 4

IMPLEMENTACIÓN DEL PROTOTIPO

4.1 Selección de componentes mecánicos

4.1.1 Celdas Braille, piñones, cremalleras y pines

Para la implementación de las celdas Braille, piñones, cremalleras y pines se consideró tres materiales, cuyas características se muestran en el Cuadro 12.

Cuadro 12

Cuadro comparativo de tipos de materiales

CARACTERÍSTICAS	PLÁSTICO ABS (IMPRESIÓN 3D)	PLÁSTICO POLIPROPILENO (INYECCIÓN DE MOLDE)
Tiempo de fabricación	23 horas	30 días incluida la fabricación del molde
Resolución	0.1 mm	0.001 mm
Precio	US \$ 202.13	US \$ 850.00

Como se observa en el Cuadro 12, la única ventaja que presenta la inyección de molde, entre las características comparadas es la resolución, pero considerando que la dimensión más pequeña para una pieza del prototipo es de 1.5 mm, se ha escogido la impresión en tres dimensiones, ya que representa una ventaja tanto económica como de tiempo.

La impresión en 3D da la libertad de fabricar virtualmente cualquier pieza, es así que los piñones, cremalleras, celdas Braille y los pines fueron desarrollados con este método.

En la Figura 121, Figura 122, Figura 123, Figura 124 y Figura 125 se muestran las piezas implementadas mediante la impresión en tres dimensiones.



Figura 121. Nivel superior de las celdas Braille



Figura 122. Vista lateral de las celdas Braille

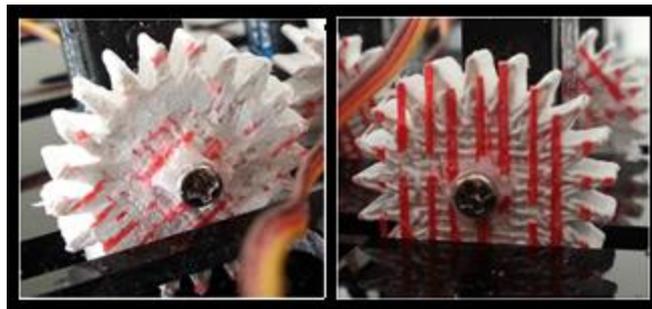


Figura 123. Piñones fabricados con impresión 3D



Figura 124. Cremallera fabricada con impresión 3D



Figura 125. Pines fabricados con impresión 3D

4.1.2 Bases para servomotores y caja del prototipo

Para las bases que sostienen los servomotores y la caja del prototipo se consideraron los materiales que se muestran en el Cuadro 13.

Cuadro 13

Cuadro comparativo de materiales para bases y caja del prototipo

CARACTERÍSTICAS	PLÁSTICO ABS (IMPRESIÓN 3D)	LÁMINA DE ACRÍLICO (CORTE LÁSER)
Tiempo de fabricación	20 horas	30 minutos
Resolución	0.1 mm	0.1 mm
Precio	US \$ 129.71	US \$ 18.00
Densidad	1.05 g/cm ³	1.18 g/cm ³

La implementación de las bases para servos y para las tapas de la caja se ha realizado utilizando un corte láser en láminas de acrílico, debido a que la tratarse de superficies planas de 2 mm de espesor, el Cuadro 13 muestra claramente que el trabajo en láminas de acrílico presenta la ventaja, tanto en tiempo de fabricación, tiempo y precio, además al tener mayor densidad, la lámina de acrílico es de mayor rigidez, lo que resulta conveniente al sostener 20 servomotores. En la Figura 126, Figura 127, Figura 128, Figura 129, Figura 130, y Figura 131 se muestran las láminas de acrílico implementadas.

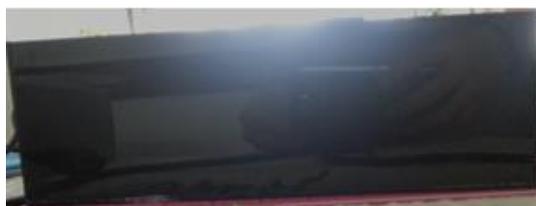


Figura 126. Tapa posterior de la caja



Figura 127. Tapa frontal de la caja

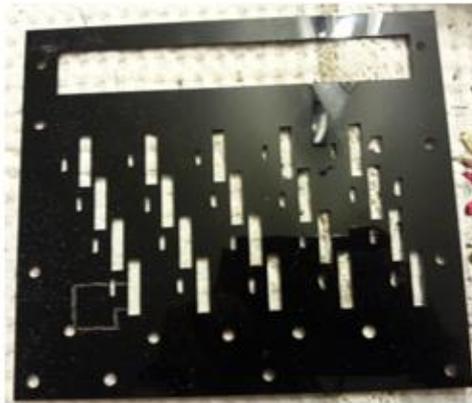


Figura 128. Lámina de acrílico para la base superior de servomotores



Figura 129. Tapa superior de la caja

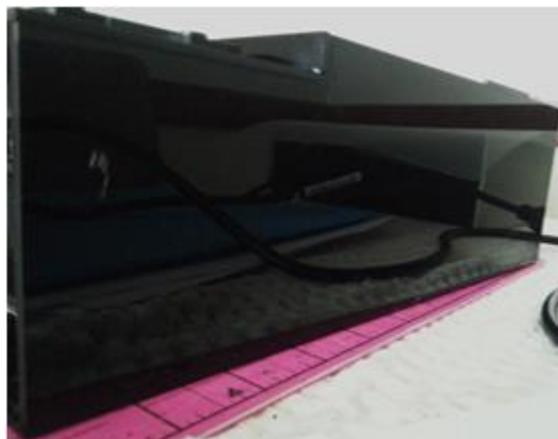


Figura 130. Tapa lateral derecha de la caja



Figura 131. Tapa de protección para las celdas Braille acoplada a la segunda tapa superior

4.2 Selección de componentes electrónicos

4.2.1 Microcontrolador

Los requerimientos esenciales para el microcontrolador son que pueda manejar tres puertos seriales simultáneamente y que cuente con cuatro entradas digitales para botones. En el Cuadro 14 se realiza una comparación entre la serie de microcontroladores más comunes.

Cuadro 14

Cuadro comparativo entre microcontroladores PIC

CARACTERÍSTICAS	PIC16F876A	PIC16F877A	PIC18F2550	PIC18F4550
Memoria de programa	14 KB	14 KB	32 KB	32 KB
Velocidad de cpu	5 MIPS	5 MIPS	12 MIPS	12 MIPS
Ram	368 Bytes	368 Bytes	2048 Bytes	2048 Bytes
Puertos seriales	1 UART	1 UART	1 UART	1 UART
Voltaje de alimentación	4.5 V – 5.5 V			
Número de pines	28	40	28	40
Precio	US \$ 5.00	US \$ 6.50	US \$ 8.32	US \$ 8.50

Considerando que se necesita un total de diez pines para cubrir las cuatro entradas digitales para botones, y los tres puertos de comunicación serial, el

PIC16F876A sería el microcontrolador adecuado, pero en las pruebas realizadas este microcontrolador presentó fallas al manejar los tres puertos seriales simultáneamente por lo que se optó por el PIC18F2550.

La placa para el microcontrolador fue diseñada de manera que en un espacio reducido se pueda implementar los elementos necesarios para poder establecer la comunicación entre todos los dispositivos. Los elementos utilizados para la implementación de la placa son los siguientes:

- Microcontrolador PIC18F2550
- 1 Cristal de 20 MHz
- 2 Capacitores de 22 pF
- 5 Resistencias de 1K Ω
- 3 borneras de 3 pines
- 5 borneras de 2 pines

En la Figura 132 se observa el circuito esquemático del circuito impreso.

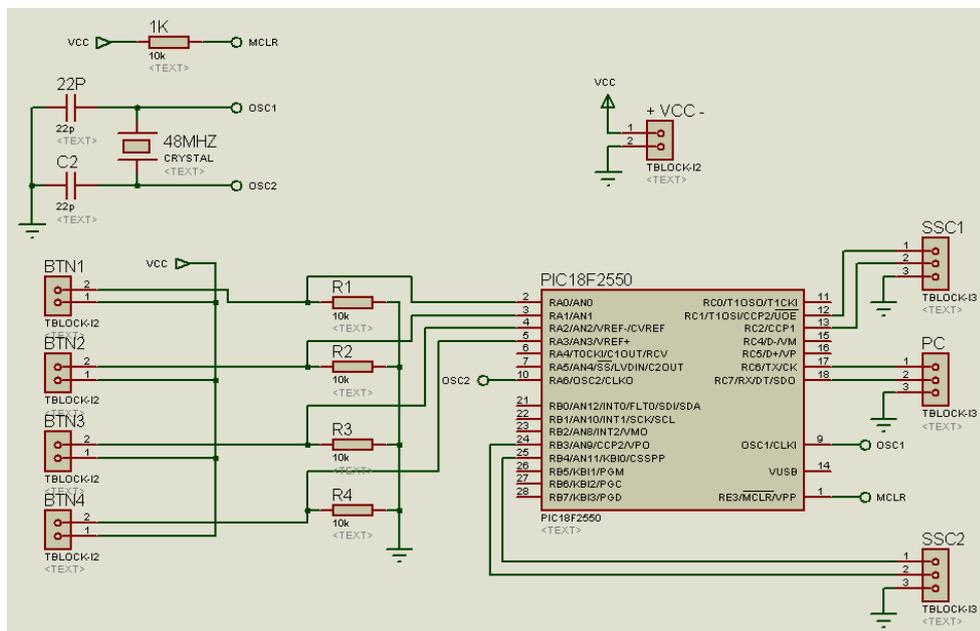


Figura 132. Circuito esquemático

En la Figura 133 se observa el diseño del circuito impreso con los componentes.

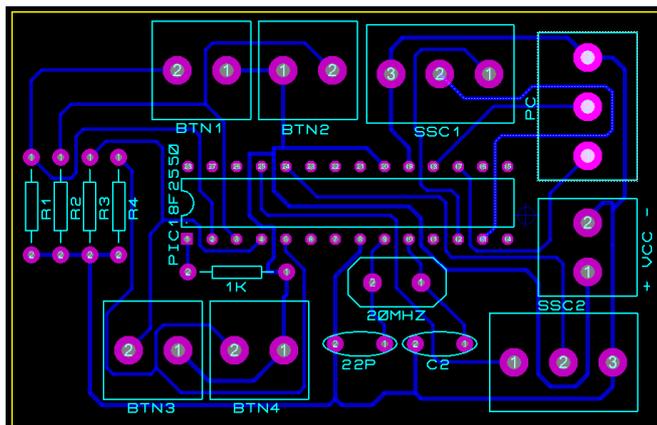


Figura 133. Diseño de la placa para el microcontrolador

En la Figura 134 y Figura 135 se observa la placa implementada.



Figura 134. Vista superior del circuito implementado

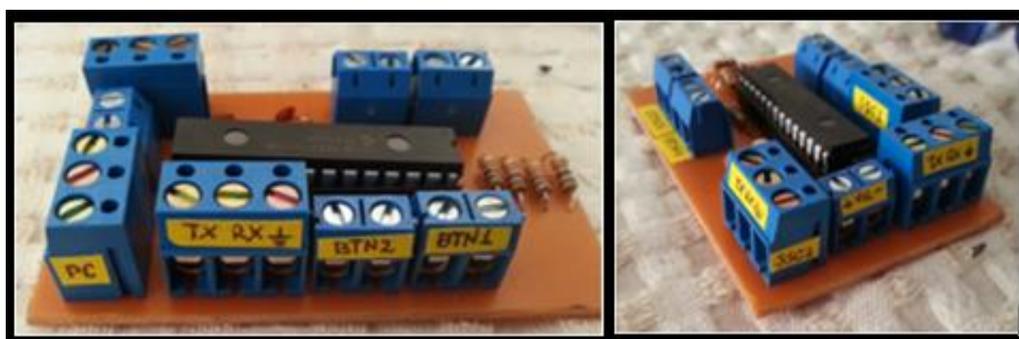


Figura 135. Circuito implementado

Como se observa en la Figura 135 todas las borneras están etiquetadas de manera que indiquen que componentes y que pines de los componentes se deben conectar.

4.2.2 Controladores para servomotores

Las tarjetas que controlan los servomotores trabajan mediante comandos seriales. En el Cuadro 15 se muestra una comparación entre dos modelos de tarjetas.

Cuadro 15

Cuadro comparativo de controladores para servomotores

CARACTERÍSTICAS	LYNXMOTION SSC-32U	POLOLU MINI MAESTRO 24-CHANNEL
Puertos de comunicación	USB/ Serial TTL	USB / Serial TTL
Resolución	0.09°	0.025°
Control de velocidad individual	Si	Si
Número de canales	32	24
Alimentación independiente para servos y para control	Si	No
Disponibilidad en el país	Si	No
Precio	US \$ 60.00	US \$ 49.95

Considerando que los servomotores se mueven 180°, y que se necesita ochos posiciones, la resolución del controlador SSC-32U es suficiente para manejar esas posiciones, además este controlador tiene aislada la parte de alimentación para los servos de la alimentación de control lo que implica una ventaja en la seguridad de la placa y los componentes conectados a esta. Finalmente, tomando en cuenta que se puede adquirir fácilmente dentro del país se optó por este controlador para los servomotores. Se implementaron dos controladores manejados por el microcontrolador mencionado anteriormente, ya que se necesita controlar 40 motores y cada tarjeta posee únicamente 32 canales.

En la Figura 136 se muestra el controlador implementado.



Figura 136. Controlador para servomotores SSC-32U

Estos controladores funcionan con comandos seriales utilizando el siguiente protocolo:

```
# <ch> P <pw> S <spd> ... # <ch> P <pw> S <spd> T <time> <cr>
```

En el Cuadro 16 se especifica cada parámetro que interviene en el protocolo de comunicación.

Cuadro 16

Detalle del protocolo de comunicación

<CH>	Número de canal en decimal, 0-31
<PW>	Ancho de pulso en microsegundos, 500-2500
<SPD>	Velocidad de movimiento en uS por segundo para un canal (opcional)
<TIME>	Tiempo en milisegundos para todo el movimiento, afecta a todos los canales, 65535 es el máximo número (opcional)
<CR>	Caracter de retorno de acarreo, ASCII 13 (Requerido para iniciar la acción)
<ESC>	Cancelar la acción actual, ASCII 27

Fuente: Manual SSC-32

Por ejemplo si se desea mover el servomotor conectado al canal 5 del controlador se envía el comando:

```
#5P1500
```

Si no se especifica la velocidad el servomotor se mueve a la máxima velocidad.

4.2.3 Servomotores

Para la implementación de los servomotores se consideró principalmente el tamaño debido a las dimensiones de los componentes mecánicos. La fuerza es otro factor importante en la selección de un servomotor. A continuación se muestran los cálculos para determinar el torque mínimo que requieren los servomotores para poder mover las cremalleras y levantar los pines.

Siendo τ el torque, se tiene:

$$\tau = r \cdot F \quad (20)$$

$$\tau = 1.19 \text{ cm} \cdot 48.91 \times 10^{-3} \text{ N} \quad (21)$$

$$\tau = 58.2 \times 10^{-3} \text{ N} \cdot \text{cm} = 5.936 \times 10^{-3} \text{ Kg} \cdot \text{cm}$$

En base a los cálculos anteriores se establece que el torque mínimo que se requiere en los servomotores es de $5.936 \times 10^{-3} \text{ Kg} \cdot \text{cm}$. En el Cuadro 17 se muestra las características entre tres marcas de servomotores.

Cuadro 17

Cuadro comparativo de servomotores

CARACTERÍSTICAS	TURNIGY TS531A	TOWERPRO SG90	HITEC HS-5035HD
Tamaño	19 x 17.5 x 8 (mm)	29 x 23.1 x 12.2 (mm)	18.6 x 15.5 x 7.6 (mm)
Material de los engranajes	Plástico	Plástico	Metal
Torque	0.6 Kg·cm	1.8 Kg·cm	0.8 Kg·cm
Peso	3.7 g	9 g	4.5 g
Voltaje de alimentación	4.8 VDC – 6 VDC	4.8 VDC – 6 VDC	4.8 VDC – 6 VDC
Disponibilidad en el país	Si	Si	No
Precio	US \$ 3.99	US \$ 5.99	US \$ 30.99

En el Cuadro 17 se observa que todos los servomotores ofrecen el suficiente torque para poder mover las cremalleras. Considerando el tamaño,

el HS-5035HD es el más pequeño, pero ha sido descartado debido a su alto costo y la disponibilidad en el país. El servomotor seleccionado es el Turnigy TS531A, tanto por su tamaño y su costo. El inconveniente con este motor es la cantidad disponible en stock, puesto que solo se ha adquirido 32 de estos. Siendo este el caso, se ha utilizado ocho servomotores TowerPro SG90. Más adelante se muestra la adaptación de estos motores al sistema. En la Figura 137 y Figura 138 se muestran los servomotores utilizados en el prototipo.



Figura 137. Servomotor TowerPro SG90



Figura 138. Servomotor TURNIGY TS531A

4.2.4 Botones para el manejo del dispositivo

Los botones del dispositivo le permiten al usuario navegar entre las líneas del texto. El prototipo cuenta con cuatro botones:

- Avanzar: permite avanzar a la siguiente línea de 20 caracteres.
- Retroceder: permite volver a la línea anterior.

- Inicio: permite ir al comienzo del documento.
- Fin: permite ir al final del documento

En la Figura 139 se muestra los botones utilizados en el dispositivo.



Figura 139. Botones implementados en el prototipo

4.2.5 Convertidor USB-TTL

El convertidor utilizado es el CP2102 este convertidor permite la comunicación serial con el microcontrolador directamente utilizando niveles de voltaje TTL. Cuenta con seis pines de los cuales se utiliza cuatro, los de comunicación (RX y TX), y los de voltaje (5V y GND) para alimentar al microcontrolador y la etapa de control de las tarjetas para controlar los servomotores. En la Figura 140 se muestra el convertidor CP2102.



Figura 140. Convertidor USB - TTL C92102

4.2.6 Fuente de alimentación

La fuente de alimentación debe ser capaz de alimentar a los 40 servomotores ya que la parte de control es alimentada por la computadora por medio del convertidor USB-TTL.

Cada servomotor Turnigy consume 190 mA en el pico más alto de corriente, mientras que los TowerPro consumen 210 mA en el pico más alto de corriente, lo que significa que en el pico de corriente de todos los servomotores el consumo de corriente es de 7.76 A, cuando los servomotores están siendo alimentados con 6 VDC, además todos los servomotores consumen 30 mA cuando están conectados sin moverse, lo que genera un consumo de corriente pico de 7.79 A. Considerando los valores nominales de corriente con los que trabajan las fuentes de voltaje DC se ha considerado trabajar con una fuente de 10 A a 6 VDC.

En el Cuadro 18 se realiza una comparación entre dos opciones de fuentes de alimentación.

Cuadro 18

Cuadro comparativo de fuentes de alimentación

CARACTERÍSTICAS	ENFORCER ST-2406-10AQ	ENFORCER ST-2406-5AQ (2 FUENTES)
Voltaje de salida	6, 12, 24 VDC	6, 12, 24 VDC
Voltaje de entrada una salida de 6vdc	16 VAC/100 VA	16 VAC/100 VA
Corriente (continua)	9 A	4 A
Corriente (pico)	10 A	5 A
Disponibilidad en el país	No	Si

Debido a la a que no existe en el país la fuente de 10 A pico, se ha optado por utilizar dos fuentes de 5 A. De manera que cada fuente alimente veinte servomotores. Cada fuente necesita un transformador ya que el voltaje de

entrada es de 16 VAC. En la Figura 141. Fuente de alimentación se muestra la implementación de la fuente de alimentación para los servomotores.



Figura 141. Fuente de alimentación

Como se observa en la Figura 141 en la parte derecha se encuentran las fuentes de voltaje DC ST-2406-5AQ y en la parte izquierda cada transformador de 16 VAC.

En la Figura 142 se muestra el diagrama de conexión de las fuentes.

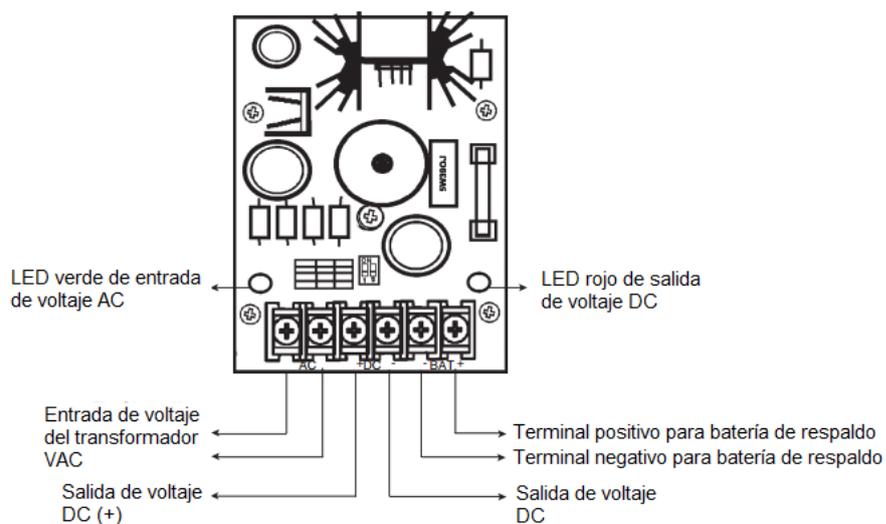


Figura 142. Diagrama de conexión de la fuente de voltaje

La fuente de voltaje posee un interruptor que permite energizar todos los servomotores una vez conectado a la red eléctrica. En la Figura 143 se muestra la fuente de voltaje con su interruptor. El lado del interruptor se conecta a la red eléctrica mientras que el otro lado se conecta al prototipo.



Figura 143. Fuente de alimentación

4.3 Integración

4.3.1 Integración de componentes mecánicos

El sistema está distribuido en dos niveles, cada lámina de acrílico sostiene los servomotores del nivel superior e inferior y estas láminas están adheridas a las celdas Braille y estas láminas se sujetan entre sí mediante tornillos y tuercas. En la Figura 144 se observa cómo se acoplan las láminas de acrílico a las celdas Braille.

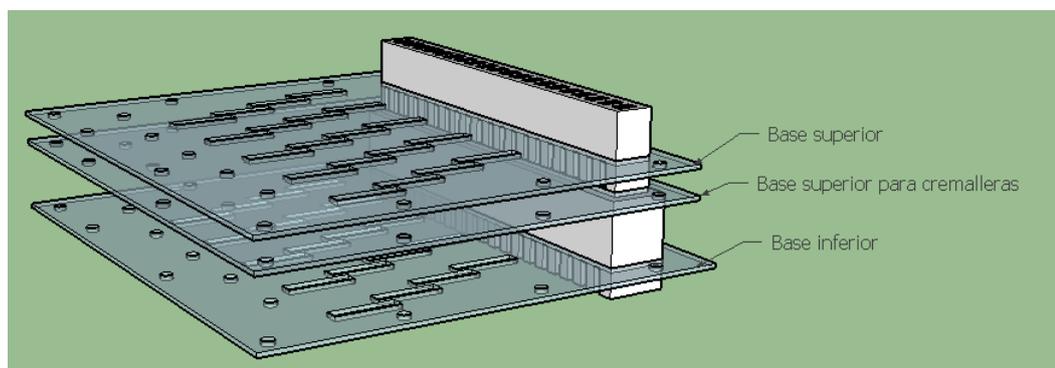


Figura 144. Acople de las láminas de acrílico a las celdas Braille

En la Figura 144 se puede observar orificios en las láminas, estos permiten sujetar las bases entre sí a través de tornillos.

En la Figura 145 se muestra la implementación de las láminas en las celdas Braille.



Figura 145. Implementación de la base superior para cremalleras

En la Figura 146 se muestra la implementación de la bases para servomotores y como se sujetan mediante tornillos.

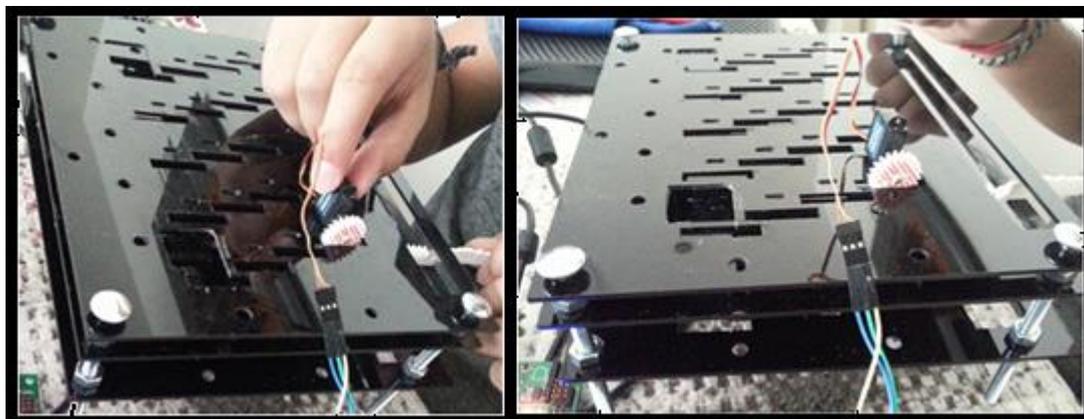


Figura 146. Implementación de bases para servomotores

En la Figura 147 se muestra la ubicación de las cremalleras en la base superior.

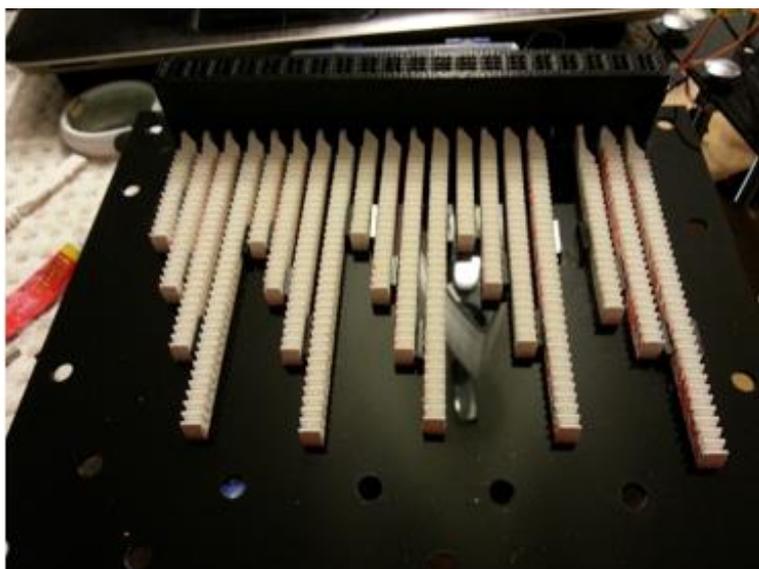


Figura 147. Implementación de las cremalleras en el nivel superior

En la base de la cremallera se encuentran ubicadas guías para que las cremalleras no se desvíen de su línea de trayectoria. En la Figura 148 se muestra una de las guías.

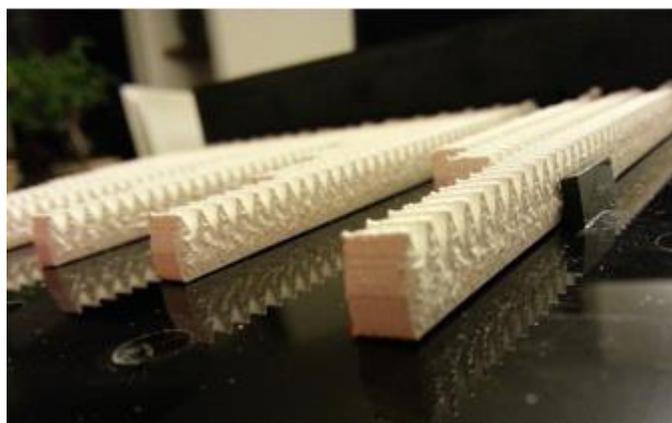


Figura 148. Guía ubicada a un costado de la cremallera

El movimiento de las cremalleras permite levantar los pines para formar los caracteres Braille. En la Figura 149 se muestran los pines en las celdas Braille.



Figura 149. Celdas Braille con pines para formar los caracteres

Los piñones son acoplados al engranaje de cada servomotor como se observa en la Figura 150.

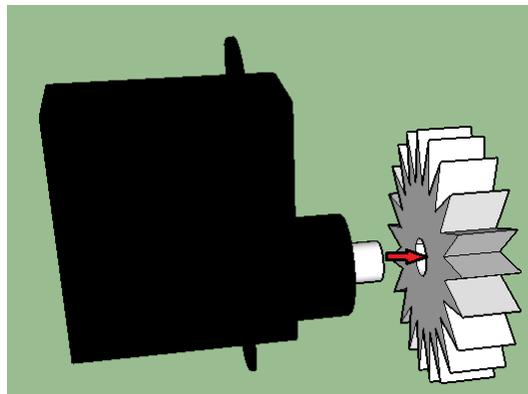


Figura 150. Acople del servomotor al piñón

Este acople permite que los servomotores puedan mover las cremalleras. En la Figura 151 se muestran los servomotores con el piñón.

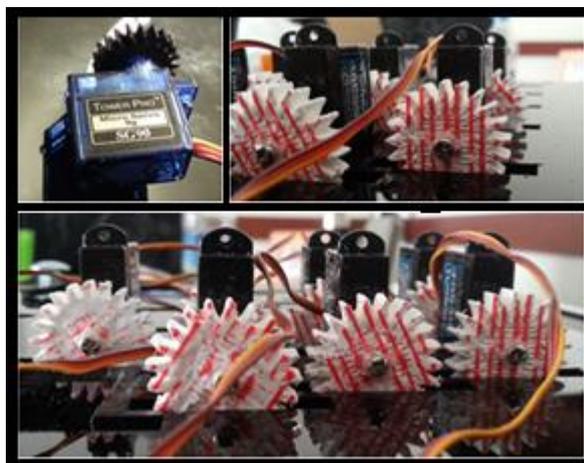


Figura 151. Servomotores con piñón

Mediante este acople los servomotores pueden manejar tres pines por cada celda Braille. Se encuentran ubicados en grupos de cuatro, como se observa en la Figura 152.

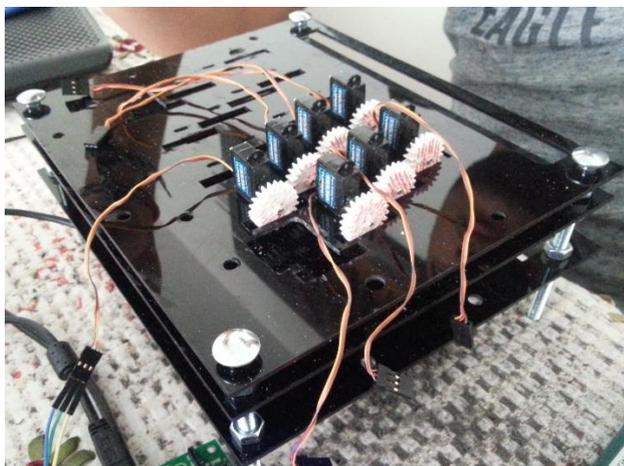


Figura 152. Ubicación de los servomotores

Esta distribución permite ahorrar espacio en el dispositivo sin que represente un inconveniente en el movimiento de las cremalleras.

En la Figura 153 y Figura 154 se muestra la implementación de los servomotores en el nivel superior e inferior.

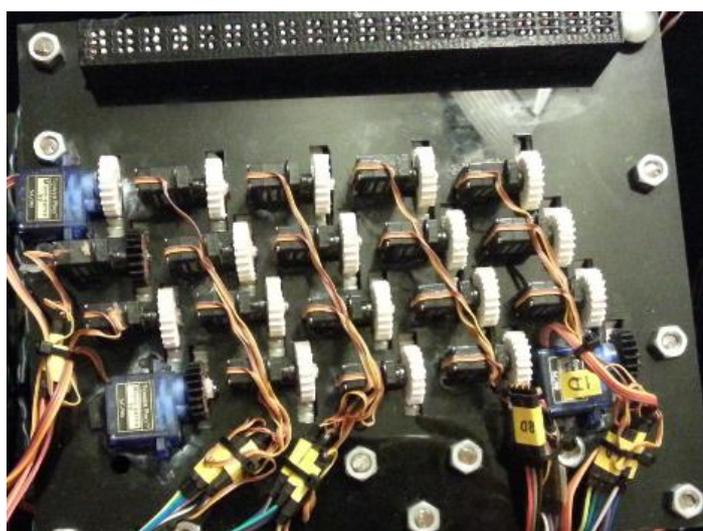


Figura 153. Implementación de los servomotores en el nivel superior

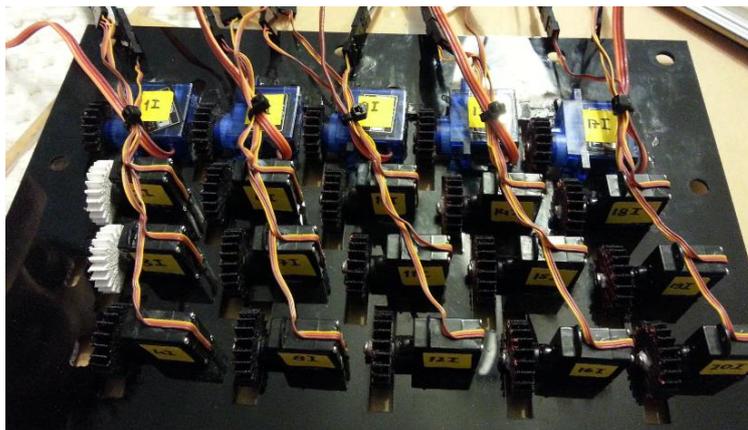


Figura 154. Implementación de los servomotores en el nivel inferior

Los servomotores del nivel superior controlan la fila derecha de las celdas, mientras que los servomotores del nivel inferior controlan la fila izquierda, es por esto que tienen etiquetas según el número de motor y la fila a la que pertenecen como se muestra en la Figura 155.

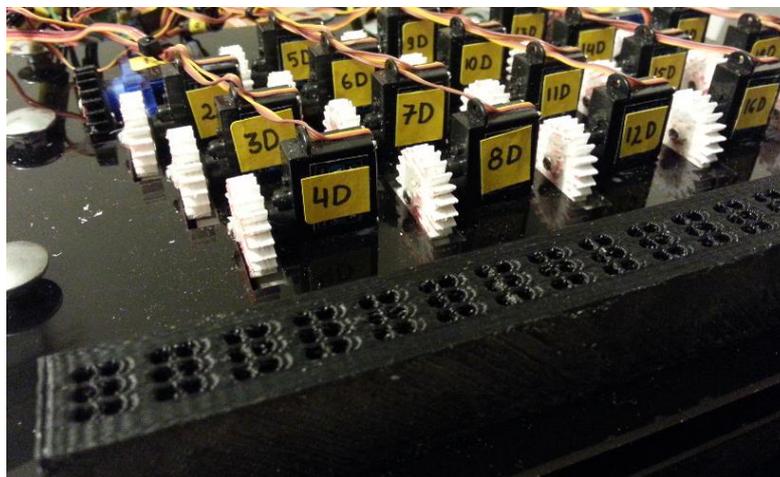


Figura 155. Numeración de los servomotores

En la Figura 156 se muestra un esquema de la asignación de filas de celdas Braille, para cada servomotor, tanto del nivel superior como el inferior. Cada servomotor está numerado seguido de la letra D si es del nivel superior (fila de la derecha) o I si es del nivel inferior (fila de la izquierda).

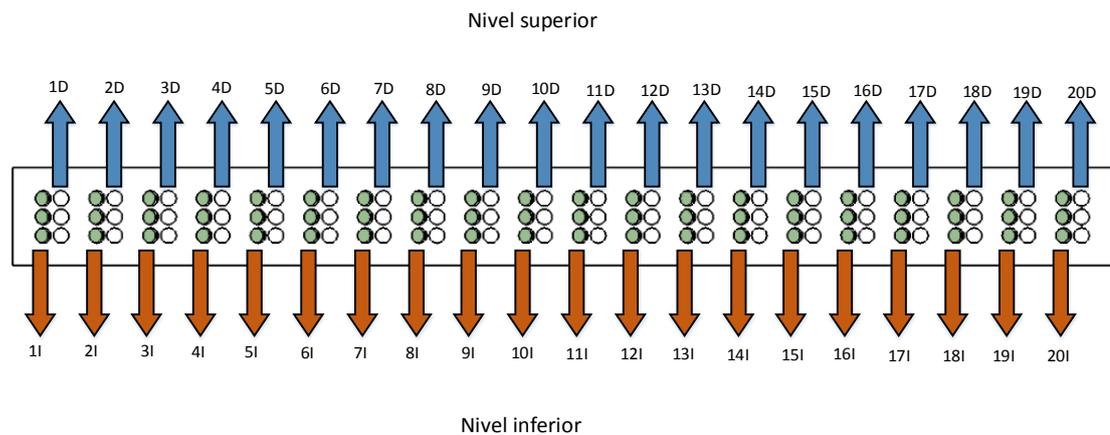


Figura 156. Distribución de filas y numeración de servomotores

En la Figura 157 se muestran los dos niveles de servomotores ubicados en sus respectivas bases.

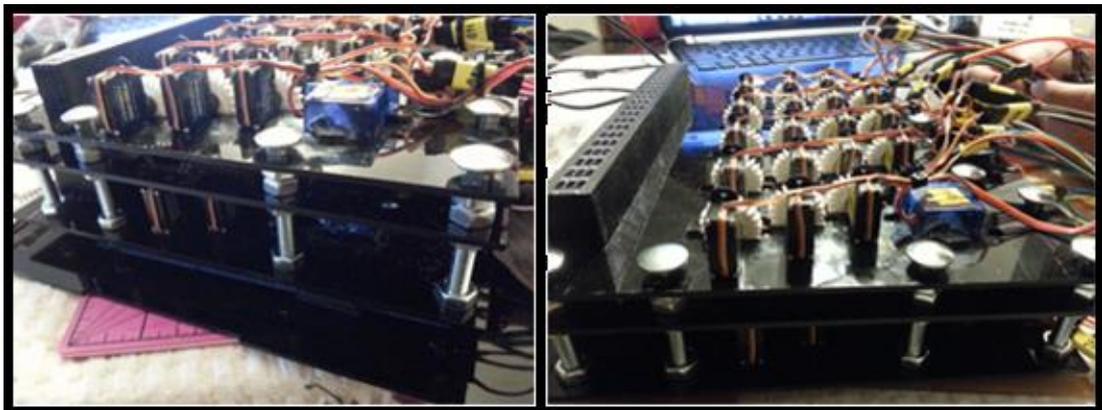


Figura 157. Ubicación de servomotores en dos niveles

Todos los componentes mecánicos mencionados anteriormente se ubican dentro de la caja, como se muestra en la Figura 158, dejando un espacio de 6 mm para ubicar los componentes electrónicos (placa del microcontrolador, controladores de servomotores y cableado).

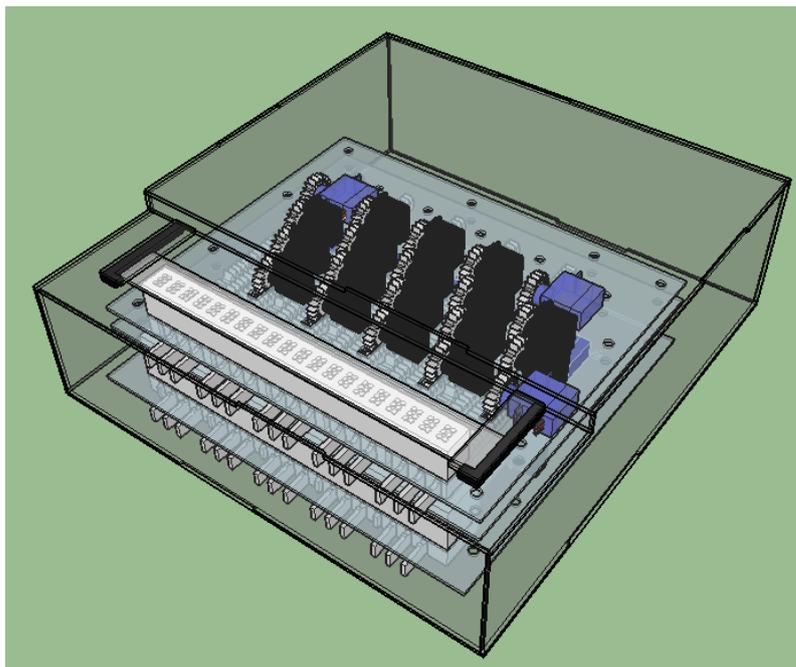


Figura 158. Ubicación de los componentes en la caja

4.3.2 Integración de componentes electrónicos

La integración de los componentes electrónicos se muestra en el diagrama de bloques de la Figura 159.

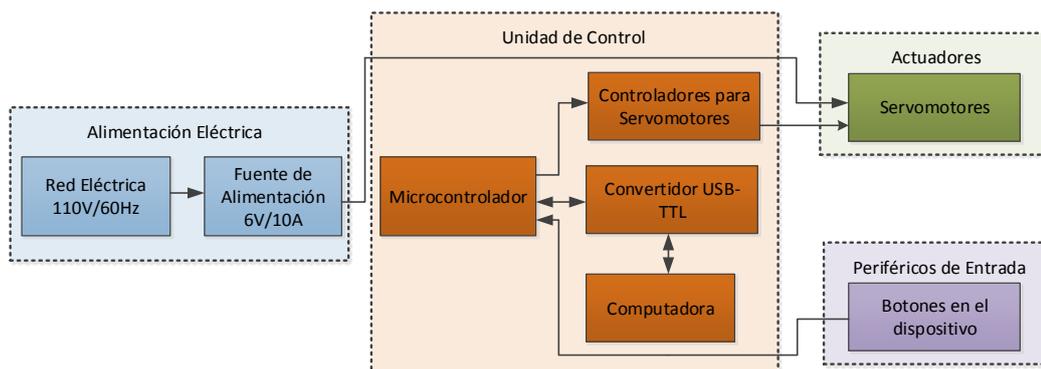


Figura 159. Diagrama de bloques del prototipo

Como se observa en la Figura 159 la integración se realiza por etapas, comenzando por la alimentación eléctrica. Como se mencionó anteriormente

se ha implementado dos fuentes de voltaje de 6 VDC con su respectivo transformador. El diagrama de conexión se muestra en la Figura 160.

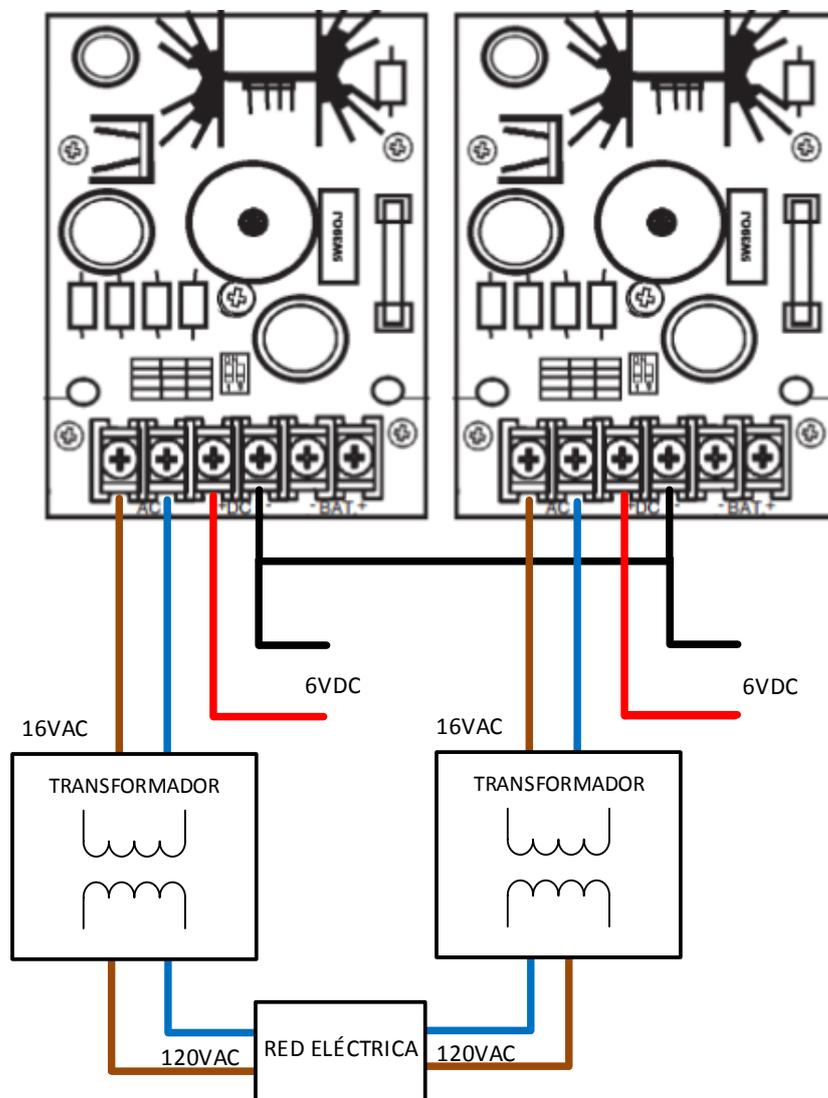


Figura 160. Diagrama de conexión de las fuentes de alimentación

Del diagrama de la Figura 160, se observa cómo se obtiene la salida de 6 VDC para la alimentación de los servomotores en cada controlador.

La conexión desde la salida de 6 VDC a los controladores de servomotores se observa en la Figura 161.

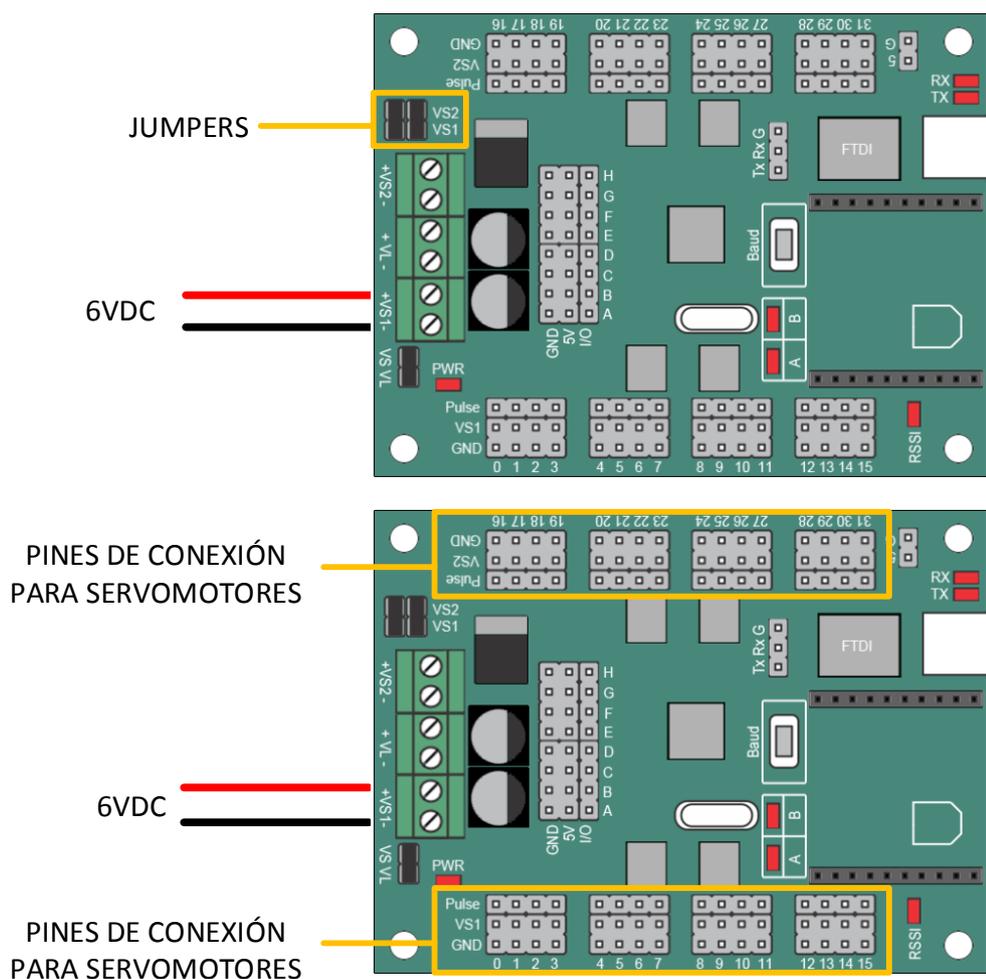


Figura 161. Diagrama de controladores para servomotores

En el diagrama de la Figura 161 se observa dos jumpers (VS1 y VS2) estos permiten que la alimentación de 6 VDC que entra por la bornera VS1 también alimente a la bornera VS2, de manera que todos los canales del 0 al 31 sean energizados únicamente por VS1 o VS2. El motivo de que existan dos jumpers es por cuestiones de distribución de corriente.

Continuando con la unidad de control, la alimentación de la parte de control así como las conexiones de comunicación se muestran en la Figura 162.

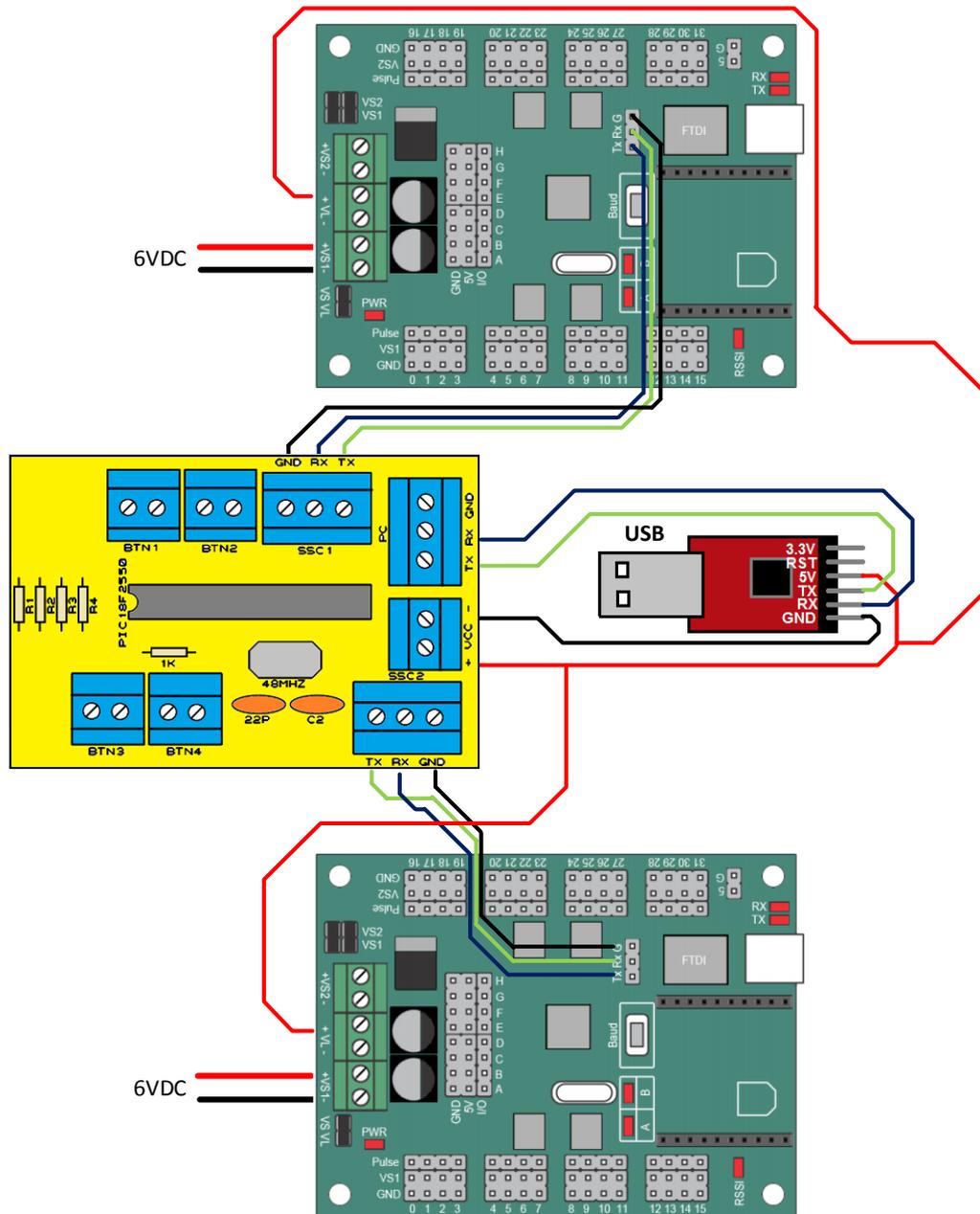


Figura 162. Diagrama de conexión de la unidad de control

Como se observa en la Figura 162, la alimentación de los canales para los servomotores, se conecta en la bornera con el nombre VS1, en los dos controladores. El microcontrolador está alimentado mediante la salida de 5V del convertidor USB-TTL, así como la entrada de voltaje TTL de los controladores, la cual se encuentra en la bornera con el nombre VL.

Para la comunicación existen tres borneras, dos para los controladores y una para el computador. El puerto UART del microcontrolador se utiliza para la comunicación con la computadora, en este caso los pines de transmisión y recepción se conectan directamente es decir TX del PIC a TX del convertidor USB y RX a RX como se muestra en la Figura 163.

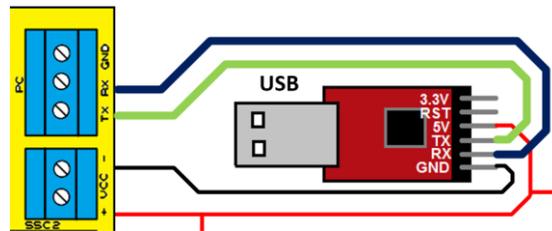


Figura 163. Diagrama de conexión entre el microcontrolador y el convertidor USB-TTL

La conexión entre el microcontrolador y los controladores se realiza por medio de puertos seriales generados por software en el microcontrolador, estos puertos se conectan de manera inversa a la conexión de la computadora, es decir TX a RX y RX a TX, como se observa en la Figura 164.

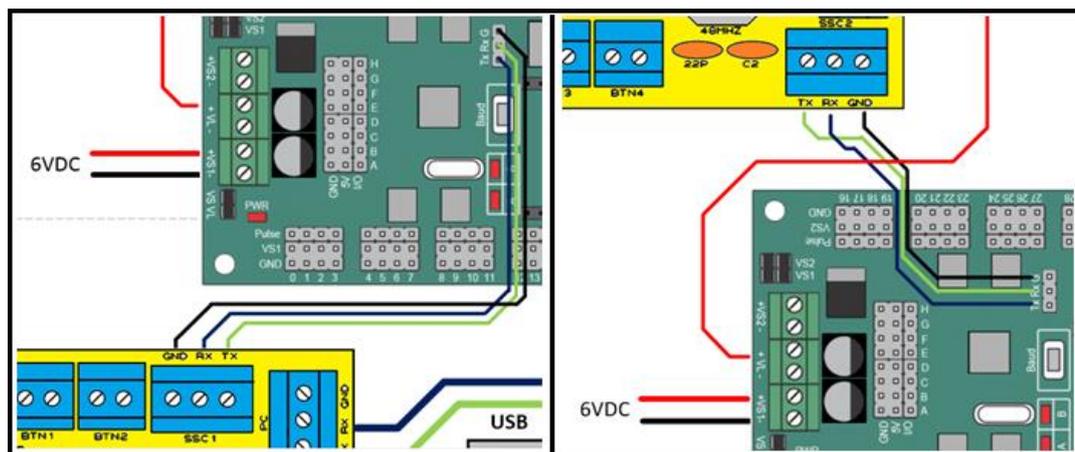


Figura 164. Conexión entre un puerto del microcontrolador y los controladores de servomotores

Los actuadores (servomotores) se conectan directamente a los controladores, cada tipo de servomotor tiene un código de colores en sus cables que determina cual es voltaje de alimentación, tierra y la señal de control.

En la Figura 165 se muestra la conexión de los servomotores a los controladores, tanto de los TURNIGY como de los TowerPro.

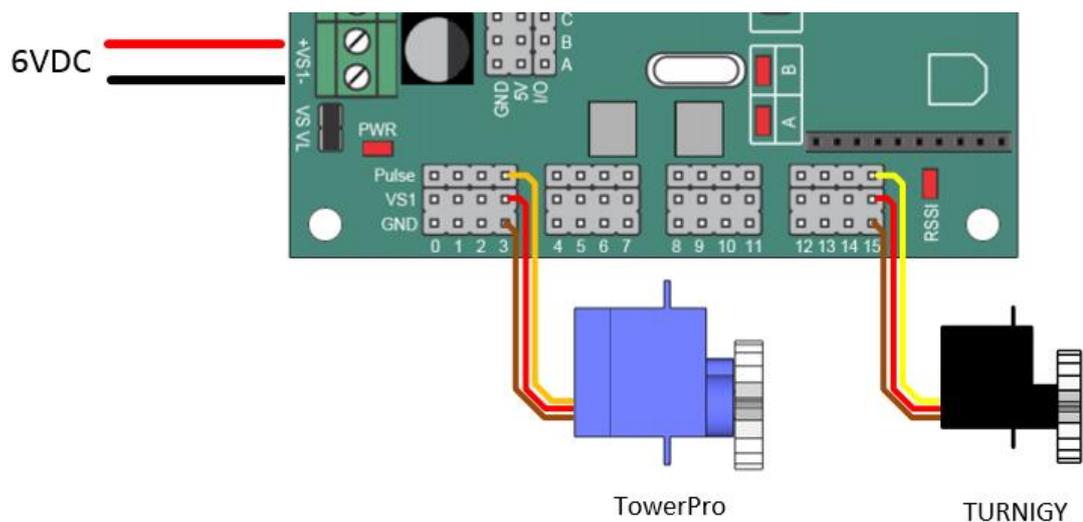


Figura 165. Diagrama de conexión de los servomotores al controlador.

Los servomotores están conectados desde el canal 1 en cada controlador. El controlador 1 (SSC1) se encarga de manejar los 20 motores de la parte izquierda del dispositivo del nivel superior e inferior. Los servomotores están conectados de manera intercalada, el servomotor conectado al canal uno es el primer motor del nivel inferior, en el canal dos está conectado el primer servomotor del nivel superior y así sucesivamente. Para este controlador los motores TowerPro están conectados en los canales: 1, 2, 9 y 17, el resto de canales hasta el número 20, están ocupados por los motores TURNIGY.

Para el controlador 2 (SSC2) se utiliza el mismo método de conexión pero para los 20 motores restantes tanto del nivel superior como en el inferior. En este caso los motores TowerPro están conectados en los canales: 5, 13, 14 y

20. En la Figura 166 se muestra los motores controlados por cada tarjeta (SSC1 y SSC2).

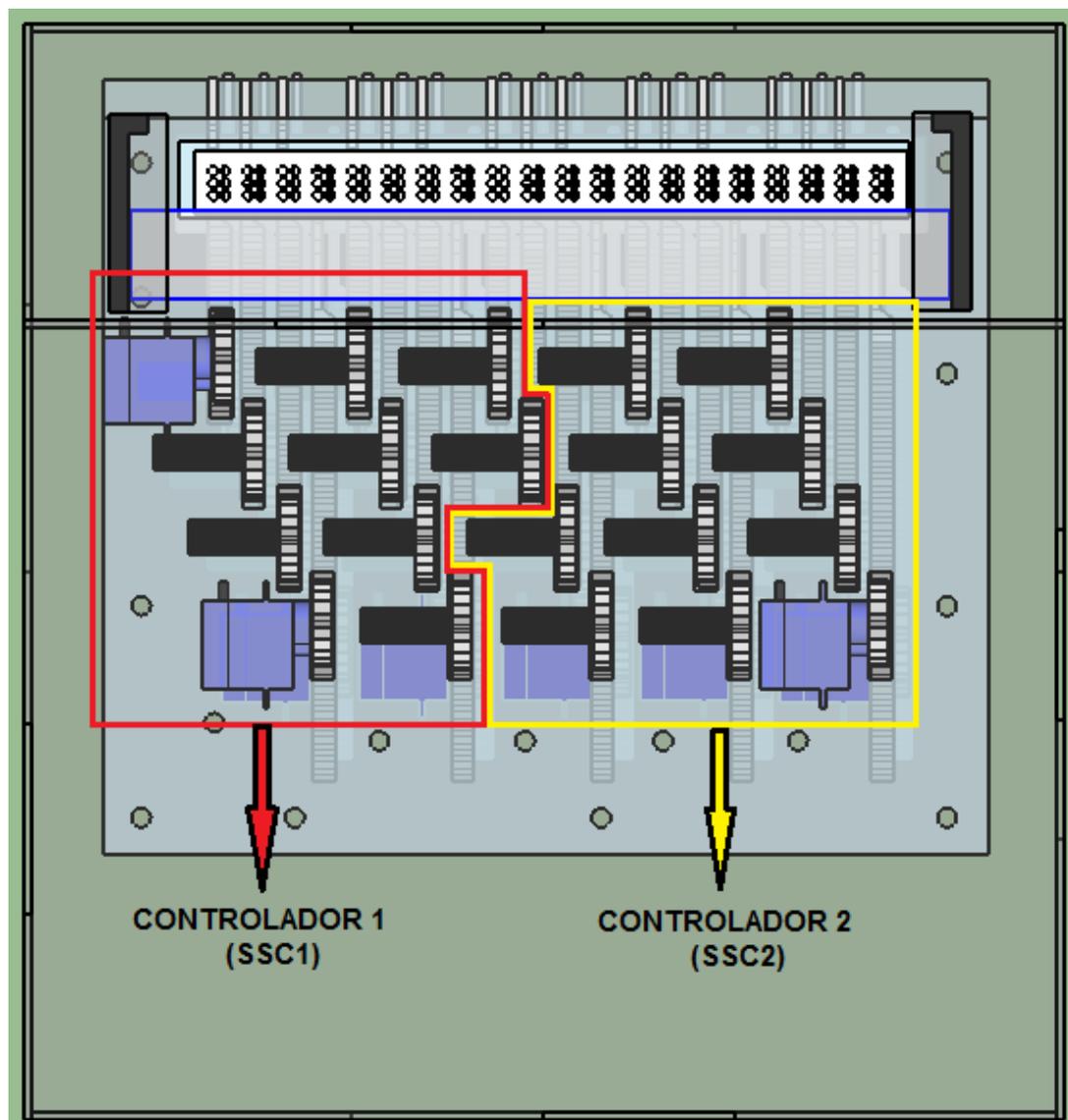


Figura 166. Servomotores controlados por cada tarjeta (SSC1 y SSC2)

En el Cuadro 19 se muestra una lista con los canales asignados a cada servomotor. Se utiliza las etiquetas de los servomotores dadas anteriormente.

Cuadro 19**Asignación de canales de controladores a servomotores**

CANAL DEL CONTROLADOR	CONTROLADOR 1 SSC1	CONTROLADOR 2 SSC2
1	1I	11I
2	1D	11D
3	2I	12I
4	2D	12D
5	3I	13I
6	3D	13D
7	4I	14I
8	4D	14D
9	5I	15I
10	5D	15D
11	6I	16I
12	6D	16D
13	7I	17I
14	7D	17D
15	8I	18I
16	8D	18D
17	9I	19I
18	9D	19D
19	10I	20I
20	10D	20D

Finalmente se tiene los periféricos de entrada (botones), que permiten para navegar en el documento, estos se conectan directamente a las borneras de la placa del microcontrolador.

El botón de avanzar se conecta a la bornera BTN1, el de retroceder a la bornera BTN2, el de inicio de documento a la BTN3 y la de fin de documento a la bornera BTN4. En la Figura 167 se muestra el diagrama de conexión de los botones.

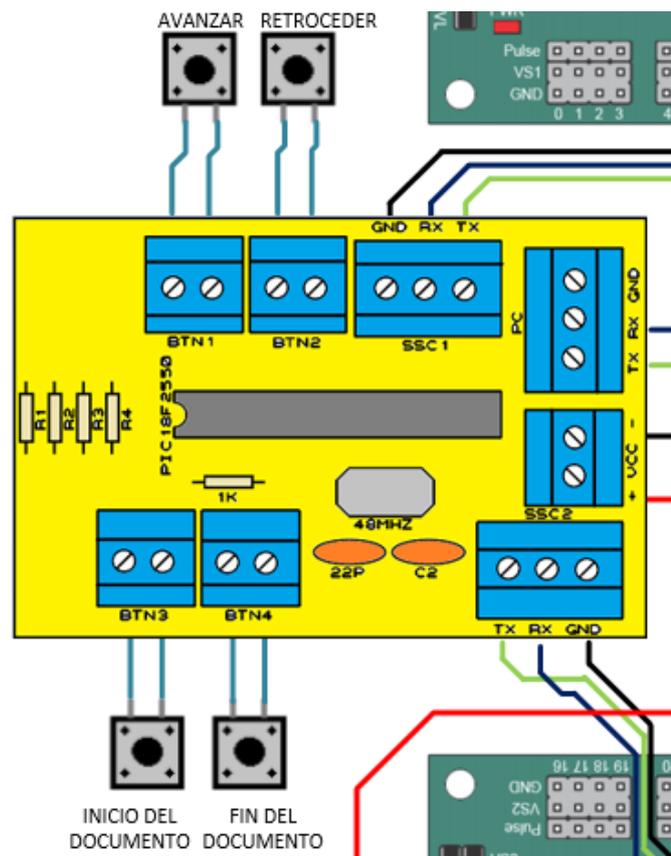


Figura 167. Diagrama de conexión de los botones

Los elementos mencionados anteriormente conforman la parte electrónica del dispositivo. En la Figura 168 se muestra un diagrama con todas las conexiones entre la etapa de alimentación, la unidad de control, los actuadores y los periféricos de entrada. En la Figura 169 se muestra las conexiones entre los elementos electrónicos implementados en el dispositivo. La fuente de alimentación está separada del prototipo y esta se conecta al dispositivo mediante un cable de poder.

Como se mencionó anteriormente la fuente de alimentación posee un interruptor que permite energizar los servomotores, por otra parte la etapa de control se energiza mediante el convertidor USB-TTL, la conexión del prototipo a la computadora se realiza mediante un cable USB tipo A macho – macho, ya que la caja posee un puerto USB hembra, el cual está conectado al convertidor USB – TTL, como se observa en la Figura 170.

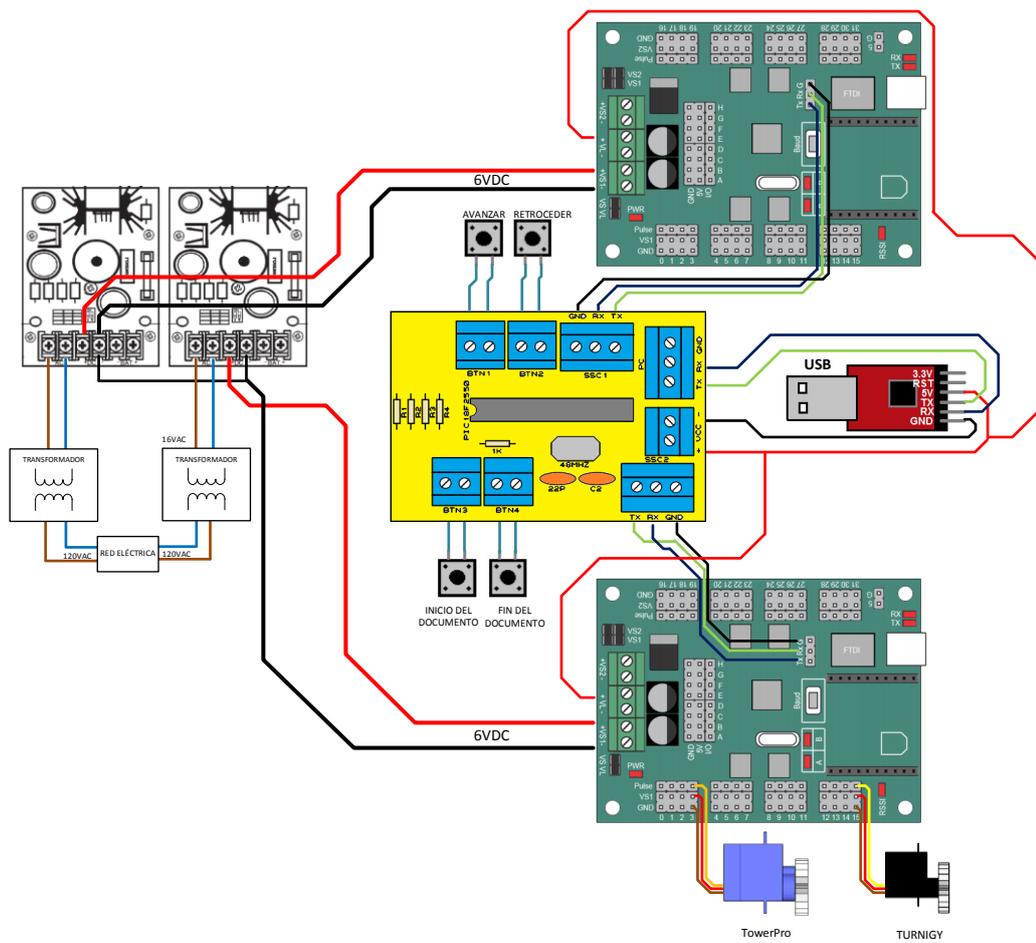


Figura 168. Diagrama de conexión de los componentes electrónicos

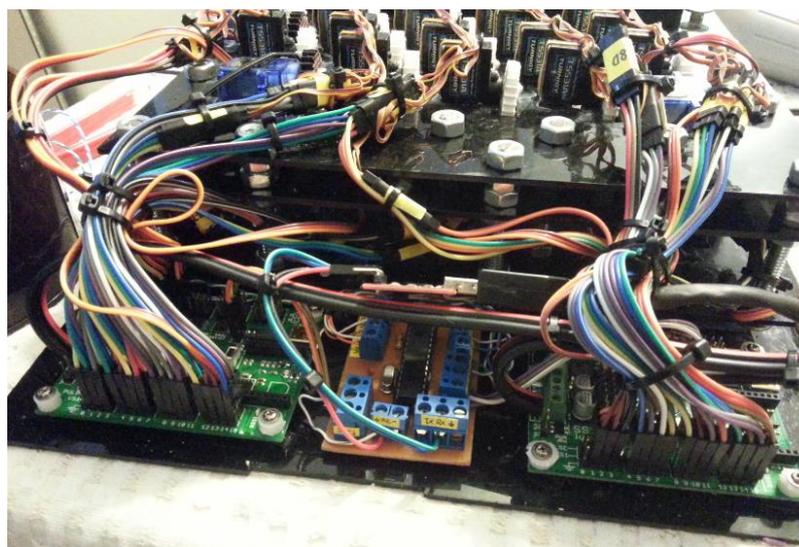


Figura 169. Conexiones de los componentes electrónicos



Figura 170. Puerto USB del prototipo

Todos estos elementos se encuentran contenidos en la caja del prototipo como se muestra en la Figura 171.



Figura 171. Caja del prototipo

En la Figura 172 se muestra el prototipo terminado.



Figura 172. Prototipo terminado

4.3.3 Integración de componentes de software

La comunicación entre el ordenador y los componentes electrónicos se realiza mediante tres puertos seriales. El primer puerto es para la comunicación entre el computador y el microcontrolador, la cual se realiza utilizando el convertidor USB-TTL y el puerto UART del microcontrolador.



Figura 173. Archivo abierto en el programa

Una vez abierto el programa y escogido el archivo, la computadora envía por un puerto serial una trama con la primera línea del texto que se desea leer. En la Figura 173 se muestra una imagen de un archivo abierto en el programa.

Lo que se envía primero es el título del archivo transformado a Braille de computador, como se observa al final de la imagen de la Figura 173, pero este texto se envía transformado a una cadena de 41 caracteres que representan las posiciones de los 40 servomotores y un caracter de confirmación de trama.

Para el texto en Braille “.DOCUMENTO DE ”, la trama que se envía es:

“ADBEDCBBDHDBBCDEGEDCAABEBCAAAAAAAAAAAAAAY”

Cada caracter de la trama es transformado a una posición de los servomotores mediante el microcontrolador, la transformación se la realiza utilizando el valor en ASCII de cada caracter que se recibe.

El valor transformado es enviado mediante los otros dos puertos seriales hacia los controladores, estos valores son calculados considerando que las posiciones para servomotores son cantidades entre 600 y 2200 (ancho de pulso en microsegundos), ya que los controladores trabajan con esas cifras. Los cálculos son los siguientes:

- Para servomotores TURNIGY

Este tipo de servomotor con el piñón acoplado, mueve la cremallera 28 mm cuando se mueve desde la posición 2200 hasta 600.

Conociendo que la cremallera debe moverse en pasos de 3.15 mm para poder pasar de una posición a otra, se realizan los siguientes cálculos:

$$2200 - 600 = 1600 \quad (22)$$

Cuando el servomotor recorre un paso de 1600 microsegundos mueve la cremallera 28 mm. Para conocer de qué valor debe ser el paso para que la cremallera se mueva 3.15 mm se realiza una regla de tres.

$$x = \frac{1600 \times 3.15}{28} \quad (23)$$

$$x = 180$$

Esto significa que se debe avanzar en pasos de 180 microsegundos para que la cremallera pueda pasar de una posición a otra. La primera posición de estos servomotores es 2200 por lo que las ocho posiciones con las que se trabaja se forman con los siguientes pasos:

- 2200 (posición 0)
- 2020 (posición 1)
- 1840 (posición 2)
- 1660 (posición 3)
- 1480 (posición 4)
- 1300 (posición 5)
- 1120 (posición 6)
- 940 (posición 7)

Las tramas que recibe el microcontrolador contienen letras entre la A y la H, como se mencionó se trabaja con los valores en ASCII de dichas letras para formar las posiciones calculadas anteriormente. Siendo el número 65 el valor en ASCII de la letra A (posición 0), se tiene:

$$posición = 2200 - (180 \cdot (Letra - 65)) \quad (24)$$

De esta manera se obtiene las posiciones de los servomotores, estas posiciones se envían a los controladores como tramas seriales.

- Para servomotores TowerPro

Este tipo de servomotor con el piñón acoplado, mueve la cremallera 33.6 mm cuando se mueve desde la posición 2200 hasta 600. Considerando los mismos requerimientos mencionados anteriormente, los cálculos se realizan de manera similar:

$$x = \frac{1600 \times 3.15}{33.6} \quad (23)$$

$$x = 150$$

Esto servomotores deben avanzar en pasos de 150 microsegundos para que la cremallera pueda pasar de una posición a otra.

La primera posición de estos servomotores es 600 por lo que las ocho posiciones son:

- 600 (posición 0)
- 750 (posición 1)
- 900 (posición 2)
- 1050 (posición 3)
- 1200 (posición 4)
- 1350 (posición 5)
- 1500 (posición 6)
- 1650 (posición 7)

Para estos servomotores cambia la fórmula para generar los pasos puesto que la posición inicial es 600, considerando este cambio se tiene:

$$posición = 600 + (150 \cdot (Letra - 65)) \quad (24)$$

Existen solo dos servomotores donde no se aplica ninguna de estas fórmulas, ya que al realizar las pruebas estos servos respondían de diferente

manera, estos servos son el 5I (TowerPro) y el 17I (TURNIGY), ubicados en el nivel inferior.

Cada caracter de la trama representa una posición para cada servomotor, por lo que se utiliza dos letras de la trama para formar un caracter Braille, como se muestra en la Figura 174.

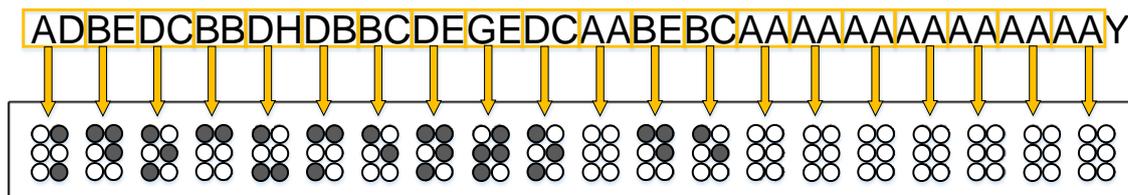


Figura 174. Asignación de caracteres Braille por cada letra de la trama recibida

En el Cuadro 20 y Cuadro 21 se muestra la asignación de cada letra de la trama para cada servomotor, utilizando las etiquetas mencionadas anteriormente.

Cuadro 20

Asignación de caracteres de la trama para el controlador 1

Nº DE CARACTER EN LA TRAMA	SERVOMOTOR DEL CONTROLADOR 1	Nº DE CARACTER EN LA TRAMA	SERVOMOTOR DEL CONTROLADOR 1
1	1I	11	6I
2	1D	12	6D
3	2I	13	7I
4	2D	14	7D
5	3I	15	8I
6	3D	16	8D
7	4I	17	9I
8	4D	18	9D
9	5I	19	10I
10	5D	20	10D

Cuadro 21**Asignación de caracteres de la trama para el controlador 2**

Nº DE CARACTER EN LA TRAMA	SERVOMOTOR DEL CONTROLADOR 2	Nº DE CARACTER EN LA TRAMA	SERVOMOTOR DEL CONTROLADOR 2
21	11I	31	16I
22	11D	32	16D
23	12I	33	17I
24	12D	34	17D
25	13I	35	18I
26	13D	36	18D
27	14I	37	19I
28	14D	38	19D
29	15I	39	20I
30	15D	40	20D

CAPÍTULO 5

PRUEBAS Y ANÁLISIS DE RESULTADOS

5.1 Pruebas de servomotores

5.1.1 Distancia recorrida por las cremalleras

Una vez establecidos los pasos para el movimiento de los servomotores, se procede a realizar las pruebas del desplazamiento de las cremalleras, verificando si la distancia recorrida es la requerida para pasar de una posición de pines a otra. Para esto se empieza desde la posición 0 para cada servomotor, en el caso de los TURNIGY es 2200 μ s y para los TowerPro es 600 μ s.

En el Cuadro 22 y Cuadro 23 se muestra los datos obtenidos para el controlador 1 y el controlador 2 respectivamente, la distancia está dada en milímetros y las posiciones en microsegundos. Las distancia equivalente a cada posición es considerada desde la posición inicial (posición 0).

Cuadro 22

Datos obtenidos para el controlador 1

N° DE SERVOMOTOR	MARCA	POS. 0/ DIST.	POS. 1/ DIST.	POS. 2/ DIST.	POS. 3/ DIST.	POS. 4/ DIST.	POS. 5/ DIST.	POS. 6/ DIST.	POS. 7/ DIST.
1I	Towerpro	600 / 0	750 / 3.15	900 / 6.30	1050 / 9.45	1200 / 12.60	1350 / 15.75	1500 / 18.90	1650 / 22.05
1D	Towerpro	600 / 0	750 / 3.20	900 / 6.40	1050 / 9.55	1200 / 12.75	1350 / 15.85	1500 / 18.95	1650 / 22.10
2I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.35	1660 / 9.50	1480 / 12.60	1300 / 15.75	1120 / 18.95	940 / 22.05
2D	Turnigy	2200 / 0	2020 / 3.14	1840 / 6.28	1660 / 9.42	1480 / 12.56	1300 / 15.70	1120 / 18.85	940 / 22.00
3I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.30	1660 / 9.45	1480 / 12.60	1300 / 15.75	1120 / 18.90	940 / 22.05
3D	Turnigy	2200 / 0	2020 / 3.12	1840 / 6.25	1660 / 9.42	1480 / 12.55	1300 / 15.71	1120 / 18.86	940 / 22.01
4I	Turnigy	2200 / 0	2020 / 3.16	1840 / 6.32	1660 / 9.47	1480 / 12.62	1300 / 15.77	1120 / 18.92	940 / 22.07
4D	Turnigy	2200 / 0	2020 / 3.16	1840 / 6.33	1660 / 9.47	1480 / 12.62	1300 / 15.76	1120 / 18.91	940 / 22.06
5I	Towerpro	600 / 0	750 / 3.18	900 / 6.33	1050 / 9.48	1200 / 12.63	1350 / 15.78	1500 / 18.93	1650 / 22.08
5D	Turnigy	2200 /	2020 /	1840 /	1660 /	1480 /	1300 /	1120 /	940 /

Continua 

		0	3.15	6.30	9.44	12.59	15.74	18.90	22.05
6I	Turnigy	2200 / 0	2020 / 3.14	1840 / 6.28	1660 / 9.43	1480 / 12.57	1300 / 15.72	1120 / 18.87	940 / 22.03
6D	Turnigy	2200 / 0	2020 / 3.14	1840 / 6.28	1660 / 9.42	1480 / 12.56	1300 / 15.71	1120 / 18.86	940 / 22.02
7I	Turnigy	2200 / 0	2020 / 3.14	1840 / 6.29	1660 / 9.44	1480 / 12.58	1300 / 15.73	1120 / 18.88	940 / 22.04
7D	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.30	1660 / 9.45	1480 / 12.61	1300 / 15.76	1120 / 18.91	940 / 22.05
8I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.29	1660 / 9.44	1480 / 12.59	1300 / 15.74	1120 / 18.89	940 / 22.04
8D	Turnigy	2200 / 0	2020 / 3.16	1840 / 6.32	1660 / 9.47	1480 / 12.62	1300 / 15.77	1120 / 18.92	940 / 22.07
9I	Towerpro	600 / 0	750 / 3.18	900 / 6.33	1050 / 9.48	1200 / 12.63	1350 / 15.78	1500 / 18.93	1650 / 22.08
9D	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.30	1660 / 9.45	1480 / 12.61	1300 / 15.76	1120 / 18.91	940 / 22.05
10I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.29	1660 / 9.44	1480 / 12.59	1300 / 15.74	1120 / 18.89	940 / 22.04
10D	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.30	1660 / 9.45	1480 / 12.60	1300 / 15.75	1120 / 18.90	940 / 22.05

Cuadro 23

Datos obtenidos para el controlador 2

N° DE SERV OMO TOR	MARCA	POS. 0/ DIST.	POS. 1/ DIST.	POS. 2/ DIST.	POS. 3/ DIST.	POS. 4/ DIST.	POS. 5/ DIST.	POS. 6/ DIST.	POS. 7/ DIST.
11I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.35	1660 / 9.50	1480 / 12.60	1300 / 15.75	1120 / 18.95	940 / 22.05
11D	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.30	1660 / 9.45	1480 / 12.60	1300 / 15.75	1120 / 18.90	940 / 22.05
12I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.35	1660 / 9.50	1480 / 12.60	1300 / 15.75	1120 / 18.95	940 / 22.05
12D	Turnigy	2200 / 0	2020 / 3.14	1840 / 6.28	1660 / 9.42	1480 / 12.56	1300 / 15.70	1120 / 18.85	940 / 22.00
13I	Towerpro	2200 / 0	2020 / 3.07	1840 / 6.18	1660 / 9.32	1480 / 12.30	1300 / 15.40	1120 / 18.42	940 / 21.45
13D	Turnigy	2200 / 0	2020 / 3.13	1840 / 6.25	1660 / 9.42	1480 / 12.55	1300 / 15.71	1120 / 18.86	940 / 22.01
14I	Turnigy	2200 / 0	2020 / 3.16	1840 / 6.32	1660 / 9.47	1480 / 12.62	1300 / 15.77	1120 / 18.92	940 / 22.07
14D	Turnigy	2200 / 0	2020 / 3.16	1840 / 6.33	1660 / 9.47	1480 / 12.62	1300 / 15.76	1120 / 18.91	940 / 22.06
15I	Turnigy	2200 / 0	2020 / 3.13	1840 / 6.25	1660 / 9.42	1480 / 12.55	1300 / 15.71	1120 / 18.86	940 / 22.01
15D	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.30	1660 / 9.44	1480 / 12.59	1300 / 15.74	1120 / 18.90	940 / 22.05
16I	Turnigy	2200 / 0	2020 / 3.14	1840 / 6.28	1660 / 9.43	1480 / 12.57	1300 / 15.72	1120 / 18.87	940 / 22.03
16D	Turnigy	2200 / 0	2020 / 3.14	1840 / 6.28	1660 / 9.42	1480 / 12.56	1300 / 15.71	1120 / 18.86	940 / 22.02
17I	Towerpro	600 / 0	750 / 3.14	900 / 6.29	1050 / 9.44	1200 / 12.58	1350 / 15.73	1500 / 18.88	1650 / 22.04
17D	Towerpro	600 / 0	750 / 3.15	900 / 6.30	1050 / 9.45	1200 / 12.61	1350 / 15.76	1500 / 18.91	1650 / 22.05
18I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.29	1660 / 9.44	1480 / 12.59	1300 / 15.74	1120 / 18.89	940 / 22.04

Continua 

18D	Turnigy	2200 / 0	2020 / 3.16	1840 / 6.32	1660 / 9.47	1480 / 12.62	1300 / 15.77	1120 / 18.92	940 / 22.07
19I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.35	1660 / 9.65	1480 / 12.95	1300 / 16.02	1120 / 19.05	940 / 22.25
19D	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.30	1660 / 9.45	1480 / 12.61	1300 / 15.76	1120 / 18.91	940 / 22.05
20I	Turnigy	2200 / 0	2020 / 3.15	1840 / 6.29	1660 / 9.44	1480 / 12.59	1300 / 15.74	1120 / 18.89	940 / 22.04
20D	Towerpro	600 / 0	750 / 3.15	900 / 6.30	1050 / 9.45	1200 / 12.60	1350 / 15.75	1500 / 18.90	1650 / 22.05

Una vez obtenidos los datos, se procede a calcular los errores existentes en el desplazamiento de cada cremallera, las distancias definidas en el diseño para cada posición son las siguientes:

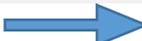
- Posición 0: 0 mm
- Posición 1: 3.15 mm
- Posición 2: 6.30 mm
- Posición 3: 9.45 mm
- Posición 4: 12.60 mm
- Posición 5: 15.75 mm
- Posición 6: 18.90 mm
- Posición 7: 22.05 mm

Todas estas distancias son calculadas desde la posición inicial. En el Cuadro 24 y Cuadro 25 se muestra los errores en el desplazamiento de las cremalleras respecto a las distancias definidas en el diseño.

Cuadro 24

Cuadro de errores para los motores del controlador 1

N° DE SERV OMO TOR	MARCA	POS. 0	POS. 1	POS. 2	POS. 3	POS. 4	POS. 5	POS. 6	POS. 7
1I	Towerpro	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
1D	Towerpro	0 %	1.58 %	1.58 %	1.06 %	1.19 %	0.63 %	0.26 %	0.23 %
2I	Turnigy	0 %	0 %	0.79 %	0.53 %	0.00 %	0 %	0.26 %	0 %
2D	Turnigy	0 %	0.31 %	0.32 %	0.32 %	0.32 %	0.32 %	0.26 %	0.23 %
3I	Turnigy	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
3D	Turnigy	0 %	0.95 %	0.79 %	0.32 %	0.40 %	0.25 %	0.21 %	0.18 %
4I	Turnigy	0 %	0.31 %	0.32 %	0.21 %	0.16 %	0.13 %	0.11 %	0.09 %
4D	Turnigy	0 %	0.31 %	0.48 %	0.21 %	0.16 %	0.06 %	0.05 %	0.05 %
5I	Towerpro	0 %	0.95 %	0.48 %	0.32 %	0.24 %	0.19 %	0.16 %	0.14 %

Continua 

5D	Turnigy	0 %	0 %	0 %	0.11 %	0.08 %	0.06 %	0 %	0 %
6I	Turnigy	0 %	0.32 %	0.32 %	0.21 %	0.24 %	0.19 %	0.16 %	0.09 %
6D	Turnigy	0 %	0.32 %	0.32 %	0.32 %	0.32 %	0.25 %	0.21 %	0.14 %
7I	Turnigy	0 %	0.31 %	0.16 %	0.11 %	0.16 %	0.13 %	0.11 %	0.05 %
7D	Turnigy	0 %	0 %	0 %	0 %	0.08 %	0.06 %	0.05 %	0 %
8I	Turnigy	0 %	0 %	0.16 %	0.11 %	0.08 %	0.06 %	0.05 %	0.05 %
8D	Turnigy	0 %	0.32 %	0.32 %	0.21 %	0.16 %	0.13 %	0.11 %	0.09 %
9I	Towerpro	0 %	0.95 %	0.48 %	0.32 %	0.24 %	0.19 %	0.16 %	0.14 %
9D	Turnigy	0 %	0 %	0 %	0 %	0.08 %	0.06 %	0.05 %	0 %
10I	Turnigy	0 %	0 %	0.16 %	0.11 %	0.08 %	0.06 %	0.05 %	0.05 %
10D	Turnigy	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %

Cuadro 25

Cuadro de errores para los motores del controlador 2

N° DE SERV OMO TOR	MARCA	POS. 0	POS. 1	POS. 2	POS. 3	POS. 4	POS. 5	POS. 6	POS. 7
11I	Turnigy	0 %	0 %	0.79 %	0.53 %	0 %	0 %	0.26 %	0 %
11D	Turnigy	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
12I	Turnigy	0 %	0 %	0.79 %	0.53 %	0 %	0 %	0.26 %	0 %
12D	Turnigy	0 %	0.32 %	0.32 %	0.32 %	0.32 %	0.32 %	0.26 %	0.23 %
13I	Towerpro	0 %	2.54 %	1.90 %	1.38 %	2.38 %	2.22 %	2.54 %	2.72 %
13D	Turnigy	0 %	0.63 %	0.79 %	0.32 %	0.40 %	0.25 %	0.21 %	0.18 %
14I	Turnigy	0 %	0.32 %	0.32 %	0.21 %	0.16 %	0.13 %	0.11 %	0.09 %
14D	Turnigy	0 %	0.32 %	0.48 %	0.21 %	0.16 %	0.06 %	0.05 %	0.05 %
15I	Turnigy	0 %	0.63 %	0.79 %	0.32 %	0.40 %	0.25 %	0.21 %	0.18 %
15D	Turnigy	0 %	0 %	0 %	0.11 %	0.08 %	0.06 %	0 %	0 %
16I	Turnigy	0 %	0.32 %	0.32 %	0.21 %	0.24 %	0.19 %	0.16 %	0.09 %
16D	Turnigy	0 %	0.32 %	0.32 %	0.32 %	0.32 %	0.25 %	0.21 %	0.14 %
17I	Towerpro	0 %	0.32 %	0.16 %	0.11 %	0.16 %	0.13 %	0.11 %	0.05 %
17D	Towerpro	0 %	0 %	0 %	0 %	0.08 %	0.06 %	0.05 %	0 %
18I	Turnigy	0 %	0 %	0.16 %	0.11 %	0.08 %	0.06 %	0.05 %	0.05 %
18D	Turnigy	0 %	0.32 %	0.32 %	0.21 %	0.16 %	0.13 %	0.11 %	0.09 %
19I	Turnigy	0 %	0 %	0.79 %	2.12 %	2.78 %	1.71 %	0.79 %	0.91 %
19D	Turnigy	0 %	0 %	0 %	0 %	0.08 %	0.06 %	0.05 %	0 %
20I	Turnigy	0 %	0 %	0.16 %	0.11 %	0.08 %	0.06 %	0.05 %	0.05 %
20D	Towerpro	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %

Como se observa en el Cuadro 24 y Cuadro 25 el error más grande es de más del 2%, el cual aparece en los motores 13I y en el 19I, en cuanto a los demás el margen de error se mantiene menor a 1.6%. Debido al error de los motores mencionados, los pasos han variado para que puedan recorrer la distancia necesaria para poder pasar de una posición a otra.

Para el motor TowerPro 13I se han escogido los siguientes pasos:

- Posición 0: 700 μ s
- Posición 1: 950 μ s
- Posición 2: 1150 μ s
- Posición 3: 1350 μ s
- Posición 4: 1500 μ s
- Posición 5: 1650 μ s
- Posición 6: 1900 μ s
- Posición 7: 2100 μ s

Utilizando estos pasos se obtiene las medidas mostradas en el Cuadro 26.

Cuadro 26

Resultados con los nuevos pasos para el servomotor 13I

N° DE SERVOMOTOR	POS. 0/ DIST.	POS. 1/ DIST.	POS. 2/ DIST.	POS. 3/ DIST.	POS. 4/ DIST.	POS. 5/ DIST.	POS. 6/ DIST.	POS. 7/ DIST.
13I	700 / 0	950 / 3.15	1150 / 6.32	1350 / 9.47	1500 / 12.63	1650 / 15.76	1900 / 18.93	2100 / 20.07

Los errores obtenidos para estos valores se muestran en el Cuadro 27.

Cuadro 27

Errores para los nuevos pasos del servomotor 13I

N° DE SERVOMOTOR	POS. 0	POS. 1	POS. 2	POS. 3	POS. 4	POS. 5	POS. 6	POS. 7
13I	0 %	0 %	0.31 %	0.21 %	0.24 %	0.06 %	0.15 %	0.09 %

Como se observa en la tabla anterior el margen de error máximo es de 0.31% por lo que estos valores funcionan correctamente para este servomotor.

Otro servomotor que presenta un margen de error considerable es el 19I ya que posee errores mayores al 2%, siendo este el caso, los pasos para este servomotor TURNIGY son los siguientes:

- Posición 0: 2200 μ s
- Posición 1: 2050 μ s
- Posición 2: 1900 μ s
- Posición 3: 1750 μ s
- Posición 4: 1600 μ s
- Posición 5: 1450 μ s
- Posición 6: 1300 μ s
- Posición 7: 1200 μ s

Utilizando estos pasos se procede a realizar las mediciones de la distancia recorrida por las cremalleras. Los resultados obtenidos se muestran en el Cuadro 28.

Cuadro 28

Resultados con los nuevos pasos para el servomotor 19I

N° DE SERVOMOTOR	POS. 0/ DIST.	POS. 1/ DIST.	POS. 2/ DIST.	POS. 3/ DIST.	POS. 4/ DIST.	POS. 5/ DIST.	POS. 6/ DIST.	POS. 7/ DIST.
19I	700 / 0	950 / 3.15	1150 / 6.28	1350 / 9.45	1500 / 12.62	1650 / 15.76	1900 / 18.91	2100 / 20.05

Los errores obtenidos con los nuevos pasos se muestran en el Cuadro 29.

Cuadro 29

Errores para los nuevos pasos del servomotor 19I

N° DE SERVOMOTOR	POS. 0	POS. 1	POS. 2	POS. 3	POS. 4	POS. 5	POS. 6	POS. 7
13I	0 %	0 %	0.32 %	0 %	0.16 %	0.06 %	0.05 %	0 %

Como se observa en el Cuadro 29 el margen de error máximo es de 0.32% por lo que los nuevos pasos establecidos permiten recorrer la distancia requerida para que la cremallera pueda pasar de una posición a otra.

5.2 Pruebas de software

5.2.1 Prueba de traducción a Braille de computador

Para la prueba de la traducción a Braille se utiliza un documento de texto donde se muestran los posibles caracteres ASCII que se puedan encontrar un texto, como se observa en la Figura 175.

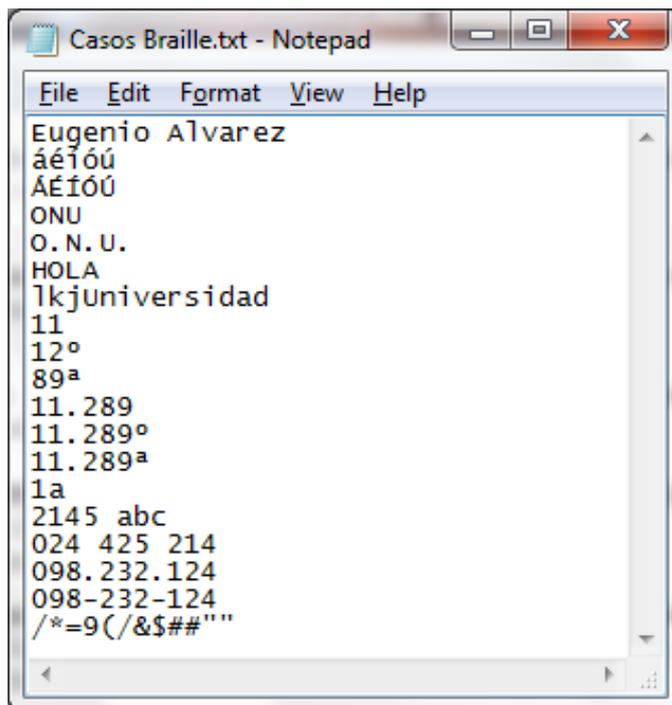


Figura 175. Documento de prueba

Como se observa en la Figura 175, el documento contiene combinaciones de letras, números y caracteres especiales.

Tomando como referencia el anexo 1 del Braille de computador, se realiza el Cuadro 30 donde se muestra el resultado de la conversión de cada línea del documento de prueba mediante el software implementado.

En el caso de letras mayúsculas, la regla Braille establece que se debe colocar un punto antes de las letras mayúsculas. Por otra parte cuando se tiene letras con tilde, el Braille de computador asigna otro caracter a dicha letra tildada, un ejemplo sería la letra “á”, la cual en Braille de computador se representa por el caracter “(“.

Cuadro 30

Resultado de la traducción a Braille de computador mediante el software implementado

TEXTO DEL DOCUMENTO DE PRUEBA	TEXTO TRADUCIDO A BRAILLE DE COMPUTADOR
EUGENIO ALVAREZ	.EUGENIO .ALVARES
ÁÉÍÓÚ	(!/+)
ÁÉÍÓÚ	..(!/+)
ONU	..ONU
O.N.U.	.O'.N'.U'
HOLA	..HOLA
LKJUNIVERSIDAD	.LKJUNIVERSIDAD
11	#AA
12 ^o	#12O
89 ^a	#89A
11.289	#AA'BHI
11.289 ^o	#11'289O
11.289 ^a	#11'289A
1A	#A"A
2145 ABC	#BADE ABC
024 425 214	#JBD DBE BAD
098.232.124	#JIH'BCB'ABD
098-232-124	#JIH-#BCB-#ABD
/*=9(/&\$##""	,197#I<,1&&_S88

De la misma manera cuando se trabaja con números se puede observar que el prefijo en Braille es representado por el carácter “#” seguido de las letras desde la A hasta la J. Como se observa en el Cuadro 30 se utilizan las reglas de Braille y las transformaciones a Braille de computadora, de esta manera se comprueba que la traducción se ha hecho de manera correcta.

5.2.2 Prueba de conversión de Braille a posiciones para servomotores

Para esta prueba se utilizará el documento de texto que se muestra la Figura 176, para comprobar si la trama de posiciones de servomotores se

envía correctamente de acuerdo a los caracteres Braille que se necesita formar. Para esto se ha considerado el Cuadro 11.

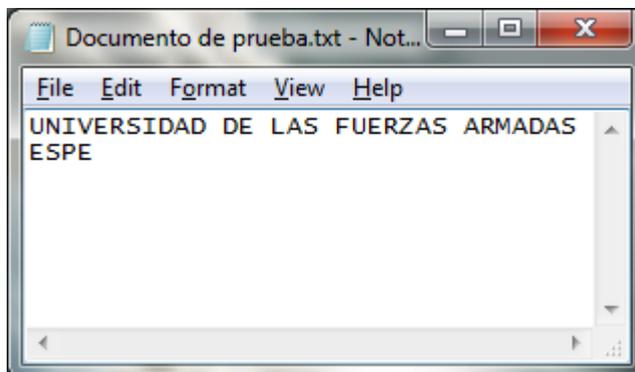


Figura 176. Documento de prueba para posiciones de servomotores

Considerando que el programa únicamente puede desplegar 20 caracteres a la vez, la primera línea se traduce a Braille de la siguiente manera:

“..UNIVERSIDAD ..DE “

Como se observa, se aplican las reglas Braille y al final tiene dos espacios para completar los 20 caracteres sin que se corte ninguna palabra. En la conversión a posiciones de servomotor, se duplica la cantidad de caracteres ya que cada caracter del texto está formado por dos letras que representan posiciones de servomotor. Para la frase mencionada anteriormente la conversión debería ser la que se muestra en el Cuadro 31.

La cadena final de caracteres sería:

ADADDHDECBFHBCFCGBCBBEBABEAAADADBEBCAAAA

Mediante el software implementado, la conversión a posiciones de servomotor es la siguiente:

“ADADDHDECBFHBCFCGBCBBEBABEAAADADBEBCAAAA”

Por lo que se puede verificar que la conversión se realiza correctamente.

Cuadro 31**Conversión a posiciones de servomotor**

CARACTER DE LA LÍNEA DEL TEXTO	POSICIONES DE SERVOMOTOR	
	LETRA 1	LETRA 2
.	A	D
.	A	D
U	D	H
N	D	E
I	C	B
V	F	H
E	B	C
R	F	C
S	G	B
I	C	B
D	B	E
A	B	A
D	B	E
.	A	A
.	A	D
D	B	E
E	B	C
	A	A
	A	A

5.3 Pruebas de comunicación**5.3.1 Prueba de envío de datos desde el computador**

El computador debe enviar la trama con los 40 caracteres que representan las posiciones de servomotores, seguidos de la letra “Y” como caracter de confirmación de trama.

Para verificar que la trama se está enviando correctamente se crea una conexión entre dos puertos seriales virtuales (COM1 y COM2), y se utiliza un terminal serial llamado Hercules 3-2-8. En la Figura 177 se muestra la selección del puerto COM1 como puerto de comunicación

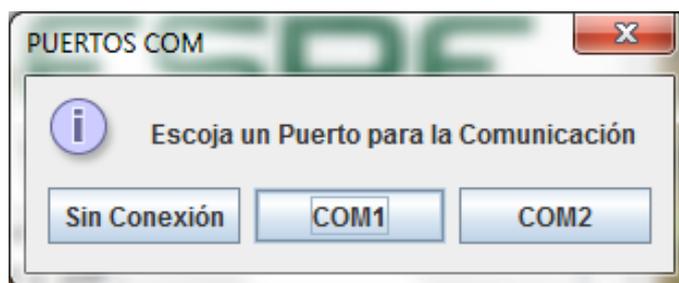


Figura 177. Selección del puerto COM1 como puerto de comunicación

Una vez seleccionado el puerto se ejecuta el terminal serial y se escoge el puerto COM2 para monitorear el envío de datos, como se muestra en la Figura 178.

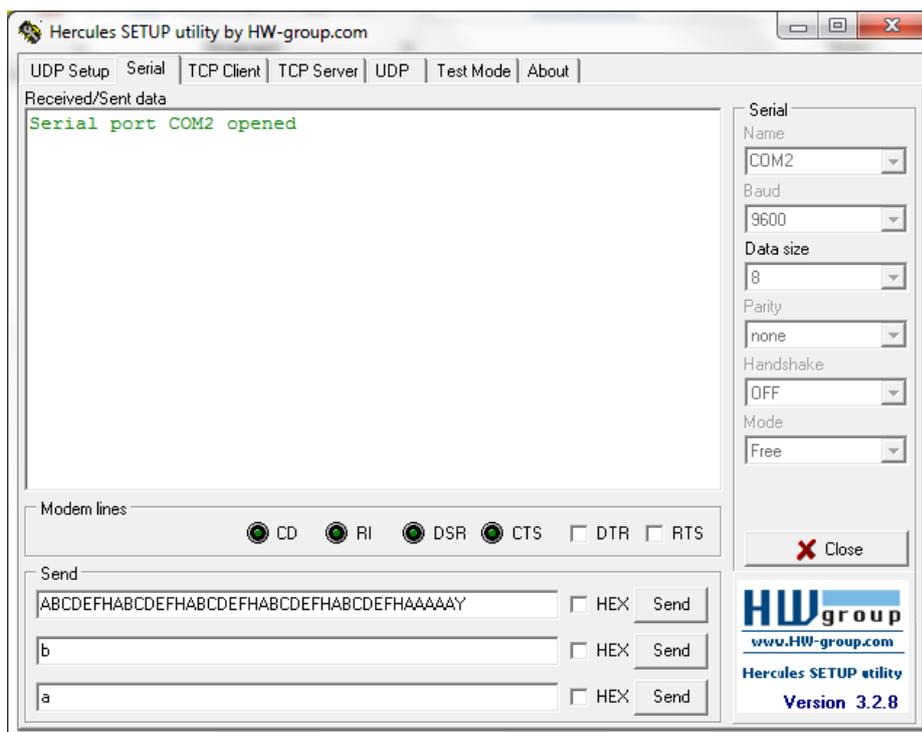


Figura 178. Terminal serial Hercules 3-2-8

Una vez ejecutado el terminal, se procede a abrir el archivo utilizado en la sección anterior, el que se observa en la Figura 176, y se comprueba que la trama sea la misma que se mostró en la prueba de software. En la Figura 179 se muestra el archivo abierto desde el software.



Figura 179. Documento abierto con el software implementado

La trama esperada es la siguiente:

ADADDHDECBFHBCFCGBCBBEBABEAAADADBEBCAAAY

En la Figura 180 se muestra la trama obtenida en el terminal.

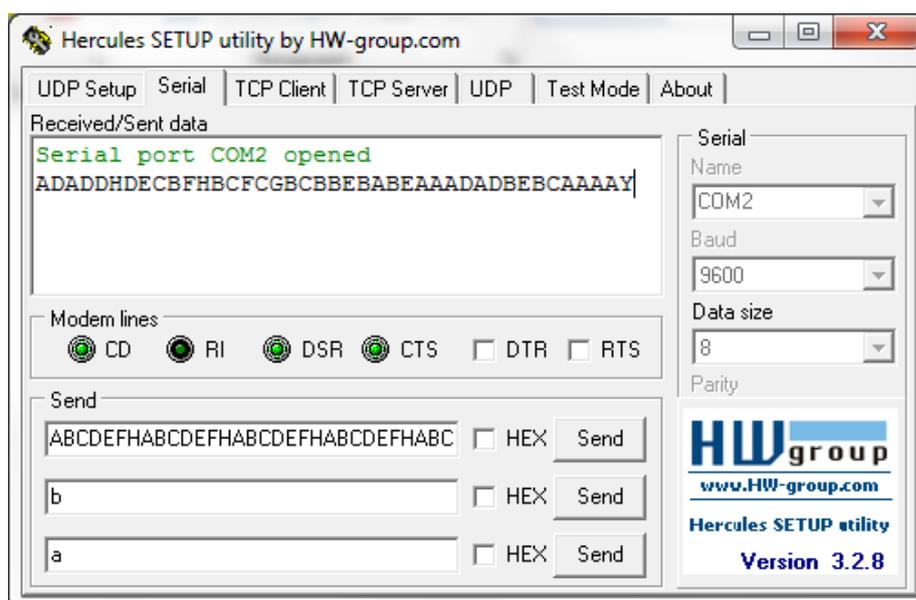


Figura 180. Trama obtenida en el terminal serial

Como se observa en la Figura 180 la trama obtenida es:

“ADADDHDECBFHBCFCGBCBBEBABEAAADADBEBCAAAAY”

Esto quiere decir que el envío de datos desde el computador se realiza correctamente, se envían los 40 caracteres más el caracter final de confirmación de trama.

5.3.2 Pruebas de comunicación del microcontrolador

Para estas pruebas en primer lugar el microcontrolador debe enviar cuatro letras hacia el computador “a”, “b”, “c” y “d”, las cuales representan las acciones de avanzar, retroceder, inicio del documento y fin del documento, respectivamente.

Con la ayuda del terminal serial mencionado anteriormente, y el convertidor USB – TTL se comprueba el envío de estos caracteres desde el microcontrolador a la computadora.

En la Figura 181, Figura 182, Figura 183 y Figura 184 se muestran los caracteres recibidos desde el PIC, al presionar cada uno de los botones.

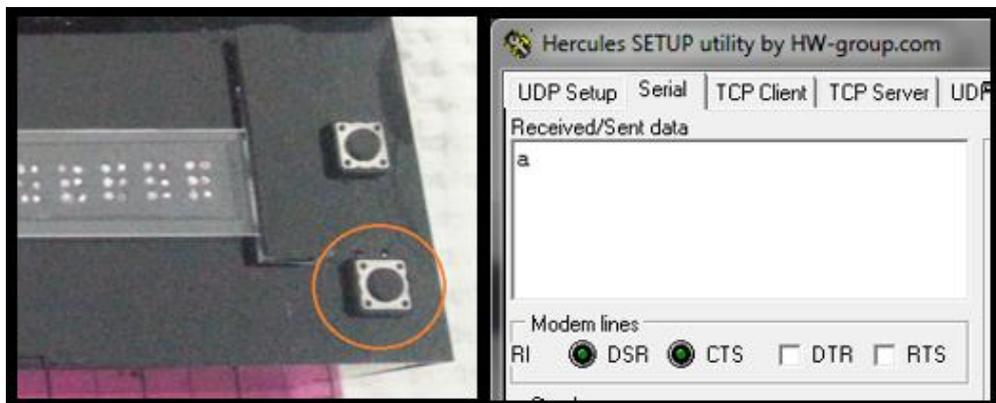


Figura 181. Caracter obtenido después de presionar el botón avanzar

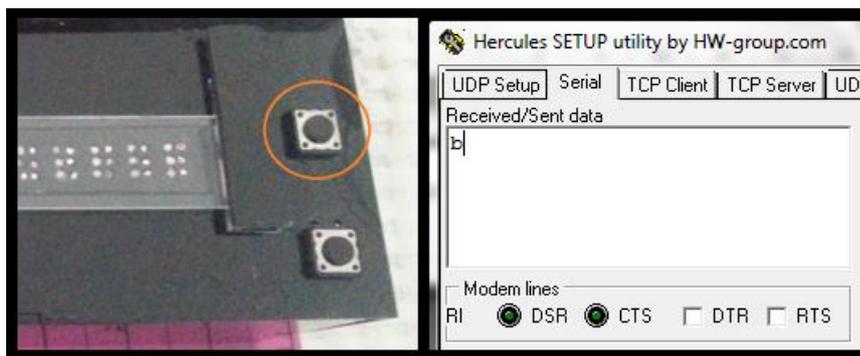


Figura 182. Caracter obtenido al presionar el botón retroceder

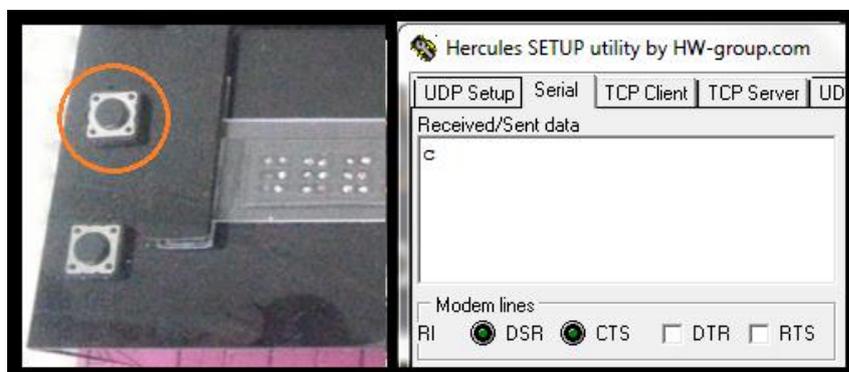


Figura 183. Caracter obtenido después de presionar el botón inicio de documento

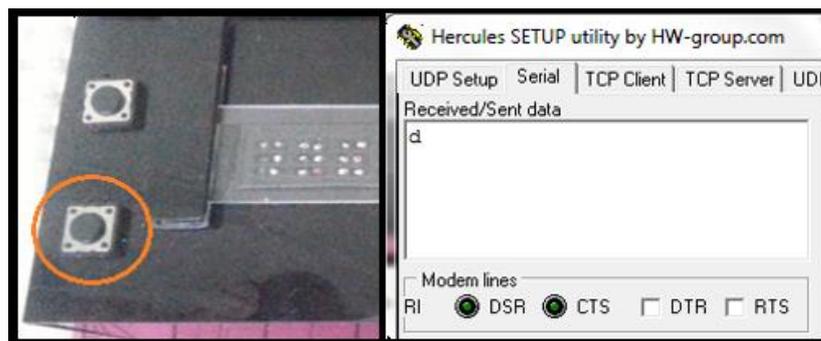


Figura 184. Caracter obtenido después de presionar el botón fin de documento

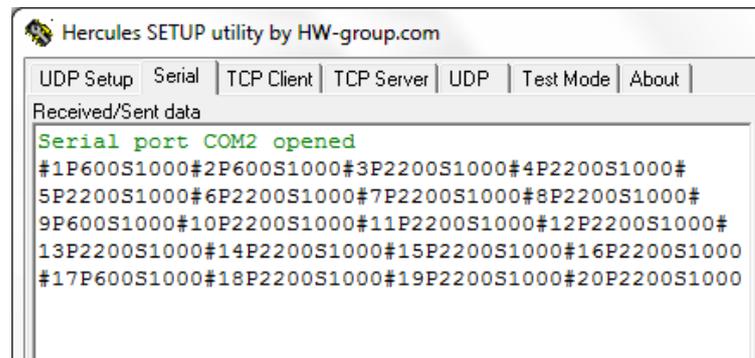
Como se observa en las imágenes anteriores todos los datos se envían correctamente al computador. A continuación se procede a comprobar las

tramas enviadas desde el microcontrolador hacia los controladores de servomotores, utilizando el mismo terminal serial y capturando las tramas mediante el convertidor USB – TTL.

Una vez comprobado la transmisión de datos al computador, se procede a realizar las pruebas de comunicación entre el microcontrolador y los controladores de servomotores. Se realiza un procedimiento similar al anterior, utilizando el convertidor USB – TTL y el terminal serial, se procede a realizar las capturas de tramas enviadas desde el PIC hacia cada controlador. Para esta prueba se utiliza la siguiente trama de posición 0 para todos los servomotores:

“AAAY”

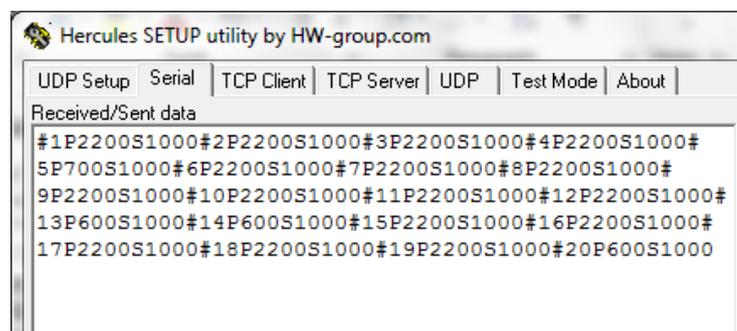
Para el controlador 1 se obtiene la trama que se observa en la Figura 185. Para el controlador 2 se obtiene la trama que muestra en la Figura 186.



```

Hercules SETUP utility by HW-group.com
UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About |
Received/Sent data
Serial port COM2 opened
#1P600S1000#2P600S1000#3P2200S1000#4P2200S1000#
5P2200S1000#6P2200S1000#7P2200S1000#8P2200S1000#
9P600S1000#10P2200S1000#11P2200S1000#12P2200S1000#
13P2200S1000#14P2200S1000#15P2200S1000#16P2200S1000
#17P600S1000#18P2200S1000#19P2200S1000#20P2200S1000
  
```

Figura 185. Trama enviada desde el microcontrolador al controlador 1 de servomotores



```

Hercules SETUP utility by HW-group.com
UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About |
Received/Sent data
#1P2200S1000#2P2200S1000#3P2200S1000#4P2200S1000#
5P700S1000#6P2200S1000#7P2200S1000#8P2200S1000#
9P2200S1000#10P2200S1000#11P2200S1000#12P2200S1000#
13P600S1000#14P600S1000#15P2200S1000#16P2200S1000#
17P2200S1000#18P2200S1000#19P2200S1000#20P600S1000
  
```

Figura 186. Trama enviada desde el microcontrolador al controlador 2 de servomotores

Como se observa en la Figura 185 y Figura 186 las tramas enviadas hacia los controladores tienen el protocolo requerido para ubicar los servomotores, además al final de cada posición se observa los caracteres "S1000" este dato permite establecer la velocidad con la que se mueven los servomotores, independientemente de la posición en la que ubiquen, tardan siempre 250 ms en posicionarse, es decir que en formar una línea de 20 caracteres tardaría 5 s, ya se mueven dos servos a la vez, uno del primer controlador y otro del segundo.

5.4 Prueba de funcionamiento

Para la prueba de funcionamiento de todo el sistema se utiliza el mismo documento de la Figura 176. Al abrir el programa por primera vez debe aparecer un mensaje para escoger el puerto COM, como se muestra en la Figura 187.



Figura 187. Selección COM4

El documento se despliega en el cuadro de texto del programa, y en la parte inferior se muestra el texto en Braille de computador, que está siendo desplegado en el dispositivo, este texto se puede evidenciar en la Figura 188.

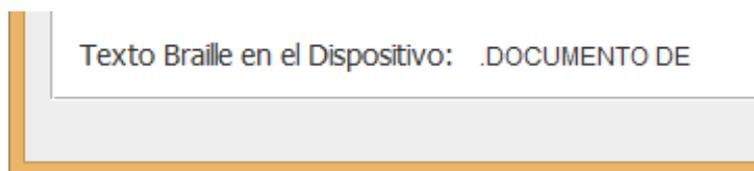


Figura 188. Texto "Documento de" en el programa

El texto de la Figura 188 es “.DOCUMENTO DE” el cual en Braille es representado por los símbolos de la Figura 189.



Figura 189. Texto “Documento de” en Braille

El dispositivo se encarga de desplegar dichos caracteres, como se muestra en la Figura 190 y Figura 191.



Figura 190. Primeros diez caracteres del texto “Documento de”



Figura 191. Últimos diez caracteres del texto “Documento de”

Al presionar el botón de siguiente en el dispositivo, en el programa se puede observar la siguiente línea como se muestra en la Figura 192.

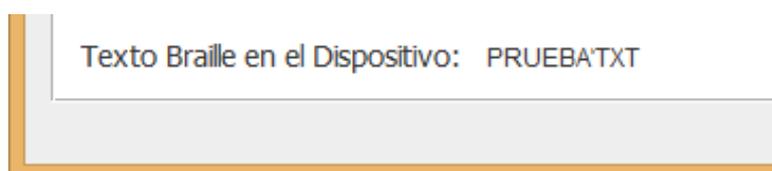


Figura 192. Texto “prueba.txt” en el programa

El texto de la Figura 192 es “PRUEBA.TXT” el cual en Braille es representado por los símbolos de la Figura 193.

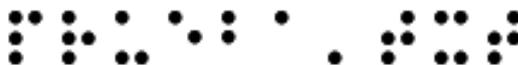


Figura 193. Texto “prueba.txt” en Braille

El dispositivo despliega los caracteres, como se muestra en la Figura 194 y Figura 195.



Figura 194. Primeros diez caracteres del texto “prueba.txt”



Figura 195. Últimos diez caracteres del texto “prueba.txt”

Se continúa con la siguiente línea como se muestra en la Figura 196.

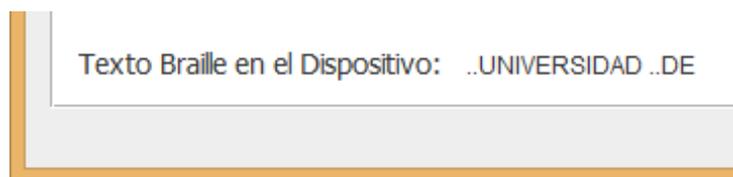


Figura 196. Texto “UNIVERSIDAD DE” en el programa

El texto de la Figura 196 es “..UNIVERSIDAD ..DE” el cual en Braille es representado por los símbolos de la Figura 197.



Figura 197. Texto “UNIVERSIDAD DE” en Braille

El dispositivo despliega los caracteres, como se muestra en la Figura 198 y Figura 199.



Figura 198. Primeros diez caracteres del texto “UNIVERSIDAD DE”



Figura 199. Últimos diez caracteres del texto “UNIVERSIDAD DE”

Se continúa con la siguiente línea como se muestra en la Figura 200.

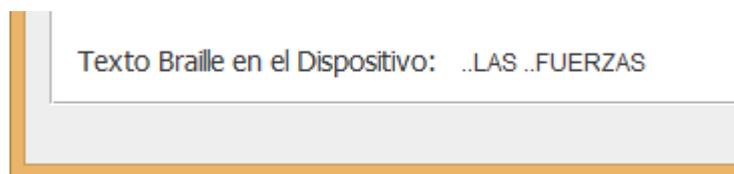


Figura 200. Texto “LAS FUERZAS” en el programa

El texto de la Figura 200 es “..LAS ..FUERZAS” el cual en Braille es representado por los símbolos de la Figura 201.

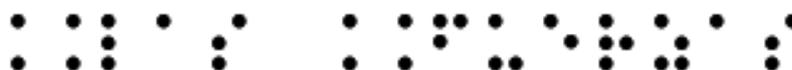


Figura 201. Texto “LAS FUERZAS” en Braille

El dispositivo despliega los caracteres, como se muestra en la Figura 202 y Figura 203.



Figura 202. Primeros diez caracteres del texto “LAS FUERZAS”



Figura 203. Últimos diez caracteres del texto “LAS FUERZAS”

Se continúa con la siguiente línea como se muestra en la Figura 204.

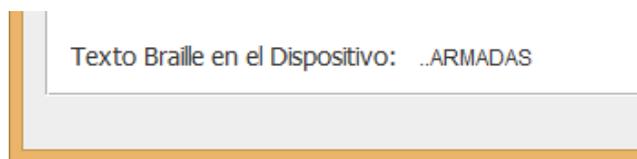


Figura 204. Texto “ARMADAS” en el programa

El texto de la Figura 204 es “..ARMADAS” el cual en Braille es representado por los símbolos de la Figura 205.

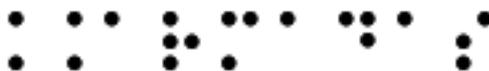


Figura 205. Texto “ARMADAS” en Braille

El dispositivo despliega los caracteres, como se muestra en la Figura 206 y Figura 207.



Figura 206. Primeros diez caracteres del texto “ARMADAS”



Figura 207. Últimos diez caracteres del texto “ARMADAS”

Se continúa con la siguiente línea como se muestra en la Figura 208.

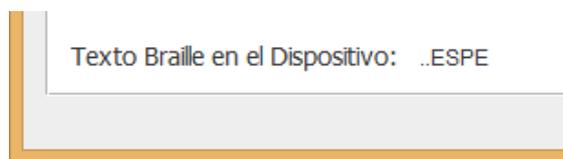


Figura 208. Texto “ESPE” en el programa

El texto de la Figura 208 es “..ESPE” el cual en Braille es representado por los símbolos de la Figura 209.

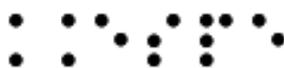


Figura 209. Texto “ESPE” en Braille

El dispositivo despliega los caracteres, como se muestra en la Figura 210 y Figura 211.



Figura 210. Primeros diez caracteres del texto “ESPE”



Figura 211. Últimos diez caracteres del texto “ESPE”

Por último, al culminar el archivo de texto, aparece la oración “Fin del Archivo” que se muestra en Braille de computadora en la Figura 212.

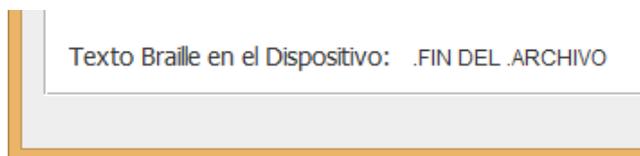


Figura 212. Texto “Fin del Archivo” en el programa

El texto de la Figura 212 es “.FIN DEL .ARCHIVO” el cual en Braille es representado por los símbolos de la Figura 213.



Figura 213. Texto “Fin del Archivo” en Braille

El dispositivo despliega los caracteres, como se muestra en la Figura 214 y Figura 215.



Figura 214. Primeros diez caracteres del texto “Fin del Archivo”



Figura 215. Últimos diez caracteres del texto “Fin del Archivo”

Al cerrar el programa el dispositivo se reinicia como se muestra en la Figura 216.



Figura 216. Dispositivo sin desplegar caracteres

Al comparar las imágenes tomadas del dispositivo con el texto en Braille, se comprueba que los pines representan de manera correcta las palabras Braille enviadas por el programa. También se puede notar en las imágenes que ciertos pines se levantan más que otros, pero la diferencia es despreciable y a pesar de esta diferencia el usuario aún puede sentir con sus dedos que el pin está levantado.

5.5 Manual de Usuario

El manual de usuario es un documento adjunto al programa, que informa al usuario sobre el funcionamiento del sistema. Además muestra los pasos a seguir en el proceso de instalación del programa y las soluciones para posibles errores que se puedan presentar en la ejecución. Este manual se encuentra en el Anexo 2.

5.6 Análisis Económico

El presente proyecto es financiado por la empresa Kradac.Cia.Ltda.

A continuación se presenta los costos implicados en la realización del prototipo.

5.6.1 Detalle de costo

El análisis de costo incluye los elementos necesarios para la implementación del prototipo tanto para la parte electrónica como para la parte mecánica. En el Cuadro 32 se detallan los elementos electrónicos utilizados para el funcionamiento del dispositivo, incluyendo el costo de importación de los actuadores utilizados en el proyecto.

Cuadro 32

Cuadro de costos de componentes electrónicos

DESCRIPCIÓN	Cantidad	VALOR UNITARIO	VALOR TOTAL
Servomotor turnigy	32	3.99	127.68
Servomotor tower pro	8	2.89	23.12
Importación de servomotores	1	65	65
Cables para servomotor	103	0.20	20.6
Zócalo 28 pines	1	0.13	0.13
Bornera 3p	3	0.31	1.94
Bornera 2p	5	0.22	1.12
Tubo termo retráctil	1 [m]	0.40	0.40
Resistencias	5	0.05	0.25
Botones	4	0.20	0.80
Interruptor	1	0.45	0.45
Cable USB	1	1.50	1.50
Puerto USB hembra	2	0.35	0.70
Convertidor USB a TTL serial	1	10	10
Controlador de servomotor	2	55	110
Microcontrolador	1	11.20	11.20
Circuito impreso	1	2.50	2.50
Cables de poder	2	3.50	7
Capacitores	2	0.15	0.30
Cristal oscilador	1	0.40	0.40
Fuente de voltaje	2	38.50	77
Transformador	2	16	32
TOTAL			494.09

En el Cuadro 33 se encuentra el costo detallado de los componentes mecánicos implementados.

Cuadro 33

Cuadro de costos de componentes mecánicos

DESCRIPCIÓN	Cantidad	VALOR UNITARIO	VALOR TOTAL
Láminas de acrílico	0.80 x 1.20 [m]	19	19
Corte laser de acrílico	1	17	17
Caja para fuente de voltaje	1	14.38	14.38
Impresión 3d	1	200	200
TOTAL			250.38

Cuadro 34

Cuadro de costos totales

DESCRIPCIÓN	VALOR
Costo total componentes electrónicos	494.27
Costo total componentes mecánicos	250.38
Transporte	50
Costos imprevistos	20
TOTAL	814.47

Por último en el Cuadro 34 se puede observar el valor total de implementación, tomando en cuenta los valores totales de componentes electrónicos, componentes mecánicos, gastos en transporte y costos imprevistos para el desarrollo del proyecto, resultando en un valor total de US\$814.47.

5.6.2 Análisis de costo y beneficio

La importancia de la realización del proyecto se centra en la ayuda que se puede brindar a personas no videntes, para que tengan acceso a documentos de texto digitales, ya que un lector Braille en el mercado tiene un precio que

varía entre US\$5000 y US\$15000, además que no existe disponibilidad de este tipo de productos en el país.

A pesar de que estos dispositivos son diseñados pensando en que serán adquiridos por institutos o fundaciones enfocadas en ayuda a personas no videntes, su elevado costo representa un obstáculo económico dentro del país, puesto que muchas de las fundaciones son sin fines de lucro.

Como se aprecia en el detalle de costo, el valor del dispositivo es de US\$814.47, comparando este valor con lo que cuesta un producto comercial, la diferencia es muy relevante, no obstante se debe considerar que el dispositivo es el primer prototipo, por lo que el diseño e implementación puede ser modificado si se continúa con el desarrollo del proyecto.

Para las pruebas de funcionalidad del dispositivo se cuenta con la ayuda del Instituto para niños ciegos y sordos “Mariana de Jesús”, con la colaboración de la Lcda. Gladys Segovia.

La Lcda. Segovia es una profesora no vidente del instituto, y lleva ejerciendo la profesión de docencia con niños y adultos con capacidades especiales por 38 años. Gladys maneja el sistema Braille desde los 12 años, edad en la cual perdió la vista y actualmente se encuentra trabajando en la inclusión de niños ciegos a la educación regular.

Para las pruebas se utilizó un documento con el siguiente texto:

prueba 1.pdf

Estos caracteres son un ejemplo del equipo “Braille”

El software divide el texto en las siguientes líneas:

Línea 1

- Texto a leer: prueba.pdf
- Texto en Braille de computadora:

PRUEBA #1'PDF

- Texto en Braille: En la Figura 217 se muestra la traducción.



Figura 217. Línea 1 traducida a Braille

- Resultados:

En la celda 12, el punto 5 de la letra D se siente más bajo que los demás, por lo que el usuario tiene inseguridad de la letra que se está desplegando.

Línea 2

- Texto a leer: Estos caracteres
- Texto en Braille de computadora:

.ESTOS CARACTERES

- Texto en Braille: En la Figura 218 se muestra la traducción.



Figura 218. Línea 2 traducida a Braille

- Resultados:

La letra T presenta problemas puesto que en la celda 4 el punto 5 no se levanta completamente, por lo que se siente más bajo que los demás, y en la celda 11 el punto 1 también tiene un altura inferior a los demás por lo que se confunde entre si se encuentra levantando o no.

Línea 3

- Texto a leer: son un ejemplo del
- Texto en Braille de computadora:

SON UN EJEMPLO DEL

- Texto en Braille: En la Figura 219 se muestra la traducción.



Figura 219. Línea 3 traducida a Braille

- Resultados:

En la celda 4 el punto 1 se encuentra ligeramente levantando por lo que el espacio se puede confundir con la letra A. En la celda 8 el punto 5 se encuentra más bajo que los demás. En la celda 14 ocurre el mismo problema que con la celda 8. La letra L de la celda 18 no se despliega.

Línea 4

- Texto a leer: equipo “Braille”
- Texto en Braille de computadora:

EQUIPO 8.BRAILLE8

- Texto en Braille: En la Figura 220 se muestra la traducción.



Figura 220. Línea 4 traducida a Braille

- Resultados:

En celda 2 el punto 2 no se levanta completamente por lo que se confunde la letra Q (12345) con la letra N (1345). En la celda 4 el punto 4 tampoco se levanta completamente. En la celda 8 el punto 1 se queda ligeramente levantado por lo que se confunde la letra V (1236) con el símbolo de comillas (236).

Los resultados obtenidos se muestran en el Cuadro 35 .

Cuadro 35

Resultado de las pruebas

N° DE LÍNEA	N° DE CELDA	PINES AFECTADOS	LETRA	PROBLEMAS ENCONTRADOS
LÍNEA 1	12	5	D	El pin se siente más bajo que los demás
LÍNEA 2	4	5	T	El pin se siente más bajo que los demás.
	11	1	a	El pin se siente más bajo que los demás.
LÍNEA 3	4	1	Espacio	El pin se queda ligeramente levantado cuando debe estar desactivado
	8	5	E	El pin se siente más bajo que los demás
	14	5	o	El pin se siente más bajo que los demás
	18	1, 2, 3	L	No se levantan los pines
LÍNEA 4	2	2	Q	El pin se siente más bajo que los demás
	4	4	I	El pin se siente más bajo que los demás
	8	1	“	El pin se queda ligeramente levantado cuando debe estar desactivado

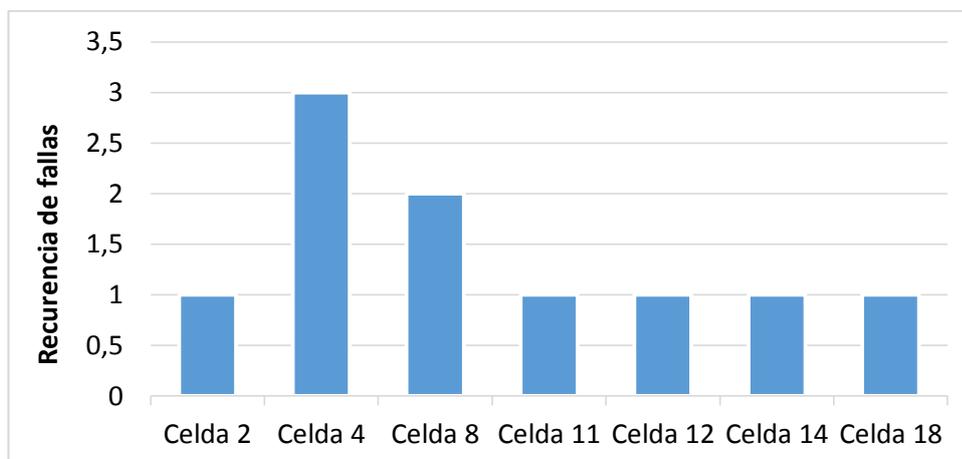


Figura 221. Celdas con falla

En la Figura 221 se muestra un gráfico de barras con las iteraciones de falla que se presentaron en cada celda en una prueba con 4 líneas de texto.

Se puede observar que la celda 4 presenta la mayor cantidad de fallas a lo largo de las pruebas, principalmente formando las letras T, l y el espacio, lo que indica que no regresa completamente a la posición inicial (todos los pines desactivados) y las posiciones de la fila de la derecha no elevan los pines correctamente.

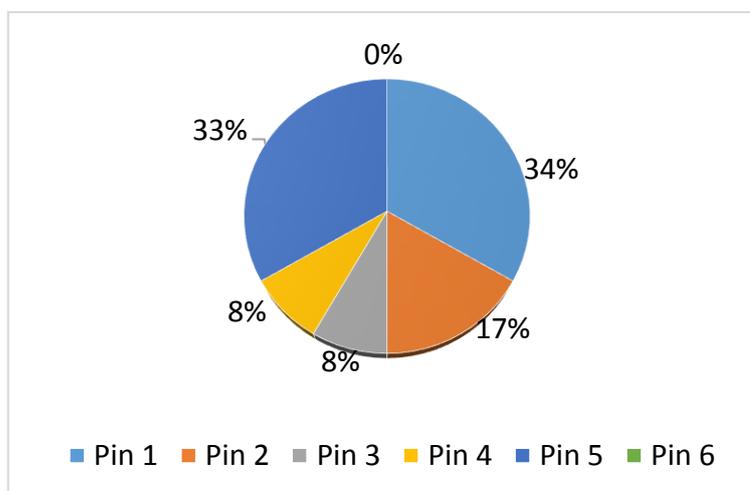


Figura 222. Pines con falla

En la Figura 222 se muestra una gráfica donde se puede observar que los pines que presentan mayores inconvenientes son el pin 1 y el pin 5. El principal inconveniente se debe a la dimensión de los pines, ya que no todos tienen exactamente la misma altura lo que puede presentar confusión al momento de realizar la lectura por parte del usuario. El error más común se puede presentar en letras donde únicamente se diferencian por la activación y desactivación de un solo pin.

Por otra parte la lectura se realiza cómodamente, sin causar fatiga en el lector, además utilizar 20 caracteres resulta satisfactorio para realizar la lectura de documentos. El dispositivo estuvo funcionando alrededor de 40 minutos y no se presentan problemas de sobrecalentamiento o del funcionamiento de los componentes electrónicos.

La opinión final por parte de la Lcda. Segovia acerca del dispositivo fue:

“Es buena la máquina porque no solo serviría para una persona ciega sino también podríamos utilizarla y adaptarla para una persona sordo-ciega, porque la persona sordo-ciega está con sus dos medios de comunicación bastante limitados, entonces esto sería un medio de comunicación con el mundo exterior, porque esta ayuda conectada a la computadora podría servirle de mucha información...”

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- El programa de computadora desarrollado permite extraer el texto de documentos digitales en formato PDF ignorando las imágenes que se puedan presentar, y en formato TXT, tomando en cuenta que este tipo de archivo utilice la codificación ANSI.
- El mecanismo diseñado para desplegar los caracteres Braille, disminuye la cantidad de actuadores necesarios para el posicionamiento de los seis pines en cada celda, ya que permite controlar a través de una sola cremallera, la disposición de tres pines simultáneamente.
- El costo del prototipo final de US\$ 814.47 es bajo en comparación a otros dispositivos Braille, ya que incluso la máquina de Perkins, siendo el equipo más accesible económicamente, tiene un precio alrededor de US\$ 1500, tomando en cuenta que su funcionamiento es exclusivamente mecánico.
- El proceso de lectura en el dispositivo no requiere una capacitación adicional por parte del usuario, ya que en el desarrollo del prototipo se utilizó el sistema Braille de seis puntos aplicado en las instituciones del país.
- Para la traducción de expresiones matemáticas, musicales o científicas se necesita aplicar reglas adicionales a las implementadas en el programa, las cuales permiten diferenciar este tipo de expresiones de un texto normal. Así en matemáticas las variables no se confunden con letras normales, en música y poesía se indica cuando se abre y cierra un verso, y con expresiones científicas se usa las unidades de medida correctamente. Estos casos especiales no están estipulados en el programa que transcribe los documentos suponiendo que están compuestos únicamente por letras y números.

- Los caracteres mostrados como resultado de la traducción a Braille de computadora, únicamente coinciden con las letras del texto, pero ignoran los formatos, símbolos gramaticales o números que se puedan presentar. Al momento de transcribir el texto con las reglas Braille, se elimina todo formato de texto y se utiliza la tabla de Braille de computadora para transcribir cada símbolo, letra y número. Según la tabla, las letras en mayúscula y minúscula son representadas por su misma letra en mayúscula, mientras que los símbolos son reemplazados por símbolos diferentes.
- El sistema es escalable hasta 32 caracteres, puesto que los controladores permiten conectar 12 servomotores en cada uno, además la distribución de servomotores permite agregar seis grupos de cuatro motores, tres en el nivel superior y tres en el inferior, lo que permite generar 12 caracteres adicionales, aumentando únicamente la dimensión horizontal.
- Hay ocasiones donde los pines pueden salir expedidos de su posición, esto se da debido a que existen momentos en los que la interacción entre la punta de la cremallera y los pines es muy brusca.
- Utilizar un controlador dedicado para el manejo de servomotores, permitió una implementación más simple del prototipo puesto que reduce el hardware necesario para esta tarea al reemplazar la necesidad de implementar varios microcontroladores para el control de los motores. Además brinda mayor precisión en el movimiento de los servomotores, ya que se enfoca específicamente en realizar una tarea determinada, y al disponer de un puerto serial para la recepción de comandos la programación del microcontrolador se simplifica.
- El uso de la librería PDFBox permite simplificar la programación de la aplicación JAVA, puesto que cuenta con los métodos necesarios para la extracción del texto de un documento de tipo PDF y al ser de código abierto, esta puede ser usada y distribuida libremente.

- Cada controlador permite conectar hasta 32 servomotores, sin embargo tiene conectados 20 por lo que trabaja al 62.5% de su capacidad máxima, además cada uno está alimentado con una fuente de 6VDC/5A, de esta manera el consumo de corriente se reparte de manera equilibrada y se evita el sobrecalentamiento de estos controladores, lo que permite un uso prolongado del dispositivo.
- La calidad en la impresión 3D de las cremalleras provoca los errores mostrados como resultados de las pruebas puesto que, al tratarse de dimensiones milimétricas cualquier imperfección implica una diferencia en la altura con la que se levantan los pines, ya sea de 0.5 mm, esa diferencia puede confundir al lector, ya que no puede estar seguro si el pin se encuentra activado o desactivado.

6.2 Recomendaciones

- El acople de piñón y cremallera debe ser el mismo para todos los servomotores, es decir que al ensamblar los piñones, los motores deben encontrarse en posición inicial y los piñones con los dientes paralelos unos con otros, para que el desplazamiento de las cremalleras sea exactamente igual.
- Se recomienda evitar la movilización del dispositivo, y en caso de ser necesario realizarlo con extremo cuidado puesto que el prototipo tiene partes móviles que pueden ser desubicadas por el movimiento excesivo al momento de trasladar el equipo.
- Es recomendable que el puerto utilizado para la comunicación sea almacenado de manera que facilite la operación del programa por parte del usuario, de tal forma que la configuración del puerto se realice una sola vez, durante la primera ejecución del programa.
- Se recomienda quitar siempre la tapa protectora de las celdas Braille antes de utilizar el dispositivo, debido a que al alimentar los motores estos tienden a mover las cremalleras hacia adelante, lo

que provoca el levantamiento de pines. La obstrucción del movimiento de los pines puede provocar averías en el prototipo.

- Es recomendable apagar el dispositivo mientras no se esté utilizando de manera que se prolongue la vida útil del prototipo, por otra parte se aconseja apagarlo cada treinta minutos durante un lapso de 10 minutos para evitar el sobrecalentamiento de los servomotores.
- Se recomienda esperar a que se forme la línea completa de caracteres Braille en el dispositivo, para pulsar el botón que permite pasar a la siguiente línea, puesto que de no hacerlo se detendrá el despliegue de los caracteres. Además, una vez que se cierre el programa se debe esperar a que las celdas se hayan reiniciados y que los pines vuelvan a la posición inicial para poder apagar el dispositivo.
- Existen ocasiones donde los pines pueden salir expedidos de su posición por una brusca interacción entre la punta de la cremallera y los pines, esto ocasiona que los pines no regresen a su posición de desactivación, dando la impresión visual de que están levantados, por lo que se recomienda pasar la yema de los dedos antes de leer.
- Es recomendable la altura de los dientes sea ligeramente menor a la utilizada en el proyecto, tanto en las cremalleras como en los piñones, de manera que se obtenga un acople más seguro entre estas dos piezas.
- Se recomienda que todos los servomotores sean de la misma marca y probar que cada uno de los pasos sean iguales en todos, para poder obtener mayor fiabilidad en la transición de posiciones de las cremalleras, además permite simplificar el código del microcontrolador, ya que si todos los pasos son iguales se puede utilizar un mismo proceso para el movimiento de todos los servomotores.

- Es recomendable utilizar servomotores con engranajes metálicos para obtener una mayor duración del producto. El inconveniente con este tipo de servomotores es el precio, debido a que el costo de implementación aumenta en un 75% aproximadamente.
- Se recomienda que al momento de leer un archivo en formato TXT, se encuentre guardado con codificación ANSI, la cual incluye caracteres especiales del idioma español, como vocales con tildes.
- El programa puede ser ejecutado en una arquitectura de computador de 32 o 64 bits, por lo que es recomendable que al momento de la instalación se copie las librerías libSerialPort.dll y libSOSerialPort.so, correspondientes a la arquitectura del computador donde se está instalando para evitar errores de comunicación con el dispositivo.

6.3 Trabajos Futuros

- Una mejora en el dispositivo, teniendo en cuenta que los controladores implementados pueden manejar un máximo de 32 servomotores, es incrementar el número de celdas Braille para que se despliegue un total de 32 caracteres, permitiendo que el usuario pueda leer mayor cantidad de texto a la vez.
- El mecanismo utilizado para el movimiento de las cremalleras, puede ser modificado para utilizar únicamente cuatro servomotores por nivel, y un motor a pasos para el desplazamiento horizontal de los servomotores. De esta manera se podrá mover las 40 cremalleras que despliegan los caracteres Braille del dispositivo.
 - Ampliar las funciones del dispositivo, para que este sea empleado como un periférico de salida del computador capaz de desplegar toda la información que se presenta en la pantalla. De igual manera se puede aumentar botones correspondientes a los seis puntos de la celda Braille, con el objetivo de que el dispositivo también sea usado como un periférico de entrada.

REFERENCIAS

- *Alasdair King*. (2001). Recuperado el 14 de Octubre de 2014, de Text and Braille Computer Translation: <http://www.alasdairking.me.uk/brailletrans/javabrilletteranslation.htm>
- *American Foundation for the Blind*. (2014). Recuperado el 17 de Julio de 2014, de What is Braille?: <http://www.afb.org/info/living-with-vision-loss/braille/what-is-braille/123>
- *American Foundation for the Blind*. (2014). Recuperado el 13 de Octubre de 2014, de Recognition of the Braille Code: <http://www.afb.org/LouisBrailleMuseum/braillegallery.asp?FrameID=190>
- *CEIBAL*. (2014). Recuperado el 20 de Octubre de 2014, de ¿Cómo se escribe el sistema Braille?: http://www.ceibal.edu.uy/UserFiles/P0001/ODEA/ORIGINAL/090531_luisbraille.elp/cmo_se_escribe_el_sistema_braille.html
- *Clasificación Internacional de Enfermedades*. (2006). Recuperado el 13 de Octubre de 2014, de Función visual: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CDUQFjAC&url=http%3A%2F%2Fwww.sssalud.gov.ar%2Fhospitales%2Farchivos%2Fcie_10_revi.pdf&ei=DU48VKvINqnksASsu4C4CA&usg=AFQjCNEyhG-fU2cvQmTaByBoomn5_tfV4Q&sig2=ZYhICYpSdIS-s2V2Csa1GA&bvm=bv
- *CONADIS*. (2013). Recuperado el 13 de Octubre de 2014, de Registro Nacional de Discapacidades: http://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2014/03/conadis_registro_nacional_discapacidades.pdf
- *Custom Computer Services*. (2014). Recuperado el 29 de Octubre de 2014, de PCWHD IDE Compiler for Microchip PIC10/12/16/18/24/dsPIC Devices: http://www.ccsinfo.com/product_info.php?products_id=PCWHD_full

- *Definicion.de.* (2014). Recuperado el 10 de Noviembre de 2014, de Definición de PDF: <http://definicion.de/pdf/>
- *Discapnet.* (2013). Recuperado el 17 de Julio de 2014, de El Alfabeto Braille:
<http://www.discapnet.es/Castellano/areastematicas/Accesibilidad/accesibilidadcomunicacion/lenguajebraile/Documents/Guias/Braille/index.htm>
- *Freedom Scientific.* (2014). Recuperado el 17 de Julio de 2014, de Focus Blue Family of Products:
<http://www.freedomscientific.com/products/fs/Focus-landing-page.asp>
- Frye, J. (2010). Recuperado el 28 de Octubre de 2014, de SSC-32 Manual:
<http://www.lynxmotion.com/images/html/build136.htm#ssc32feat>
- *Galeón – hispavista.* (2008). Recuperado el 27 de Octubre de 2014, de Microcontroladores: <http://microcontroladores-e.galeon.com>
- *Genbeta.* (2014). Recuperado el 29 de Octubre de 2014, de NetBeans: <http://www.genbetadev.com/herramientas/netbeans-1>
- *Guías de la Comisión Braille Española.* (2005). Recuperado el 14 de Octubre de 2014, de Signografía básica: <http://sid.usal.es/idocs/F8/FDO12069/signografiabasica.pdf>
- *Instituto de Tecnologías Educativas.* (2001). Recuperado el 21 de Octubre de 2014, de Educación Inclusiva: http://www.ite.educacion.es/formacion/materiales/129/cd/unidad_5/m5_escritura_sist_braille.htm
- *Instituto de Tecnologías Educativas.* (2012). Recuperado el 24 de Octubre de 2014, de Educación Inclusiva: http://www.ite.educacion.es/formacion/materiales/129/cd/unidad_5/m5_aspectos_tener_cuenta.htm
- *Instituto de Tecnologías Educativas.* (2012). Recuperado el 24 de Octubre de 2014, de Educación Inclusiva: http://www.ite.educacion.es/formacion/materiales/129/cd/unidad_5/m5_algunos_inconvenientes.htm

- *Java*. (2014). Recuperado el 28 de Octubre de 2014, de Conozca más sobre la tecnología Java: <https://www.java.com/es/about/>
- *Metec*. (2014). Recuperado el 17 de Julio de 2014, de Braille Cell P16: <http://web.metec-ag.de/braille%20cell%20p16.html>
- *Microchip*. (2014). Recuperado el 27 de Octubre de 2014, de PIC18F2550: <http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC18F2550>
- *Mikroelektronika*. (2014). Recuperado el 27 de Octubre de 2014, de El mundo de los microcontroladores. Microcontroladores PIC: <http://www.mikroe.com/chapters/view/79/capitulo-1-el-mundo-de-los-microcontroladores/>
- *Organización Mundial de la Salud*. (2014). Recuperado el 13 de Octubre de 2014, de Temas de salud. Discapacidades.: <http://www.who.int/topics/disabilities/es/>
- *Organización Mundial de la Salud*. (2014). Recuperado el 13 de Octubre de 2014, de Ceguera y discapacidad visual: <http://www.who.int/mediacentre/factsheets/fs282/es/>
- *Perkins School for the Blind*. (2013). Recuperado el 15 de Octubre de 2014, de World Braille Usage: <http://www.perkins.org/assets/downloads/worldbrailleusage/world-braille-usage-third-edition.pdf>
- *Plan Nacional Para el Buen Vivir*. (2009-2013). Recuperado el 3 de Agosto de 2014, de Políticas y Lineamientos: <http://plan.senplades.gob.ec/politicas-y-estrategias1;jsessionid=46F87C34C231ED73027DA066971CDF09.nod eaplan>
- *Silicon Laboratories*. (2013). Recuperado el 27 de Octubre de 2014, de CP2102/9: <http://www.silabs.com/support%20documents/technicaldocs/cp2102-9.pdf>

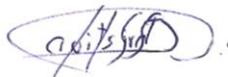
- *Tecno Accesible*. (2011). Recuperado el 24 de Octubre de 2014, de Línea Braille: <http://www.tecnoaccesible.net/node/1063>
- *Universidad de Alicante*. (2008). Recuperado el 24 de Octubre de 2014, de Líneas Braille: <http://accesibilidadweb.dlsi.ua.es/?menu=lin-braille>
- *Universidad Técnica Federico Santa María*. (2003). Recuperado el 28 de Octubre de 2014, de Servomotores: <http://www2.elo.utfsm.cl/~mineducagv/docs/ListaDetalladadeModulos/servos.pdf>

ACTA DE ENTREGA

El proyecto fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas – ESPE, desde:

Sangolquí, 30 DE JUNIO de 2015

ELABORADO POR:



CHRISTIAN SALVADOR GUERRA GAVILANES

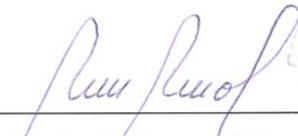
180309993-4



DAVID GUSTAVO NOVILLO VIVANCO

110424789-3

AUTORIDAD



Ing. Luis Orozco MSc.

DIRECTOR DE LA CARRERA DE INGENIERÍA ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL