



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE INGENIERÍA ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA UN SISTEMA DE PID
DE TEMPERATURA CON CELDA PELTIER PARA TERMO PORTÁTIL**

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTORES: MICHAEL ATILIO VELÁSQUEZ MENDOZA
JOSELYN LISSETTE CARVAJAL CABAY

TUTOR: MSC. MÓNICA MARÍA MIRANDA RAMOS

Guayaquil-Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Michael Atilio Velásquez** Mendoza con documento de identificación N°**0930469390** y **Joselyn Lissette Carvajal Cabay** con documento de identificación N°**0956341895**, manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 14 de Septiembre del 2022

Atentamente,



(f) Michael Atilio Velásquez Mendoza

C.I: 0930469390



(f) Joselyn Lissette Carvajal Cabay

C.I: 0956341895

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL
TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA
SALESIANA

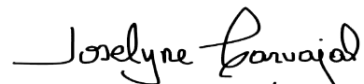
Nosotros, **Michael Atilio Velásquez Mendoza** con documento de identificación No. **0930469390** y **Joselyn Lissette Carvajal Cabay** con documento de identificación No. **0956341895**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del tema : **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA UN SISTEMA DE PID DE TEMPERATURA CON CELDA PELTIER PARA TERMO PORTÁTIL**, el cual ha sido desarrollado para optar por el título de: **INGENIERO ELECTRÓNICO** en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 14 de Septiembre del 2022



(f) Michael Atilio Velásquez Mendoza
C.I: 0930469390




(f) Joselyn Lissette Carvajal Cabay
C.I: 0956341895

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Mónica María Miranda Ramos** con documento de identificación N° **0917271785**, docente de la Universidad Politécnica Salesiana declaro que bajo mi tutoría fue desarrollado el trabajo de titulación : **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA UN SISTEMA DE PID DE TEMPERATURA CON CELDA PELTIER PARA TERMO PORTÁTIL** , realizado por **Michael Atilio Velásquez Mendoza** con documento de identificación N° **0930469390** y por **Joselyn Lissette Carvajal Cabay** con documento de identificación N° **0956341895**, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 14 de Septiembre del 2022



Msc. Mónica María Miranda Ramos
C.I: 0917271785

DEDICATORIA

Dedico este trabajo principalmente a mis padres y a Fernanda Gabriela Vargas Rosero ya que ellos siempre han estado presentes a lo largo de mi vida brindándome el ánimo necesario para seguir mis sueños dándome sus buenos deseos.

Michael Velásquez

DEDICATORIA

Este logro más en mi vida le dedico a mi Dios, mis padres y hermano por ser forjadores de mi meta y por inculcarme siempre la responsabilidad para ser un profesional de bien y nunca darme por vencida ante los obstáculos que se me han presentado.

Joselyn Carvajal

AGRADECIMIENTO

Agradezco a mis padres quienes fueron un pilar fundamental a lo largo de mis estudios. A la Universidad Politécnica Salesiana que siempre tiene sus puertas abiertas para inculcar una buena educación. A todos los profesores que me supieron transmitir sus conocimientos y así emplearlos en la práctica laboral. A mis compañeros que hice en el transcurso de mi carrera, gracias.

Michael Velásquez

AGRADECIMIENTO

Agradezco a mi madre Mirian Cabay y a mi padre Héctor Carvajal por darme el apoyo que me dieron donde gracias a ellos pude culminar con mis estudios Universitarios, también doy gracias a Dios por darme la fortaleza de seguir adelante a pesar de las dificultades que tuve que pasar dentro y fuera de la Universidad. Quiero agradecer también a los Ing. Orlando Barcia, Ing. Susana Lam, Ing. Víctor Choez, Ing. Mónica Miranda y a mi hermano Ing. Héctor Carvajal por ayudarnos en cada momento y darnos ánimos a seguir adelante y no rendirnos en ningún momento donde me enseñó a perseverar y tener metas en la vida para así desarrollarme como profesional y también como persona. También agradezco a mis seres queridos que siempre han estado y agradezco a la familia Jácome han estado siempre en las buenas y en las malas los considero como mi familia.

Joselyn Carvajal

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE PROYECTO	TEMA DE PROYECTO DE TITULACIÓN
2022	VELÁSQUEZ MENDOZA MICHAEL ATILIO CARVAJAL CABAY JOSELYN LISSETTE	MSC.MÓNICA MIRANDA	" DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA UN SISTEMA DE PID DE TEMPERATURA CON CELDA PELTIER PARA TERMO PORTÁTIL "

El proyecto de titulación presenta el estudio de un diseño e implementación de control proporcional, integral y derivado (PID), on-off de temperatura mediante las celdas peltier TEC-12706, la cual varía su temperatura en función de la tensión que pase por ella. Utilizando una plataforma de microcontrolador (Arduino) se establece un confort de manera independiente.

Los sistemas de refrigeración termoeléctricos es una alternativa que da a conocer sus usos y beneficios, su ventaja principal es no utilizar gases refrigerantes, los cuales resultan ser tóxico al ser expuesto a la atmósfera, porque estos debilitan a la capa de ozono que nos ayuda a proteger de la radiación ultravioleta (UV), en espacios reducidos, la celda de peltier TEC-12706 considerada altamente ecológica actúa a la perfección.

Se calcula el modelo matemático de la planta para crear la función de transferencia continua, para hallar los coeficientes faltantes se hace una prueba en lazo abierto con un fin práctico que utiliza el método de Ziegler Nichols. La finalidad de este proyecto es buscar, crear, adaptar y motivar, tanto a docentes como estudiantes a la inclusión a nuevas tecnologías.

Palabras Clave: PID, Celdas Peltier, Automatización, Efecto termoeléctrico, Termopar.

ABSTRACT

YEAR	STUDENTS	PRJ. DIRECTOR	SUBJECT
2022	VELASQUEZ MENDOZA MICHAEL ATILIO CARVAJAL CABAY JOSELYN LISSETTE	MSC.MÓNICA MIRANDA	“DESIGN AND IMPLEMENTATION OF A PROTOTYPE FOR A TEMPERATURE PID SYSTEM WITH PELTIER CELL FOR PORTABLE THERMOS.”

The titling project presents the study of a design and implementation of proportional, integral and derived control (PID), on-off of temperature by means of the peltier cells TEC-12706, which varies its temperature depending on the current that passes through it. Using a microcontroller platform (Arduino) establishes comfort independently.

Thermoelectric cooling systems is an alternative that discloses its uses and benefits, its main advantage is not to use refrigerant gases, which are detailed when exposed to the atmosphere, since these weaken the ozone layer that helps us protect from ultraviolet (UV) radiation, in confined spaces the peltier cell TEC-12706 considered highly ecological acts perfectly.

The mathematical model of the plant is calculated to create the continuous transfer function, to find the missing coefficients an open loop test is made with a practical purpose that uses the Ziegler Nichols method. The purpose of this project is to search, create, adapt and motivate both teachers and students to include new technologies.

Palabras Clave: PID, Peltier Cells, Automation, Thermoelectric Effect, Thermocouple.

ABREVIATURAS

PID: Proporcional Integral Derivado.

RF: Frigorífico.

ON: Encendido.

OFF: Apagado.

DB: Diagrama de bloque.

FC: Función de control.

SC: Semiconductor.

TECNOL: Tecnología.

UV: Radiación ultravioleta.

SAO: Capa de ozono.

DC: Corriente continua.

CA: Corriente alterna.

PWM: Modulación de ancho de pulso.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA	¡Error! Marcador no definido.
DERECHOS DE AUTOR	¡Error! Marcador no definido.
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN	III
DEDICATORIA	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
AGRADECIMIENTO	VII
RESUMEN.....	VIII
ABSTRACT.....	IX
ÍNDICE GENERAL.....	XI
ÍNDICE DE FIGURAS	XIV
ÍNDICE DE TABLAS	XXIII
ÍNDICE DE ANEXOS.....	XXV
INTRODUCCIÓN	1
1. EL PROBLEMA	2
1.1. Descripción del problema.....	2
1.2. Antecedentes	3
1.3. Importancia y alcances	3
1.4. Delimitación del problema	4
1.4.1 Temporal.....	4
1.4.2 Geográfica	4
1.4.3 Académica	4
1.5. Objetivos.....	4
1.5.1 Objetivo General	4
1.5.2 Objetivo Específicos.....	5
2. MARCO TEÓRICO REFERENCIAL.....	5
2.1. Controlador PID	5
2.2. Lazo Abierto.....	7
2.3. Lazo Cerrado O Retroalimentado.....	7
2.4. Control ON/OFF	8

2.5.	Ley De Enfriamiento De Newton	8
2.6.	Ley De Stefan-Boltzmann	9
2.7.	Método De Sintonización De Ziegler Y Nichols	10
2.7.1	Tipo De Funcionamiento Del Lazo De Control.....	10
2.7.2	Método De Lazo Abierto.....	11
2.7.3	Método De Lazo Cerrado	12
2.7.4	Criterio De Desempeño	12
2.8.	Efectos Termoeléctricos.....	12
2.8.1.	Efecto Peltier.....	12
2.8.2	Celda Peltier.....	13
2.8.2.1	Aplicaciones	14
2.8.2.2	Propiedades	15
2.9.	Arduino Nano	16
2.9.1	Características Técnicas Arduino Nano.....	17
2.9.2	Aplicaciones Y Propiedades.....	19
2.10.	Control Driver VNH2SP30	19
2.10.1	Características Técnicas De Motor Monster VNH2SP30.....	20
2.11.	Pantalla De Cristal Líquido 16X2	21
2.11.1	Características Técnicas Lcd 16X2	22
2.12.	Módulo LCM 1602 I2C.....	22
2.13.	Protocolo De Comunicación I2C.....	23
2.14.	Sensor De Temperatura DS18B20	23
2.14.1	Aplicaciones.....	24
2.15.	Módulo Bluetooth HC-06	25
2.16.	Power Suppli 12VDC-15A	26
2.17.	Calentador Eléctrico	26
2.18.	Matlab	27
2.19.	Android.....	28
2.20.	Mit App Inventor	28
3.	Diseño	29
3.1.	Funcionalidad	29
3.2.	Funcionalidad manual o analógica.....	30
3.3.	Funcionalidad Por Teléfono	31
3.4.	Diseño Envase Acero	32
3.5.	Diseño 3d Termo Portátil	33
3.6.	Diseño De Tarjeta PCB	34

3.7.	Diseño Aplicación App Inventor	35
3.8.	Diseño Del Algoritmo Android.....	36
3.8.1.	Comunicación De La Aplicación Apk Android Con Arduino.....	36
4.	IMPLEMENTACIÓN	38
4.1	primera parte de ensamblaje termo portátil 3D	38
4.2	Segunda Parte De Ensamble Termo Portátil	49
5.	ANÁLISIS DE RESULTADOS	56
5.1	Análisis Del Proyecto.....	56
5.2	Análisis Formula De La Segunda Ley De La Termodinámica Y Boltzmann. 61	
5.2.1	Cálculos Linealización Por Serie De Taylor	63
5.2.2	Linealizando El Sistema Por Series De Taylor:	64
5.3	Final módulo general.	66
5.4	Análisis Método De Ziegler-Nichols	67
5.5	Análisis De La Planta.....	72
5.6	Matlab	76
5.7	Configurar Documento De Excel	79
6.	CONCLUSIONES	87
7.	RECOMENDACIONES	88
8.	PROYECTOS DE INVESTIGACIÓN VINCULADOS	89
9.	REFERENCIAS BIBLIOGRÁFICAS	90
10.	ANEXOS	95

ÍNDICE DE FIGURAS

Figura 1. Lazo abierto.	7
Figura.2 Lazo cerrado	8
Figura 3. Ley de enfriamiento de Newton.....	9
Figura 4. Esquema efecto peltier.....	13
Figura 5. Celda peltier.....	14
Figura 6. Efecto producido por corriente en una celda peltier.....	15
Figura 7. Dimensiones físicas de la celda peltier TEC1.12706.....	16
Figura 8. Tarjeta Arduino nano.....	17
Figura 9. Controlador driver vnh2sP30.....	20
Figura 10. LCD16x2.....	21
Figura 11. Módulo LCM 1602 I2C	23
Figura 12.Sensor de temperatura DS18B20.....	24
Figura 13. Módulo Bluetooth HC-06.....	25
Figura 14. Power Suppli 12vDC-15A.....	26
Figura 15. Calentador Eléctrico.....	26
Figura 16. Software Matlab.....	27
Figura 17. Android.....	28
Figura 18. Logo APPInventor.....	28
Figura 19. Diagrama en bloques del proceso.....	29
Figura 20. Diseño de envase de acero.....	32
Figura 21. Diseño físico envase de acero.....	32
Figura 22. Diseño 3D parte frontal contorno del termo.....	33
Figura 23. Diseño 3D parte trasera contorno del termo.....	33
Figura 24. Esquemático Proteus.....	34

Figura 25. Diseño PCB 3D.	34
Figura 26. Pantalla principal APK.	35
Figura 27. Pantalla de comunicación vía bluetooth.	35
Figura 28. Botón Inicio Programación Screen 1.	36
Figura 29. Botón de bluetooth selección de lista Screen 2.	36
Figura 30. Botón de bluetooth selección de lista Screen 2.	37
Figura 31. Botón enfriar Screen 2.	37
Figura 32. Botón apagar Screen 2.	37
Figura 33. Ensamble 1.	38
Figura 34. Ensamble 2.	38
Figura 35. Ensamble 3.	39
Figura 36. Ensamble 4.	39
Figura 37. Ensamble 5.	40
Figura 38. Ensamble 6.	40
Figura 39. Ensamble 7.	41
Figura 40. Ensamble 8.	41
Figura 41. Ensamble 9.	42
Figura 42. Ensamble 10.	42
Figura 43. Ensamble 11.	43
Figura 44. Ensamble 12.	43
Figura 45. Ensamble 13.	44
Figura 46. Ensamble 14.	44
Figura 47. Ensamble 15.	45
Figura 48. Ensamble 16.	45
Figura 49. Ensamble 17.	46

Figura 50. Ensamble 18.	46
Figura 51. Ensamble 19.	47
Figura 52. Ensamble 20.	47
Figura 53. Ensamble 21.	48
Figura 54. Ensamble 22.	48
Figura 55. Ensamble 23.	49
Figura 56. Ensamble 24.	49
Figura 57. Ensamble 25.	50
Figura 58. Ensamble 26.	50
Figura 59. Ensamble 27.	51
Figura 60. Ensamble 28.	51
Figura 61. Ensamble 29.	52
Figura 62. Ensamble 30.	52
Figura 63. Ensamble 31.	53
Figura 64. Ensamble 32.	53
Figura 65. Ensamble 33.	54
Figura 66. Ensamble 34.	54
Figura 67. Ensamble 35.	55
Figura 68. Ensamble 36.	55
Figura 69. Planta.....	56
Figura 70. PID.....	57
Figura 71. Gráfica Sigmoidal.....	58
Figura 72. Lazo cerrado.	59
Figura 73.PWM	59
Figura 74. PWM ciclo de Trabajo.	61

Figura 75. Análisis Planta lazo abierto.	67
Figura 76. La transformada inversa Z.....	70
Figura 77. Análisis tiempo de variación 1.....	72
Figura 78. Análisis tiempo de variación 1.....	73
Figura 79. Análisis tiempo de variación 2.....	74
Figura 80. Análisis tiempo de variación 3.....	74
Figura 81. Matlab Gráfica ft en tiempo continuo.....	76
Figura 82. Matlab Gráfica en tiempo discreto.....	77
Figura 83. Simulink Gráfica ft en tiempo continuo.....	78
Figura 84. Simulink Gráfica ft en tiempo continuo.....	78
Figura 85. Configuración Excel parte 1.....	79
Figura 86. Configuración Excel parte 2.....	79
Figura 87. Configuración Excel parte 3.....	80
Figura 88. Configuración Excel parte 4.....	80
Figura 89. Configuración Excel parte 5.....	81
Figura 90. Configuración Excel parte 6.....	81
Figura 91. Configuración Excel parte 7.....	82
Figura 92. Configuración Excel parte 8.....	82
Figura 93. Configuración Excel parte 9.....	83
Figura 94. Configuración Excel parte 10.....	83
Figura 95. Configuración Excel parte 11.....	84
Figura 96. Configuración Excel parte 12.....	84
Figura 97. Configuración Excel parte 13.....	85
Figura 98. Configuración Excel parte 14.....	85
Figura 99. Configuración Excel parte 15.....	86

Figura 100. Módulo Arduino nano.	97
Figura 101. Módulo BLUETOOTH HC-06.	98
Figura 102. Módulo MONSTER SHIELD.	99
Figura 103. Módulo LCD 16x2.....	100
Figura 104. Celda Peltier.....	101
Figura 105. Gráfica de Comportamiento Celda Peltier.	101
Figura 106. Sensor DS18B20.....	102
Figura 107. Power Suppli 12vDC-15A.....	103
Figura 108. Diodo LED.....	104
Figura 109. Dimensiones Diodo LED	104
Figura 110. Resistencia.....	105
Figura 111. Bornera.	106
Figura 112. Regulador de voltaje 7805.....	107
Figura 113. Potenciómetro.	108
Figura 114. Capacitor electrolítico.....	109
Figura 115. Capacitor Cerámico.....	110
Figura 116. Conector de alimentación 12v.	111
Figura 117. Cooler 12 V.	112
Figura 118. Switch de Bloqueo.....	113
Figura 119. Switch.	114
Figura 120. Diseño de termo portátil 1 en software inventor.....	115
Figura 121. Materiales termo portátil 1.....	116
Figura 122. Conexiones termo portátil 1.....	117
Figura 123. Herramienta ident Software Matlab.	117
Figura 124. Prueba comunicación serial software LabVIEW.	118

Figura 125. Datos visual LCD 16x2.....	119
Figura 126. Esquemático proteus termo portátil 1.....	119
Figura 127. Prototipo 2 en software inventor.....	120
Figura 128. Bomba Brushless Water Pump.....	120
Figura 129. Conexiones Prototipo 2.....	121
Figura 130. Envase de prueba.....	121
Figura 131. Prototipo 3 en software inventor.....	122
Figura 132. Caja de madera Prototipo 3.....	122
Figura 133. Diseño envase metálico prototipo 4.....	123
Figura 134. Reservorios de aluminio.....	123
Figura 135. Sistema retroalimentado.....	124
Figura 136. Prueba prototipo 4.....	124
Figura 137. Conexiones prototipo 4.....	125
Figura 138. Prototipo 4 en software inventor.....	125
Figura 139. Envase de PLA.....	126
Figura 140. Envase final AutoCad.....	126
Figura 141. Diseño PBC.....	127
Figura 142. Impresión PLA software ultimaker cura.....	127
Figura 143. Impresión PLA software ultimaker cura.....	128
Figura 144. Foto de Programación Arduino.....	128
Figura 145. Programación APK en APP Inventor.....	129
Figura 146. Segmento Interfaz De Usuario.....	129
Figura 147. Paleta Disposición.....	130
Figura 148. Paleta Medios.....	130
Figura 149. Paleta Conectividad.....	131

Figura 150. Editor De Bloques.	132
Figura 151. Si Entonces.	132
Figura 152. Abrir Otra Pantalla.	132
Figura 153. Cadena De Texto.	132
Figura 154. Código Arduino Parte 1.	133
Figura 155. Código Arduino Parte 2.	134
Figura 156. Código Arduino Parte3.	135
Figura 157. Código Arduino Parte 4.	136
Figura 158. Código Arduino Parte 5.	137
Figura 159. Código Arduino Parte 6.	138
Figura 160. Código Arduino Parte 7.	139
Figura 161. Código Arduino Parte 8.	140
Figura 162. Código Arduino Parte 9.	141
Figura 163. Prueba 1 Excel lazo abierto.	142
Figura 164. Prueba 1 Gráfica Lazo Abierto.	142
Figura 165. Prueba 1 Planta Lazo Abierto.	143
Figura 166. Prueba 1 Código Arduino Parte1.	143
Figura 167. Prueba 1 Código Arduino Parte 2.	144
Figura 168. Prueba 1 Código Arduino Parte3.	145
Figura 169. Prueba 1 Código Arduino Parte4.	146
Figura 170. Prueba 1 Código Arduino Parte5.	147
Figura 171. Prueba 1 Código Arduino Parte6.	148
Figura 172. Prueba 1 Código Arduino Parte7.	149
Figura 173. Prueba 1 Código Arduino Parte8.	150
Figura 174. Prueba 2 Excel KP/2.	151

Figura 175. Prueba 2 Gráfica en KP/2.....	151
Figura 176. Prueba 2 Planta KP/2.....	152
Figura 177. Prueba 2 Código Arduino Parte1.....	152
Figura 178. Prueba 2 Código Arduino Parte2.....	153
Figura 179. Prueba 2 Código Arduino Parte3.....	154
Figura 180. Prueba 2 Código Arduino Parte4.....	155
Figura 181. Prueba 2 Código Arduino Parte5.....	156
Figura 182. Prueba 2 Código Arduino Parte6.....	157
Figura 183. Prueba 2 Código Arduino Parte7.....	158
Figura 184. Prueba 2 Código Arduino Parte8.....	159
Figura 185. Prueba 3 Excel theta igual A retardo.	160
Figura 186. Prueba 3 Gráfica theta igual A retardo.	160
Figura 187. Prueba 3 Planta theta igual A retardo.....	161
Figura 188. Prueba 3 Código Arduino Parte1.....	161
Figura 189. Prueba 3 Código Arduino Parte2.....	162
Figura 190. Prueba 3 Código Arduino Parte3.....	163
Figura 191. Prueba 3 Código Arduino Parte4.....	164
Figura 192. Prueba 3 Código Arduino Parte5.....	165
Figura 193. Prueba 3 Código Arduino Parte6.....	166
Figura 194. Prueba 3 Código Arduino Parte7.....	167
Figura 195. Prueba 3 Código Arduino Parte8.....	168
Figura 196. Prueba 4 Excel ts igual a 2.....	169
Figura 197. Prueba 4 Gráfica ts igual a 2.....	169
Figura 198. Prueba 4 Planta ts igual a 2.....	170
Figura 199. Prueba 4 Código Arduino Parte1.....	170

Figura 200. Prueba 4 Código Arduino Parte2.....	171
Figura 201. Prueba 4 Código Arduino Parte3.....	172
Figura 202. Prueba 4 Código Arduino Parte4.....	173
Figura 203. Prueba 4 Código Arduino Parte5.....	174
Figura 204. Prueba 4 Código Arduino Parte6.....	175
Figura 205. Prueba 4 Código Arduino Parte7.....	176
Figura 206. Prueba 4 Código Arduino Parte8.....	177
Figura 207. Prueba 5 Excel PID.	178
Figura 208. Prueba 5 Gráfica PID.	178
Figura 209. Prueba 5 Planta PID.....	179
Figura 210. Prueba 5 Código Arduino Parte1.....	179
Figura 211. Prueba 5 Código Arduino Parte2.....	180
Figura 212. Prueba 5 Código Arduino Parte3.....	181
Figura 213. Prueba 5 Código Arduino Parte4.....	182
Figura 214. Prueba 5 Código Arduino Parte5.....	183
Figura 215. Prueba 5 Código Arduino Parte6.....	184
Figura 216. Prueba 5 Código Arduino Parte7.....	185
Figura 217. Prueba 5 Código Arduino Parte8.....	186

ÍNDICE DE TABLAS

TABLA 1. Especificaciones técnicas celdas peltier.....	15
TABLA 2. Características técnicas Arduino Nano.....	17
TABLA 3. Características técnicas LCD 16x2.....	22
TABLA 4. PWM cálculo 12v.....	60
TABLA 5. Voltaje requerido 12v.	60
TABLA 6. Sintonía por el método de Ziegler y Nicols.	71
TABLA 7. Cronograma de proyecto de tesis.....	95
TABLA 8. PRESUPUESTO DEL PROYECTO.	96
TABLA 9. Datasheet Arduino nano.....	97
TABLA 10. Configuración Predeterminada BLUETOOTH HC-06.	98
TABLA 11. Datasheet Módulo MONSTER SHIELD.....	99
TABLA 12. Datasheet Módulo LCD	100
TABLA 13. Datasheet Celda Peltier.....	101
TABLA 14. Datasheet Sensor DS18B20.	102
TABLA 15. Datasheet Power Suppli 12vDC-15A.....	103
TABLA 16. Datasheet Diodo LED.....	104
TABLA 17. Datasheet Resistencia.	105
TABLA 18. Datasheet Bornera.	106
TABLA 19. Datasheet Regulador de voltaje 7805.....	107
TABLA 20. Datasheet Potenciómetro.	108
TABLA 21. Datasheet Capacitor electrolítico.....	109
TABLA 22. Datasheet Capacitor Cerámico.....	110
TABLA 23. Datasheet Conector de Voltaje CON-00 2 – M.....	111

TABLA 24. Datasheet Cooler 12 V.	112
TABLA25. Datasheet Switch de Bloqueo.	113
TABLA26. Datasheet Switch.	114

ÍNDICE DE ANEXOS

ANEXO A. CRONOGRAMA DE DURACIÓN DEL PROYECTO.	95
ANEXO B. PRESUPUESTO DEL PROYECTO	96
ANEXO C. DATASHEET MÓDULOS.....	97
ANEXO D. PROTOTIPOS	115
ANEXO E. DISEÑO FINAL.....	126
ANEXO F. CÓDIGO ARDUINO	133
ANEXO G. PRUEBAS.....	142

INTRODUCCIÓN

En el presente proyecto se estudia una propuesta experimental y busca nuevas alternativas de energía termoeléctrico para enfriar un termo portátil, disponiendo el uso de un sistema de enfriamiento de control PID de lazo cerrado ON-OFF con celdas peltier TEC-12706 que es un elemento de enfriamiento termoeléctrico.

Las celdas peltier TEC-12706 casi no se encuentran en el país por ende no se han realizado muchas investigaciones y proyectos de tecnologías con respecto a ellas.

Se conoce que el PID es un controlador interno que se va a programar en Arduino nano donde tiene programado matemáticas para agilizar el proceso para enviar el error a 0 de una mejor forma con ayuda del modelo matemático encontrado mediante cálculos.

Para analizar de mejor manera la planta que realiza dos funciones una analógica donde el usuario establece la temperatura en el rango de 23°C a 29°C de manera manual y la segunda donde el usuario utiliza una aplicación creada para dispositivos Android donde se establece la temperatura máxima por comunicación vía bluetooth.

Además, se aplican leyes como la ley de enfriamiento de Newton, Stefan Boltzmann y la linealización por serie de Taylor que nos ayuda a analizar los coeficientes necesarios para sacar de una manera segura la función de transferencia continua.

Adaptar la tecnología en varios procesos es necesario en la actualidad, para controlar y mejorar sistemas dado que el ser humano ha desarrollado mecanismos para controlar las variables físicas.

1. EL PROBLEMA

1.1. Descripción del problema

El diseño de sistemas PID en la industria se ha incrementado de manera creciente estos sistemas son muy útiles para procesos de automatización, así mismo existen dispositivos como las tarjetas controladoras Arduino que ajustan el sistema a la temperatura mediante un sensor de aplicación requerida agregando control PID de lazo cerrado ON-OFF.

Como futuros ingenieros podemos usar nuestro conocimiento para utilizar recursos y diseñar un nuevo método o sistema para obtener un proceso diferente al tradicional de estabilización, por lo antes expuesto el propósito de este proyecto es profundizar mediante el estudio del control PID de lazo cerrado ON-OFF utilizando el uso del modelo matemático mediante la construcción del método de control y el uso de software para promover la investigación.

El ser humano desde la antigüedad ha desarrollado mecanismos para controlar las variables físicas que nos rodean y hacer que alcancen valores ideales, bajo esta premisa las personas han realizado métodos como el control PID para determinados procesos a controlar como la temperatura, velocidad, caudal, presión, etc.

La ventaja de estos sistemas para mantener una temperatura constante se debe a la interacción de la celda peltier TEC-12706 con el equipo electrónico y el sistema de control que incluyen el uso de nuevas tecnologías.

1.2. Antecedentes

El ser humano desde la antigüedad ha desarrollado mecanismos para controlar las variables físicas que nos rodean y hacer que alcancen valores ideales, bajo esta premisa las personas han realizado métodos como el control PID para determinados procesos a controlar como la temperatura, velocidad, caudal, presión, etc.

Los sistemas de refrigeración incluyen el uso de nuevas técnicas para su funcionamiento tecnológico, se pondrán en práctica los conocimientos adquiridos durante todo el proceso de estudio de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana de la sede Guayaquil en materias de teoría de control, microcontroladores, electrónica analógica, electrónica digital, electiva, energía y medio ambiente.

Para tener una mejor comprensión y dominio del tema de controladores PID de lazo cerrado ON-OFF , como también el uso y las aplicaciones de la celda Peltier TEC-12706 en el campo de la refrigeración es necesario que los alumnos puedan resolver los problemas de control automático en la actualidad, como ingenieros buscamos la innovación y creación de nuevas tecnologías haciendo que este proceso sea estable y rápido para garantizar el uso eficiente de energía.

1.3. Importancia y alcances

El presente tema tiene como finalidad fortalecer el diseño y el control PID de lazo cerrado ON-OFF para mejorar un proceso y rendimiento óptimo en cuanto a la automatización, para poder adaptar tecnologías alternativas en el campo de la refrigeración como lo es la celda peltier y las aplicaciones tomando en cuenta el aprendizaje de procesos que se realizan dentro de una industria para desarrollar nuevos controles PID dependiendo la necesidad que requiera la planta.

Además, se busca promover y motivar a los alumnos de la Universidad Politécnica Salesiana a la innovación para que puedan resolver problemas mejorando una planta con controladores PID de manera automática y la obtención de datos para corregir posibles daños en un futuro dentro de una empresa aplicando el conocimiento adquirido en clases, como también a los docentes y profesionales para dar soluciones eficaces y mejoras de las mismas.

1.4. Delimitación del problema

1.4.1 Temporal

El objeto de la investigación y desarrollo de este proyecto se realizará en la ciudad de Guayaquil, Ecuador en el período de 2022 - 2022 y tendrá aproximadamente una duración de 6 meses.

1.4.2 Geográfica

El proyecto será un termo y podrá ser transportado con total facilidad de una estancia a otra.

1.4.3 Académica

Este proyecto se realizará cumpliendo con las normas impuestas por la Universidad Politécnica Salesiana basado en su grado investigativo y modelo de presentación para proyectos de titulación, también se pondrán en práctica los conocimientos adquiridos durante todo el proceso de estudio en materias como: Electiva II, Sensores y Transductores, Electrónica Digital, Sensores, Electrónica Analógica, Sistemas Micro procesados I y II y Energía y Medio Ambiente.

1.5. Objetivos

1.5.1 Objetivo General

Diseñar e implementar un prototipo para un sistema de PID de temperatura con celda peltier utilizando Arduino para termo portátil.

1.5.2 Objetivo Específicos

- Diseñar el control PID en lazo cerrado ON-OFF para la preservación Temperatura de refrigeración.
- Obtener el modelo matemático del sistema PID.
- Instalar equipos dentro del termo y energizar los equipos con Arduino.
- Desarrollar la programación del termo para poder regular la temperatura.
- Visualizar los datos de la temperatura en el LCD 16x2.
- Realizar cinco pruebas para sistemas de refrigeración.

2. MARCO TEÓRICO REFERENCIAL

2.1. Controlador PID

Hay muchos tipos de controladores que se pueden utilizar en aplicaciones industriales de laboratorio y de rutina; algunos controladores más utilizados son On / Off, PID, Fuzzy y Neural, los dos últimos puntos son complejos ya que trabajan con inteligencia artificial, el controlador también puede distinguirse como controladores "feedforward" y "feedback", el trabajo del controlador de avance es el resultado que se da en concordancia con el siguiente paso:

- **SP=** Compara el punto de ajuste para encontrar el error.
- **PV=** El bloque del sensor mide la variable de proceso.

El controlador PID utiliza un circuito de retroalimentación que controla un parámetro del sistema llamado variable de proceso (PV), que coincide con la salida deseada del punto de ajuste (SP), el PV es cualquier cantidad física del sistema que pueda medirse utilizando el sensor.

El controlador PID utiliza un "error" la diferencia entre SP y PV es que en cada lazo para controlar el sistema, mide el nivel de error en cada bucle y entrega una señal a un actuador para que minimice la cantidad de el error del PID.(Planas, 2016)

Los diferentes términos de un algoritmo y sus características son descritos de la siguiente forma:

Término P: Esto solo considera el error actual, el término P-TERM calcula la distancia entre el PV y el SP en el momento actual, sin embargo, acercándonos cada vez más a SP el error se vuelve tan pequeño que el controlador no puede activar el PV para capturar el SP, por ende siempre hay un error de estado estable (SSE) que se expresa como el punto de ajuste en el sistema, no obstante el valor más alto del término proporcional puede activar el PV para establecer puntos pero hará que el sistema sea inestable debido a oscilaciones y sobretensiones, por lo tanto para el controlador la respuesta en este caso es más como la respuesta de un controlador de encendido / apagado, donde el término P se expresa como:

$$\text{P-Term} = K_p \cdot e(t) \quad C_p(s) = k_p \quad \text{Ec.(1)}$$

Donde: Ganancia proporcional K_p y $e(t)$ = error en el momento actual "t".

Término I: Este es el término más importante en los controladores PID, se considera todos los errores que existe dentro del sistema desde el punto de partida hasta un momento específico del proceso, el término I se expresa como:

$$\text{I-Term} = \int_0^t K_i \cdot e(t) dt \quad \text{Ec (2)}$$

Donde la ganancia integral y el tiempo total de funcionamiento del controlador se hace con la contribución I-Term que intenta equilibrar el PV en ambos lados del punto de ajuste.

Término D: Genera una señal de control proporcional a la derivada del error, por lo tanto el control derivado conoce la dinámica del error en donde este puede ser aumentando o disminuido, también produce correcciones antes de que la señal de error siempre que sea demasiado grande. (Planas, 2016)

$$u(t)k_d \cdot \frac{de(t)}{dt} \quad C_p(s) = k_d \cdot s \quad \text{Ec (3)}$$

2.2. Lazo Abierto

El lazo abierto en la salida no tiene acción de control, en otras palabras la salida del sistema de control de lazo abierto no se mide ni se realimenta al compararlo con la entrada. (GONZÁLEZ, 2014)

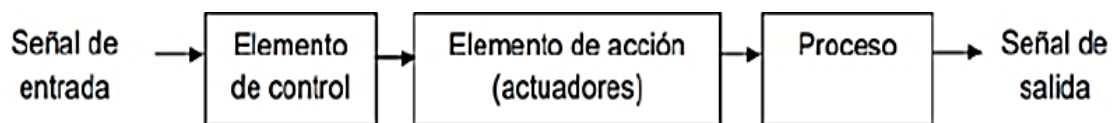


Figura 1. Lazo abierto. (Ogata, 2010)

2.3. Lazo Cerrado O Retroalimentado

El lazo cerrado monitorea continuamente la señal de salida y trabaja con la señal de referencia para calcular la señal de error que luego se aplica al controlador para que genere la señal de control e intente llevar la señal de salida al valor deseado. (POTINGO, 2011)

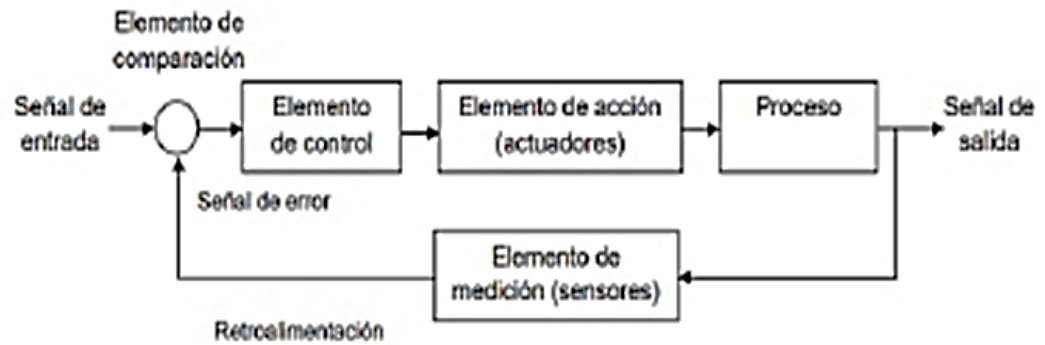


Figura.2 Lazo cerrado (Ogata, 2010)

2.4. Control ON/OFF

Es un sistema de control con solo dos posiciones fijas en muchos casos se encienden y apagan.(GONZÁLEZ, 2014)

2.5. Ley De Enfriamiento De Newton

La temperatura cambia a una velocidad cuando los fluidos entran en contacto con las superficies de diferentes temperaturas, el calor se transfiere a una velocidad que depende de la densidad del fluido y su viscosidad.(García, 2012)

$$\frac{dT}{dt} = k(T - T_m) \quad \text{Ec (4)}$$

Donde:

- T= Temperatura de un cuerpo.
- t= Tiempo.
- T_m= Temperatura del ambiente.

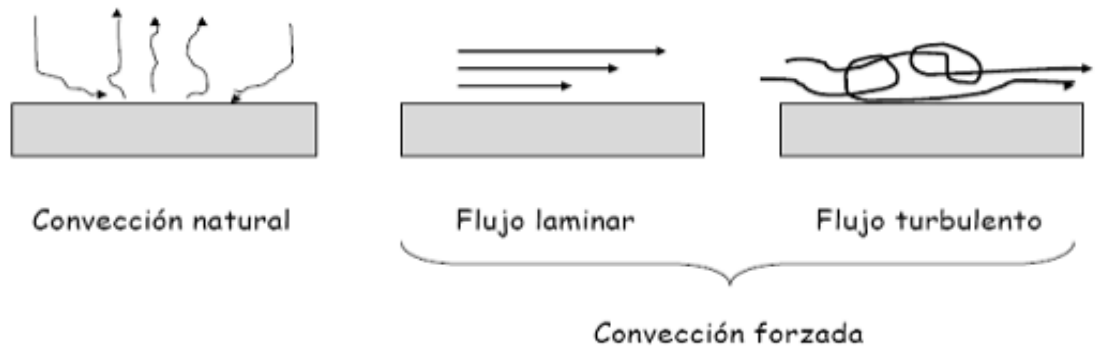


Figura 3. Ley de enfriamiento de Newton.(García, 2012)

2.6. Ley De Stefan-Boltzmann

La energía radiada por segundo por unidad de área por un radiador de cuerpo negro es proporcional a la cuarta potencia de la temperatura absoluta y viene dada por:

$$\frac{P}{A} = \sigma T^4 \quad \text{Ec (5)}$$

$$\sigma = 5,6703 \times 10^{-8} \frac{\text{vatios}}{\text{m}^2 \text{k}^4} \quad \text{Ec (6)}$$

Para objetos calientes que no sean radiadores ideales, la ley se expresa como:

$$\frac{P}{A} = e\sigma T^4 \quad \text{Ec (7)}$$

Donde e es la emisividad del objeto ($e = 1$ para un radiador ideal). Si un objeto caliente irradia energía a su entorno más frío a la temperatura T_c , la tasa de pérdida radiactiva neta toma la forma:

$$P = e\sigma A(T^4 - T_C^4) \quad \text{Ec (8)}$$

- T_C^4 =Temperatura circundante.
- A= Área radiante.
- e= Emisividad (=1 para radiador ideal).
- P= Potencia radiada neta.
- T= Temperatura del radiador.
- σ = Constante de Stefan.

La fórmula de Stefan-Boltzmann también está relacionada con la densidad de energía radiada en un volumen de espacio dado.(salamanca, 2017)

2.7. Método De Sintonización De Ziegler Y Nichols

Ziegler y Nichols realizan pruebas simuladas y experimentó con el equipo de su laboratorio, en este proceso se determinan los parámetros del controlador P, PI y PID, comenzando puramente con el controlador proporcional a la elección que encontraron, el controlador tiene la suficiente sensibilidad de ganancia que es necesaria para llegar a un compromiso entre las razones de la magnitud que es la respuesta en el decaimiento del error período sobre todo los errores permanentes. Se considera entonces que la atenuación es del 25% proporcional de un ajuste adecuado esto logrado con una sensibilidad que aproxima su mitad que es igual a la máxima sensibilidad. Para los modos integral y derivado se encontraron su ajuste que puede estar relacionado con oscilación crítica siempre amortiguada de ahí se dedujo la ecuación de ajuste del controlador.(Alfaro, 2011)

2.7.1 Tipo De Funcionamiento Del Lazo De Control

Ziegler y Nichols interpretan que al introducir un cambio de escalón a la salida del controlador a la entrada de la planta está siempre estará en tiempo continuo, es

importante recordar esto ya que estas reglas de ajuste solo es para afinar un controlador que operará como un regulador en lugar de ser un servo, las propuestas de nuevas reglas compara su afinación con las respuestas obtenidas del método de Ziegler y Nichols antes del cambio de escalón de valor deseado para demostrar su llamada superioridad, ya que estas comparaciones no son tan importante porque Ziegler y Nichols nunca se preocupan por las reacciones de los demás si no por los cambios en los valores del sistema de control requerido(Alfaro, 2011)

2.7.2 Método De Lazo Abierto

Como se mencionó anteriormente para usar las relaciones basadas en la información adquirida se prueba en lazo abierto donde es necesario dibujar unas rectas tangentes al mismo punto de inflexión para la determinación de la latencia (tiempo muerto aparente) y la pendiente de la recta, mientras que Ziegler y Nichols explica en cuanto al modelo de primer orden se toma en cuenta el tiempo muerto de la curva de respuesta para lograr un nuevo equilibrio, este modelo de planta que es más adecuado para la información obtenida de tangente que es un proceso integral dada por la ecuación:

$$G_p = \frac{Re^{-Ls}}{s} \quad \text{Ec(9)}$$

Es muy común encontrar ecuaciones de bucle abierto de Ziegler y Nichols para usar con parámetros de modelos de primer orden más el tiempo muerto, donde K_p es la ganancia estática, T es una constante y L es el tiempo muerto aparente.(Alfaro, 2011)

$$G_p = \frac{K_P e^{-Ls}}{Ts+1} \quad \text{Ec (10)}$$

2.7.3 Método De Lazo Cerrado

Los métodos de lazo cerrado se necesitan para saber los parámetros claves, como la ganancia k_{cu} y el período de oscilación T_U , para obtener la máxima estabilidad del sistema que utilice un controlador proporcional puro. (Alfaro, 2011)

2.7.4 Criterio De Desempeño

El criterio de rendimiento seleccionado por Ziegler y Nichols tiene un 25% de descomposición para un cambio de paso en la carga y la magnitud del error máximo decae en la cuarta parte de cada ciclo de oscilación. (Alfaro, 2011)

2.8. Efectos Termoeléctricos

Estos efectos implican la correlación entre las corrientes de transferencia de pares eléctricos tomando en cuenta la tensión, el calor y la temperatura que son aplicadas en el mundo de la refrigeración.

2.8.1. Efecto Peltier

Se refiere a la refrigeración o calentamiento entre dos conductores diferentes al pasar la corriente fluye a través de él y depende de la temperatura de unión. (Ramírez et al., 2017)

El efecto resultante es el flujo de calor lateral de un conductor o unión no uniforme con el valor:

$$q_p = -I.T(a_x - a_y) \quad \text{Ec (11)}$$

Donde:

- I = Intensidad de la corriente.
- T = Temperatura conocida.
- α_x y α_y son los coeficientes termoeléctricos de cada metal.
- q_p = Potencia calorífica generada por el efecto.

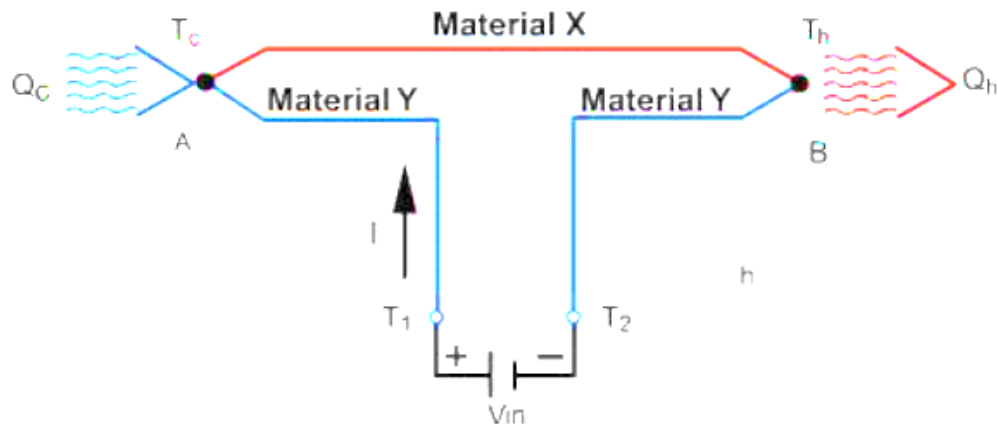


Figura 4. Esquema efecto peltier. (Campos, 2014)

2.8.2 Celda Peltier

Las celdas termoeléctricas estándar son las más simples de los dispositivos termoeléctricos, tienen un rango de temperatura de $\Delta T = 75^\circ\text{C}$, su temperatura en caliente tiene una temperatura máxima de 80°C y una temperatura ambiente de 25°C . (Romero, 2017)

Una celda peltier es un elemento compuesto por dos materiales semiconductores, el bismuto N y el telurio P que están unidos entre sí por una lámina de cobre que al conectar una fuente de alimentación de CC al elemento estará caliente y el otro lado estará frío, sin embargo, la polaridad de la fuente de alimentación en sus caras que forman la celda peltier hace que se comporte sucesivamente. (Serrano, 2019)

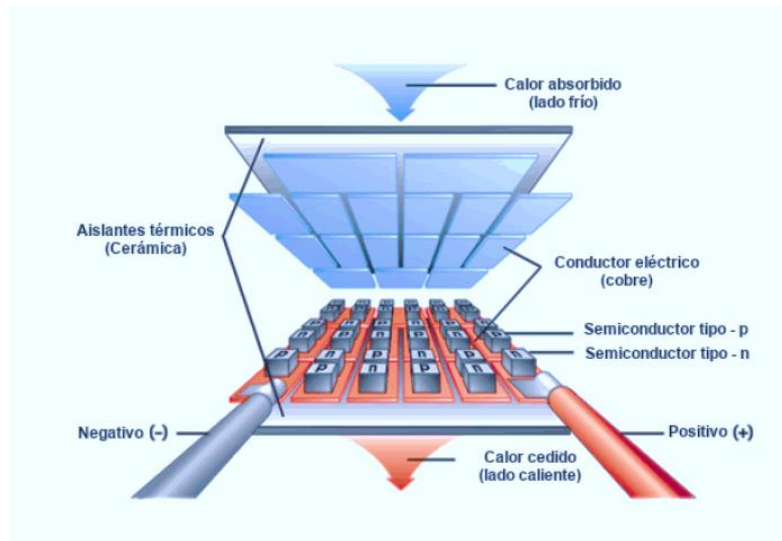


Figura 5. Celda peltier.(Cuyubamba, 2018)

2.8.2.1 Aplicaciones

Estas aplicaciones se muestran a continuación son:

- Aplicaciones médicas.
- Concentración a baja temperatura.
- Contenedores de transporte de alimentos.
- Medida de parámetros térmicos.
- Neveras y calentadores portables.
- Refrigeración por láser médico.
- Sistema de aire acondicionado.
- Sistema de calefacción.

Cuando un electrón se transfiere de un material dopado positivamente de tipo P a un material dopado negativamente de tipo N, su nivel de energía es más bajo que uno más alto y luego absorbe energía mientras cuando los electrones de un material de tipo N dopado negativamente a un material de tipo P dopado positivamente, los electrones se transfieren a un nivel de energía más bajo que luego libera energía por el calor.(Romero, 2017)

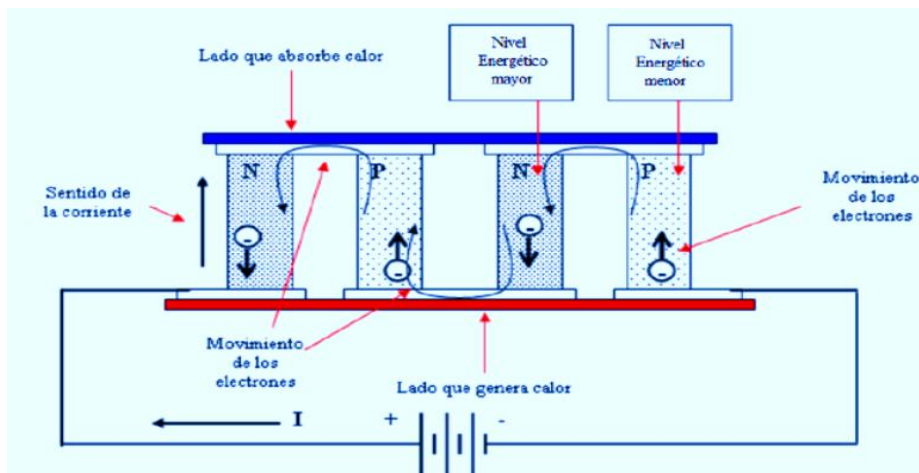


Figura 6. Efecto producido por corriente en una celda peltier. (Romero, 2017)

2.8.2.2 Propiedades

Estas propiedades se muestran a continuación:

- Funciona con tensión continua.
- No produce ruido ni vibraciones.
- No requieren mantenimiento.
- Son muy fiables.
- Son pequeños y ligeros.
- Tienen una vida útil de más de veinte años.

Los modelos de las celdas peltier producen un desplazamiento de calor o frío esto va a depender de la corriente suministrada:

NO.	CATALOG NUMBER	Th=25°C				N	DIMENSIONES (mm)		
		Qmax(Watts)	I _{max} (Amps)	V _{max} (Volts)	T _{max} (°C)		A	B	C
1	TEC1-12703	38	3.3	15.4	68°C	127	40	40	4.7
2	TEC1-12704	40	4.3	15.4	68°C	127	30	31	3.3
3	TEC1-12706	63	6.4	15.4	68°C	127	40	40	3.9
4	TEC1-12707	75.5	7.4	15.4	66°C	127	40	40	3.5
5	TEC1-12710	100	10.5	15.4	68°C	127	40	40	3.3
6	TEC1-12715	180	15.6	15.4	80°C	127	40	40	3.8
7	TEC1-12730	350	30.7	15.4	90°C	127	62	62	3.9

TABLA 1. Especificaciones técnicas celdas peltier. (Guerrez, 2017)

La dimensión más común de la celda peltier son:

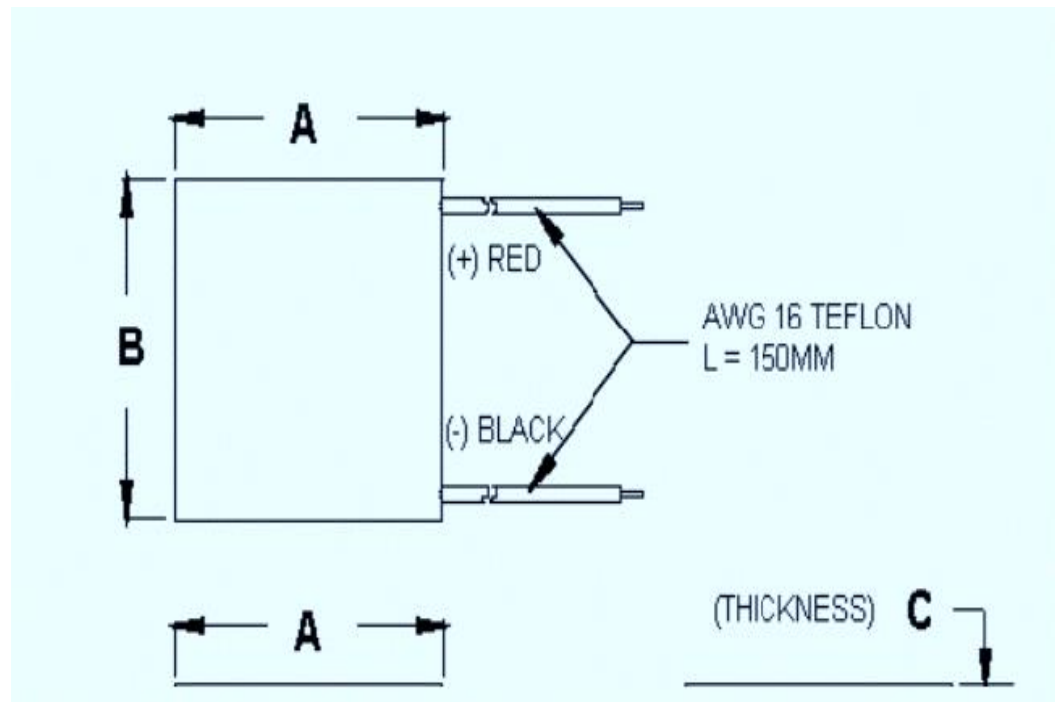
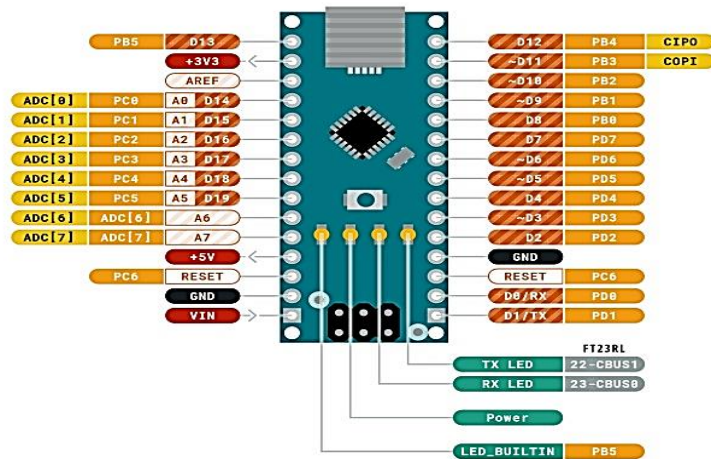


Figura 7. Dimensiones físicas de la celda peltier TEC1.12706.(Guierrez, 2017)

2.9. Arduino Nano

Arduino nano es una placa compacta compatible con placas de pruebas, esta cumple con todas las funciones basadas en el microcontrolador ATmega328P.

Tiene la misma funcionalidad que el Arduino UNO en términos de potencia y conectividad del microcontrolador solo que su conector USB es diferente, tomando en cuenta el conector de alimentación y los pines cambian el formato del pin de cabecera.(Arduino, 2022)



Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

ARDUINO . CC
This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1888, Mountain View, CA 94042, USA.

Figura 8. Tarjeta Arduino nano.(Arduino, 2022)

2.9.1 Características Técnicas Arduino Nano

Estas características se muestran a continuación:

NOMBRE	ARDUINO NANO
MCU	Atmega328p/ Atmega168.
ALIMENTACIÓN DE ENTRADA	5 v
ENTRADA DE VOLTAJE RECOMENDADO	7-12v
Máxima CORRIENTE POR CONECTOR DE ENTRADA Y SALIDA	40 miliamperio
FRECUENCIA DE RELOJ	16Mhz
MEMORIA FLASH	16/32k
USB	mini
SRAM	1KB/2KB
PWM	6 de 14pin digital
GPIO (PIN DIGITAL)	14
PIN ANALÓGICO	8
EEPROM	512bytes/1kb

TABLA 2. Características técnicas Arduino Nano.(Watson, 2018)

- **Vin:** La tensión de alimentación de entrada de la placa cuando se utiliza una fuente de alimentación externa de 7 a 12 V.
- **5V:** Es el voltaje de suministro regulado para la placa, que se utiliza para alimentar el controlador y otros componentes de la placa.
- **3V3:** Este es el voltaje mínimo producido por el regulador de voltaje.
- **Pines GND:** Estos son los pines de tierra en la placa, hay varios pines de tierra en la placa y se pueden interconectar.
- **Pines de reinicio:** El Arduino nano tiene 2 pines de reinicio integrados en la placa, si cualquiera de estos pines de reinicio está bajo se reiniciará el microcontrolador.
- **Pin No. Pin #13** - El LED incorporado está conectado al pin #1, #13, este LED se utiliza para comprobar si la placa funciona correctamente.
- **AREF:** Este pin se usa como referencia para el voltaje de entrada.
- **Pines analógicos:** Hay 8 pines analógicos en la placa etiquetados A0-A7, estos pines se utilizan para medir voltajes analógicos que van desde 0 a 5V.
- **Pines digitales:** El Arduino nano tiene 14 pines digitales que van desde D0 a D13, estos pines digitales se utilizan para conectar sensores y módulos digitales de terceros a la placa nano.
- **Pines PWM:** Arduino nano tiene 6 pines PWM que son Pin #3, 5, 6, 9, 10 y 11 (son todos pines digitales), estos pines se utilizan para generar señales PWM (modulación de ancho de pulso) de 8 bits.
- **Interrupción externa:** Los pines 2 y 3 se utilizan para generar interrupciones externas, generalmente utilizadas en situaciones de emergencia cuando necesitamos detener el programa principal y llamar a instrucciones importantes el programa principal reanuda la interrupción.
- **Pines seriales:** Estos pines se utilizan para la comunicación serial donde:
 - Pin#0 es RX para recibir datos en serie.
 - El pin #1 es el Tx utilizado para transmitir datos en serie.
- **Protocolo SPI:** Tiene cuatro pines 10 (SS->Slave Select), 11 (MOSI -> Slave In Master In), 12 (MISO -> Slave Out Master In) y 13 (SCK -> Serial Clock) se utilizan para SPI (serial).El Protocolo de interfaz periférica (SPI) es un bus de interfaz utilizado principalmente para transferir datos entre microcontroladores y otros periféricos como sensores, registros, tarjetas SD, etc.
- **Protocolo I2C:** La comunicación I2C se realiza mediante los pines A4 y A5, donde A4 significa línea de datos en serie (SDA) para transportar datos y A5 significa línea de reloj en serie (SCL), que es la señal de reloj generada por el maestro para la sincronización de datos I2C.

2.9.2 Aplicaciones Y Propiedades

Se puede controlar elementos e interactuar con el hardware y software, además desarrollar procesos autónomos donde tenemos:

Aplicaciones:

- Drones y rovers.
- Impresoras 3D y máquinas CNC.
- Producto comercial.
- Productos domésticos.
- Tecnología robótica.

Propiedades:

- Advertencia temprana.
- Automatización de informes/cuadros de mando.
- Automatización de procesos.
- Eficiencia energética.
- Instalación de control remoto.
- Mantenimiento predictivo
- Monitoreo en tiempo real.

2.10. Control Driver VNH2SP30

Este control driver vnh2sp30 se utiliza para controlar la dirección del motor y a la vez también se utiliza para controlar la velocidad donde se tiene una tolerancia de corriente alta la cual puede soportar corriente directa de 14A además trabaja con una alimentación entre 5,5V y 16V, se tiene en cuenta el pin que está conectado al MOSFET entre más largo sea el ciclo de trabajo mayor será el voltaje producido.(Mohammad, 2020)

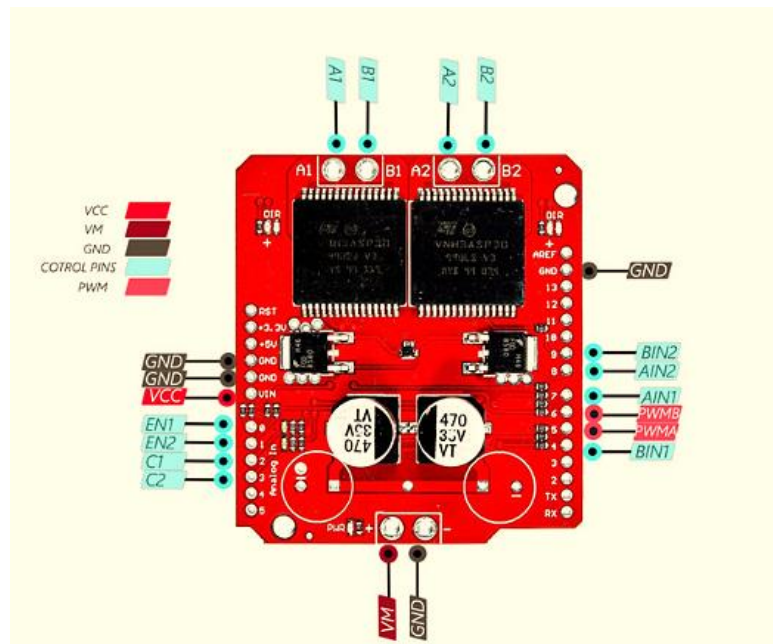


Figura 9. Controlador driver vnh2sP30. (Mohammad, 2020)

2.10.1 Características Técnicas De Motor Monster VNH2SP30

- **A1:** Extremo positivo del motor A.
- **A2:** Extremo negativo del motor A.
- **AIN1:** Señal de control para el motor A.
- **AIN2:** Señal de control para el motor A.
- **B1:** Terminal positivo para motor B.
- **B2:** Terminal negativo del motor B.
- **BIN1:** Señal de control para motor B.
- **BIN2:** Señal de control para el motor B.
- **C1:** Señal de medida de corriente del motor A.
- **C2:** Señal de medida de corriente del motor B.
- **EN1:** Señal de activación del motor A.
- **EN2:** Señal de activación del motor B.
- **PWMA:** Señal de control de velocidad para el motor A.
- **PWMB:** Señal de control de velocidad para el motor B.
- **TIERRA:** Tierra.
- **VCC:** Entrada de alimentación.
- **VM:** Voltaje del módulo.

2.11. Pantalla De Cristal Líquido 16X2

Los módulos de pantalla de cristal líquido (LCD) que muestran caracteres como texto y números son los más baratos y fáciles de usar en todos estos monitores LCD, se pueden comprar en varios tamaños que se miden por la cantidad de filas y columnas de caracteres que pueden contener una interfaz con HD44780 o KS0066 que es compatible con el Arduino, este módulo electrónico es muy básico recomendado en diversos dispositivos y circuitos comparados con otros LED de multisegmento estos son baratos fáciles de programar y se pueden personalizar caracteres a diferencia como el del siete segmentos, este puede mostrar 16 caracteres y hay 2 líneas.

El registro de comando almacena las instrucciones dadas al LCD para realizar acciones predefinidas, como tareas, borrar la pantalla, establecer la posición del cursor, controlar la pantalla, etc.

El registro de datos almacena y se muestra en la pantalla LCD los datos que es el valor ASCII del carácter.(Ufoaroh et al., 2015)

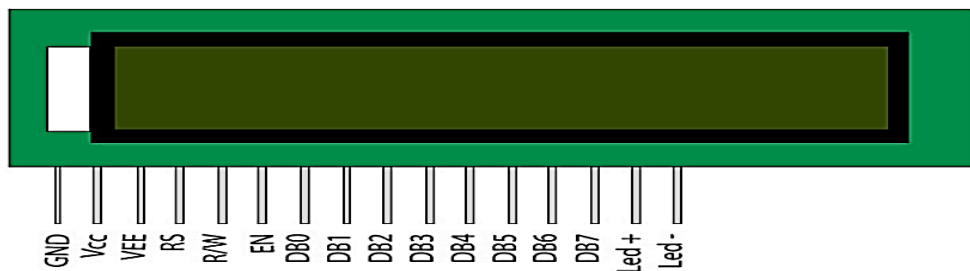


Figura 10. LCD16x2. (NIKITHA & SURESH, 2016)

2.11.1 Características Técnicas Lcd 16X2

Estas Características se muestran a continuación:

PIN Nº	FUNCIÓN	NOMBRE
1	Tierra (0 V)	Ground
2	Tensión De Alimentación; 5 V (4,7 V – 5,3 V)	Vcc
3	Ajuste De Contraste; A Través De Una Resistencia Variable	VEE
4	Selecciona El Registro De Comando Cuando Está Bajo; Y Registro De Datos Cuando Es Alto	Seleccionar Registrarse
5	Bajo Para Escribir En El Registro; Alto Para Leer El Registro	Lectura/escritura
6	Envía Datos A Pines De Datos Cuando Se Da Un Pulso Alto A Bajo	Habilitar
7		DB0
8		DB1
9		DB2
10	Pines de Datos De 8 Bits	DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	VCC de Retroiluminación (5 V)	Led+
16	Tierra de Retroiluminación (0 V)	Led-

TABLA 3. Características técnicas LCD 16x2. (Nangoi, 2015)

2.12. Módulo LCM 1602 I2C

El uso del módulo convertidor I2C está destinado a guardar la cantidad de pines utilizados en el LCD 16X2 donde su interfaz I2C de 4 pines permite la conexión y la comunicación, cuenta con 16 pines de entrada y 4 salidas digitales controlados por protocolo I2C.(GUALPA, 2018)

Donde:

- Pin GND: Tierra.
- Pin SCL: Línea de sincronización en el protocolo I2C o el reloj.
- Pin SDA: Protocolo I2C línea de datos.
- Pin VCC: 5 V DC.

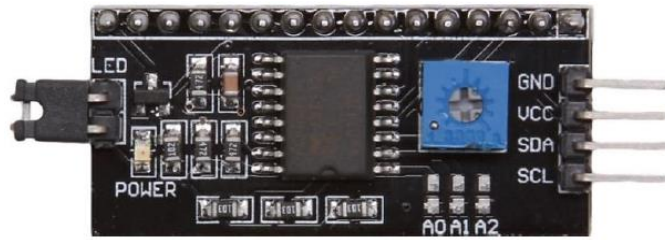


Figura 11. Módulo LCM 1602 I2C.(GUALPA, 2018)

2.13. Protocolo De Comunicación I2C

El protocolo contiene la conexión de interfaz de bus que está integrado en dispositivos de comunicación en serie I2C que significa circuito integrado. Fue creado por Philips Semiconductors en el año 1982, este protocolo de comunicación de corto alcance ampliamente utiliza la línea de datos (SDA) que transporta datos, mientras que la línea de reloj (SCL) está sincronizada en el transmisor y el receptor a transferir, los dispositivos requieren muy pocos pines porque realizan la misma función que sus pines equivalentes.

El bus I2C tiene tres modos de operación tales como el modo estándar (0 a 100 kbps), modo rápido (0 a 400 kbps) y modo de alta velocidad (0 a 3,4 Mbps).(Kaith et al., 2015).

2.14. Sensor De Temperatura DS18B20

El DS18B20 es un sensor de temperatura digital con una resolución ajustable de 9-12 bits que tiene una precisión de medición de $\pm 0,5$ °C de -10 °C a +85 °C más allá de este rango de temperatura la precisión se reduce o se vuelve menos preciso, aunque el puede medir temperaturas de -55°C a 125°C también se direcciona a través del llamado "One-Wire-Bus". Esto se hace posible al conectar varios sensores a un pin de datos en el Arduino al mismo tiempo y consultarlos uno tras otro, es muy utilizado en industria

porque es digital donde su configuración es no volátil y se puede tener una conexión de 128 esclavos en paralelo.

El término cable único proviene del hecho de que el sensor puede funcionar con un solo cable de datos y el GND sin alimentación adicional, sin embargo, a menudo se informan problemas especialmente cuando se utilizan varios sensores y cables más largos, por lo que el sensor debe recibir la alimentación correctamente. (Fezari & Dahoud, 2019)



Figura 12. Sensor de temperatura DS18B20. (kiwi, 2022)

DS18B20 tiene 3 pines en total, que son:

- **Pin # 1:** Vcc (Se tiene que proporcionar +5V).
- **Pin # 2:** Pin de datos (donde se obtiene las lecturas de temperatura).
- **Pin # 3:** GND (Se tiene que proporcionar tierra aquí).

2.14.1 Aplicaciones

- Control de termostatos.
- Sistemas industriales.
- Sistemas térmicos.
- Temperatura de productos de consumo.

2.15. Módulo Bluetooth HC-06

El módulo HC-06 incorpora la tecnología bluetooth también conocida como SPP (Port Protocol serie), que está diseñada para establecer una conexión inalámbrica mediante el puerto serie del dispositivo al que lo conectamos.

De fábrica viene configurado como esclavo, es decir se mantiene esperando recibir un comando del maestro, además tiene unas pocas instrucciones y su conexión es sencilla.(Mucientes, 2021)

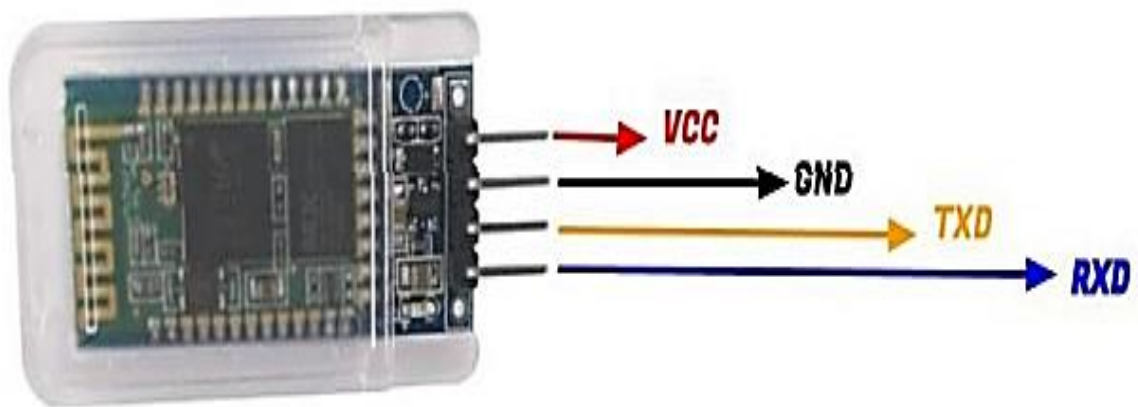


Figura 13. Módulo Bluetooth HC-06.(Mucientes, 2021)

Donde:

- Pin de Voltaje (-) =**GND**.
- Pin de Recepción=**RX**.
- Pin de Transmisión de datos =**TX**.
- Pin de Voltaje (+), 3.3v a 6v= **Vcc**

2.16. Power Suppli 12VDC-15A

La fuente de alimentación de 12VDC convierte la corriente alterna en continua, el modelo EELPB1215 tiene su entrada de 110 VCA a 220VCA y una salida constante de 12 V DC, esta fuente de voltaje tiene un sistema de protección contra la sobre carga donde su corriente máxima de recepción es de 15A, por lo tanto, cuando se supera esta corriente el sistema de protección actúa para evitar daños. (Soto, 2015)



Figura 14. Power Suppli 12vDC-15A.(Alibaba, 2022)

2.17. Calentador Eléctrico

El calentador eléctrico entre los equipos de transferencia de calor es de uso común, su gran importancia es para el ahorro de energía y la reducción de emisiones, los calentadores eléctricos aumentan la temperatura a través del precalentamiento, el calentamiento y otros procesos están fabricados en acero inoxidable y tienen la ventaja de que son corrosivos para los fluidos.(Márquez et al., 2016)



Figura 15. Calentador Eléctrico.(amazon, 2022)

2.18. Matlab

Matlab es creado a partir que los ingenieros y científicos usan computadoras para resolver una variedad de problemas, incluyendo desde la evaluación de funciones simples hasta la solución de sistemas de ecuaciones, por lo tanto Matlab tiende a convertirse en el entorno informático técnico elegido por muchos ingenieros donde además consta con un sistema interactivo que combina la computación numérica, la visualización científica y la computación simbólica.(Etter, 2011)

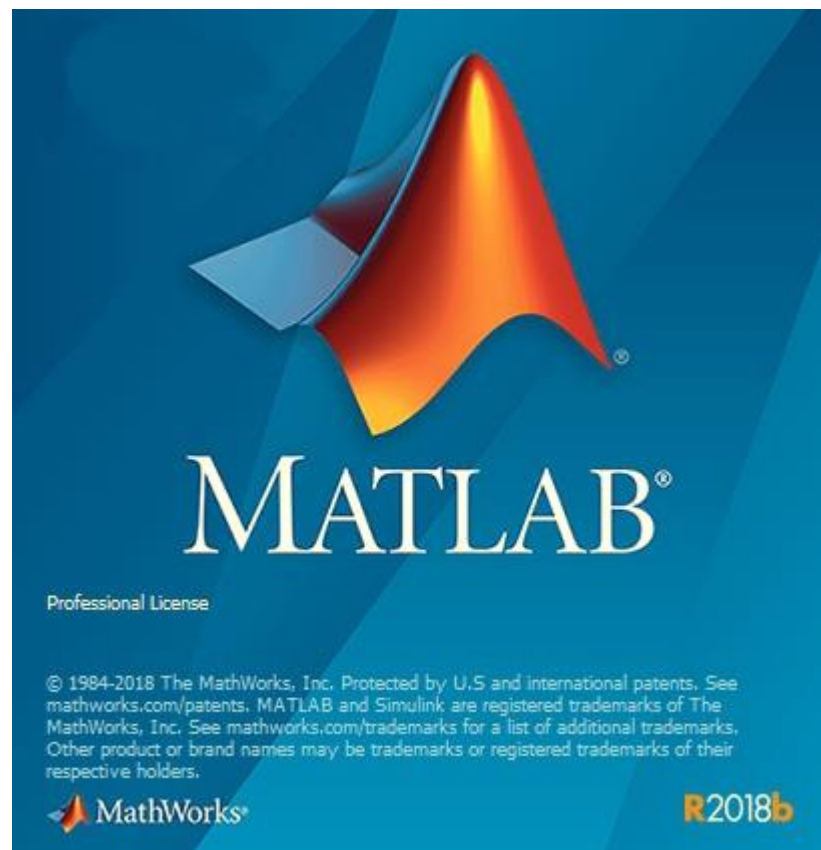


Figura 16. Software Matlab.(Moler, 2018)

2.19. Android

Es un sistema para teléfonos móvil basado en Linux este sistema es uno de los más utilizados en el mundo. Su programación es de código abierto lo que lo hace muy popular ya que permite a los fabricantes, operadores y desarrolladores brindar más funcionalidades. Android es gratuito y multiplataformas para una posible combinación de hardware y software.(Herraiz, 2012)



Figura 17. Android.(Innovationm, 2021)

2.20. Mit App Inventor

Es una creación de programación innovadora que permite convertir lenguajes de codificación de textos complejos en estructuras de bloques, lo que da como resultado una estructura visual de arrastrar y soltar. Esta no se basa en la programación informática sino en la "programación informática educativa", que pretende democratizar el desarrollo de aplicaciones a través de los usuarios jóvenes para que no sean consumidores de tecnología sino creadores.(appinventor, 2022)



Figura 18. Logo APPInventor.(appinventor, 2022)

3. Diseño

3.1. Funcionalidad

En este proyecto se pretende aprovechar a detalle las dos funcionalidades que tiene a controlar el termo portátil mediante un sistema controlado por PID, viendo su estructura de control y dar a conocer la tarjeta controladora Arduino nano ya que cuenta con diversas aplicaciones y módulos compatible.

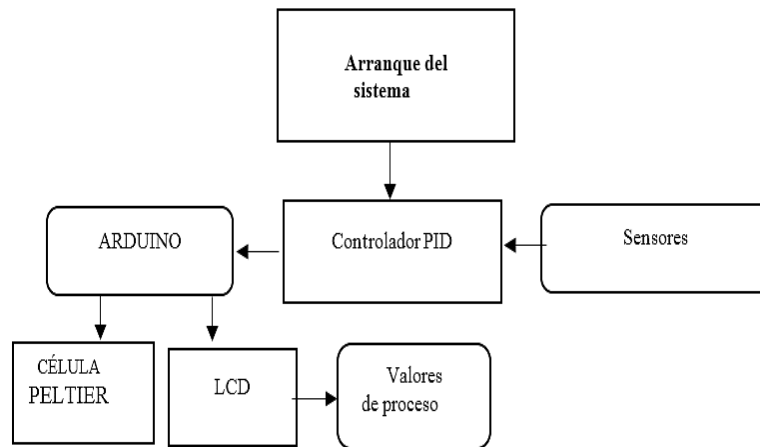


Figura 19. Diagrama en bloques del proceso.

El PID y las configuraciones de los módulos y actuadores electrónicos del termo portátil se encuentran en la programación del Arduino nano, la función del control driver vnh2sp30 es un amplificador de corriente que permite alimentar las celdas peltier TEC1-12706, esta comunicación se configura el Arduino nano de manera adecuada para utilizar sus pines de salidas, de no ser así no podría funcionar las celdas peltier TEC1-12706, porque el Arduino nano entrega un máximo de 40 miliamperio por pin y no es suficiente para poder alimentar a la celda peltier TEC1-12706 ya que el consumo por cada celda es de 6 amperios máximo, por ende podría causar dos cosas que mi Arduino

nano se queme por no generar dicha corriente o que las celdas peltier TEC1-12706 no vayan a funcionar.

El prototipo utiliza una fuente de poder centralizada de 12 Voltios DC a 15 Amperios, esta fuente va a suministrar el voltaje y amperaje necesario, para poder trabajar de una manera segura y eficaz. Para poder encender el termo portátil se utiliza un interruptor de bloqueo de dos posiciones, su función es encender y apagar el sistema del equipo este funciona como un interruptor normal, la diferencia es que para manipular el encendido y apagado del termo portátil se necesita ingresar una llave y esto crea una seguridad al momento de que una persona no tenga el conocimiento requerido para manipular de manera adecuada el equipo.

3.2. Funcionalidad manual o analógica.

En la parte manual se utiliza un switch de dos posiciones, la primera posición tiene que estar activado en la parte manual, esto va a verificar con un led de color amarillo. El sensor DS18B20 nos brinda los datos de temperatura dentro del envase de acero inoxidable al realizar el SP (set point), se considera el sensor digital donde el pin doce del Arduino nano nos dará los datos de lectura de la temperatura deseada que se necesita alcanzar, el SP este se manipula manualmente por el potenciómetro 10kΩ en el pin A6. Una vez establecida la temperatura el Arduino comienza a trabajar el PID con ayuda del sensor programado este comparará la temperatura deseada del SP con la temperatura del sensor, esto hace que estas temperaturas se igualen, para alcanzar estas dos temperaturas donde se emite una señal PWM generada por el Arduino al control driver vnh2sp30 para suministrar la corriente necesaria a las celdas peltier TEC1-12706 y así poder enfriar el líquido del recipiente para visualizar esta temperatura en el LCD 16x2.

Una vez alcanzada la temperatura deseada por el SP, las celdas peltier TEC1-12706 deja de trabajar al 100% de su capacidad.

3.3. Funcionalidad Por Teléfono

La segunda posición del interruptor va a encender un diodo led de color rojo como indicador para que el usuario sepa el modo en que se encuentra, también en esta posición se deshabilita el potenciómetro de la manera manual. El módulo bluetooth HC-06 que tiene un alcance de 10 metros de cobertura alrededor del termo portátil trabaja con una frecuencia de banda ISM (bandas de frecuencia industrial, científica y medica) de 2.4 MHZ , donde se busca establecer la comunicación por los pines D2 y D3 del Arduino a un teléfono smartphone con sistema operativo Android, esta aplicación funciona de la siguiente manera donde tiene 3 botones:

- El primer botón comunica el Bluetooth con el smartphone.
- El segundo botón funciona de manera máxima a 23°C al utilizar las celdas peltier TEC1-12706, esta temperatura se observar en el display LCD 16x2.
- El tercer botón permite parar la aplicación y el proceso del termo, al cerrar la aplicación hay que volver a configurar el bluetooth.

Se considera que esta programación hecha en Arduino nos va a servir tanto como manera manual como para establecer la comunicación vía bluetooth, en cuanto a las conexiones física se utiliza cable de color negro para el negativo y blanco para el positivo que se alimenta con 12DC donde también se establece una tierra común en la placa PCB para energizar el equipo por completo y evitar anomalías con los módulos.

3.4. Diseño Envase Acero

El envase que se observa en la figura 20, está realizado en AutoCad este diseño propuesto nos ayudara a tener una previa digital de como va a quedar en la vida real.

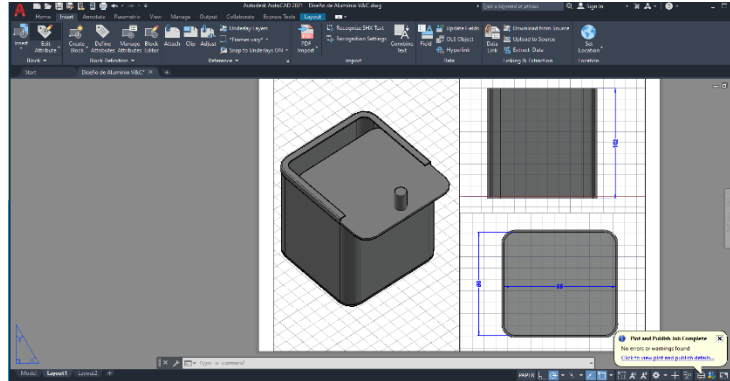


Figura 20. Diseño de envase de acero.

Se realizo el envase de un material hecho de acero ya que este tiene propiedades como la soldabilidad a diferencia con otros tipos de metales, otra propiedad que tiene el acero es que consta también de dureza y el tiempo de oxidación es menor a otro metal.

Ver el anexo E.



Figura 21. Diseño físico envase de acero.

3.5. Diseño 3d Termo Portátil

El diseño que se observa en la figura 22, está realizado en FreedCad para su respectiva impresión 3D en PLA este diseño va a cubrir el envase hecho de acero inoxidable. Ver anexo E.

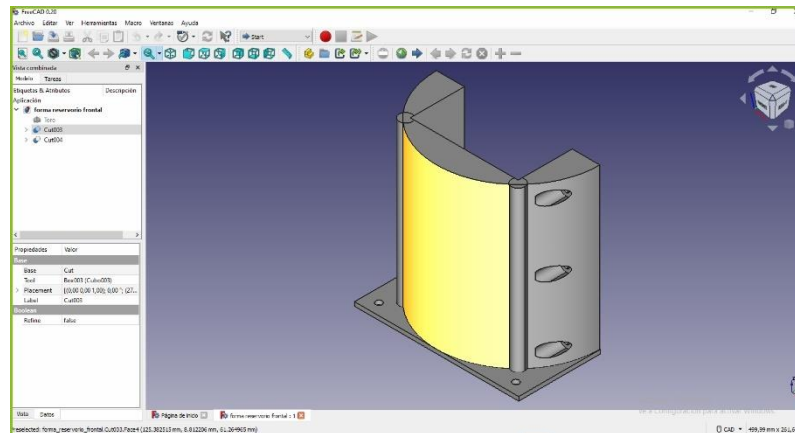


Figura 22. Diseño 3D parte frontal contorno del termo.

Se selecciona el material para imprimir en 3D, el PLA es también conocido como ácido poliláctico este tiene un nivel de inflamabilidad bajo es más resistente que el polietileno, también se añadieron ventoleras para que tenga una mejor disipación y rendimiento al momento de trabajar.



Figura 23. Diseño 3D parte trasera contorno del termo.

3.6. Diseño De Tarjeta PCB

El diseño de la tarjeta PCB esta realizado en proteus, este programa permite realizar conexiones de manera esquemática antes de pasarlo a PCB, también es un software que permite simular circuitos electrónicos. Ver anexo E.

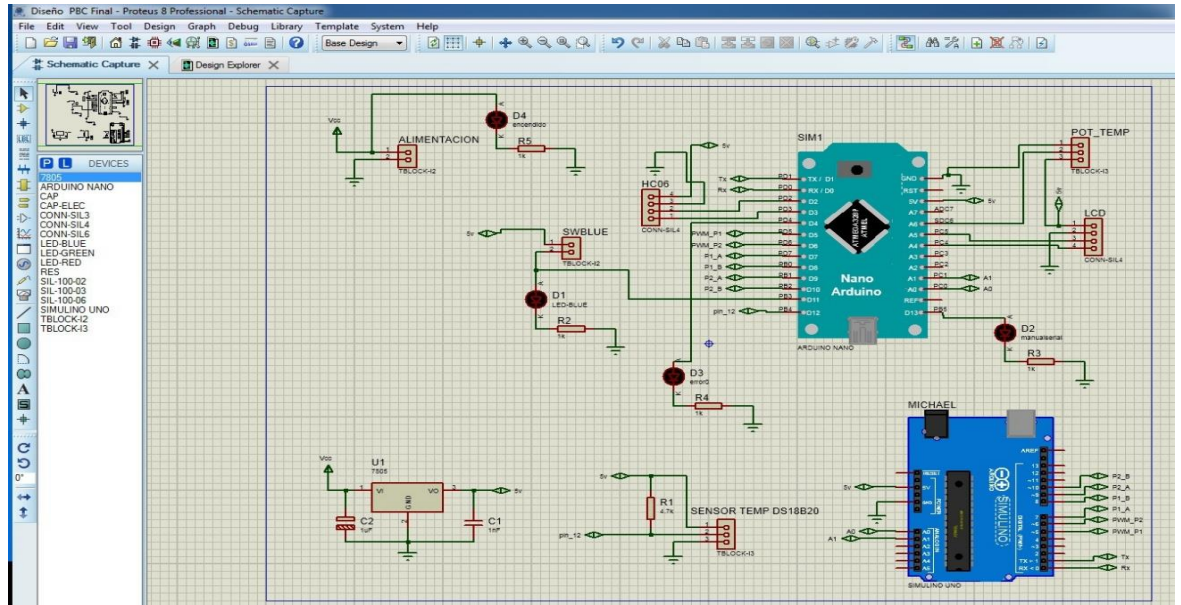


Figura 24. Esquemático Proteus.

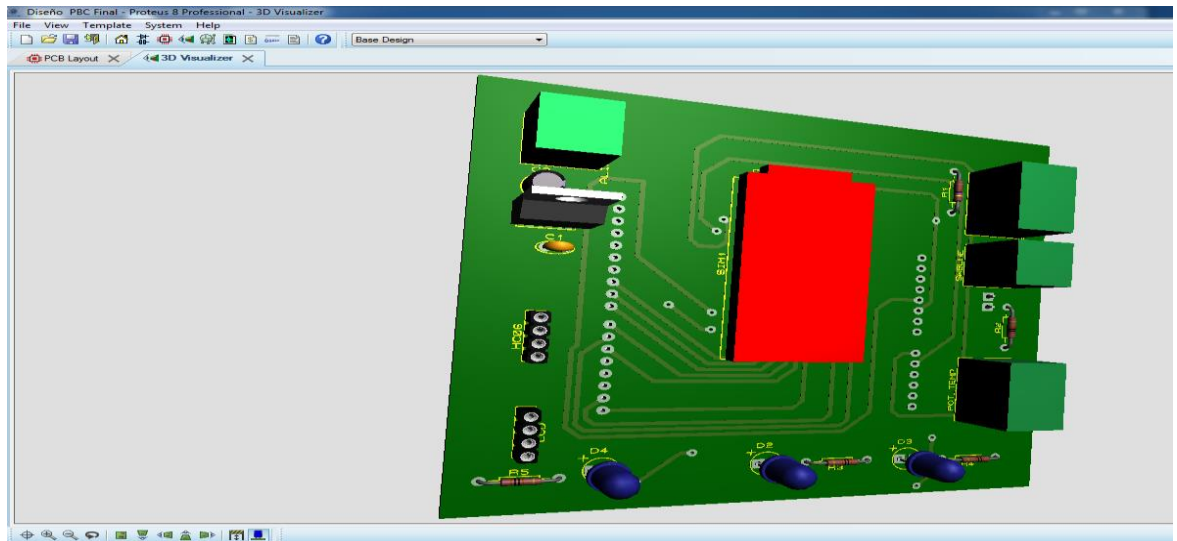


Figura 25. Diseño PCB 3D.

3.7. Diseño Aplicación App Inventor

La aplicación app inventor es diseñada para teléfonos que soportan el software Android, esta aplicación va permitir manipular las celdas peltier TEC1-12706 para que enfríen al máximo. Ver anexo E.

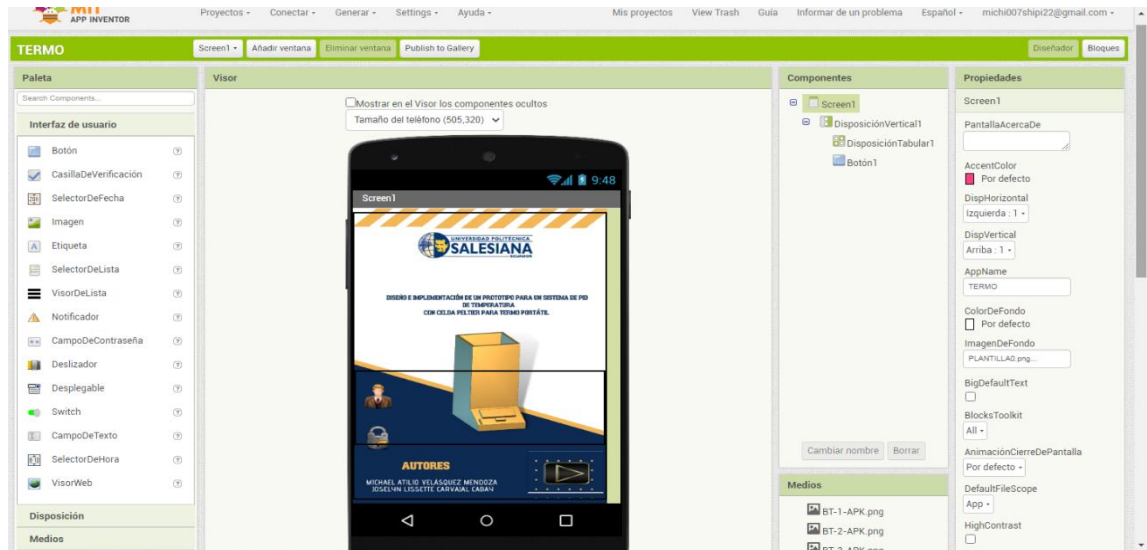


Figura 26. Pantalla principal APK.

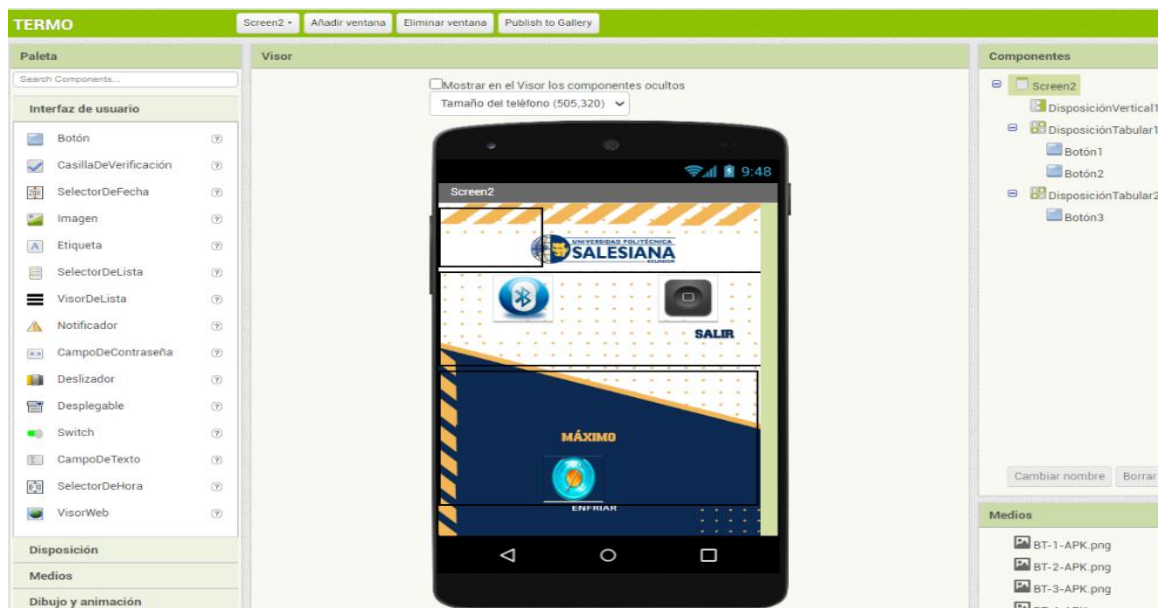


Figura 27. Pantalla de comunicación vía bluetooth.

3.8. Diseño Del Algoritmo Android

3.8.1. Comunicación De La Aplicación Apk Android Con Arduino

La lógica de los bloques de programación que es de arrastrar y soltar esta establece comunicar el módulo de Arduino y el smartphone con software Android.

Cuando se ejecuta botón 1 se reproduce el sonido 1 y abre la siguiente pantalla llamada Screen 2.

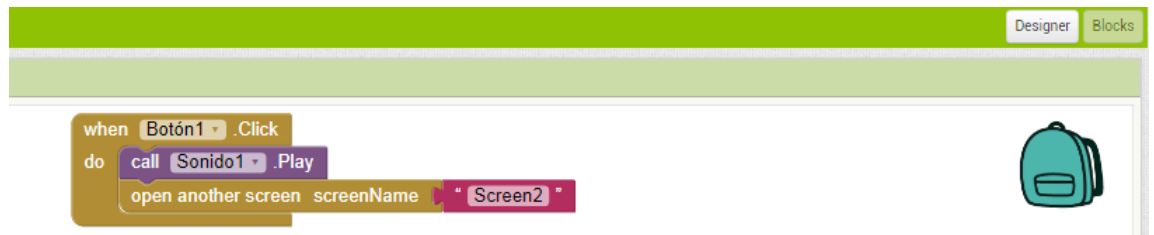


Figura 28. Botón Inicio Programación Screen 1.

Cuando se ejecuta botón bluetooth 1 antes de selección, esta busca todos los módulos activados de bluetooth aparece una lista para seleccionar el adecuado.

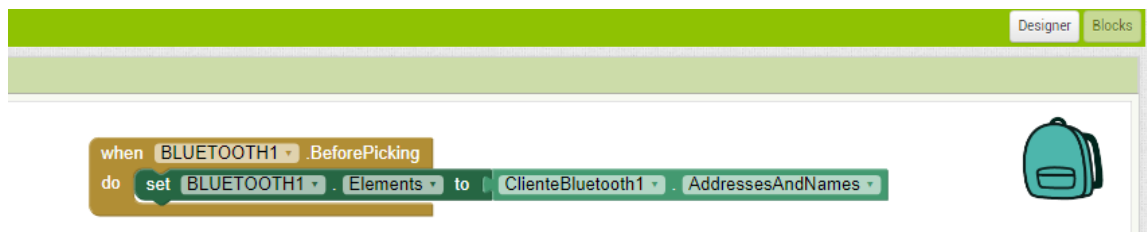


Figura 29. Botón de bluetooth selección de lista Screen 2.

Cuando se ejecuta botón bluetooth 1 después de selección, el módulo de bluetooth que tiene por nombre proyecto de tesis se vincula con el módulo Arduino.

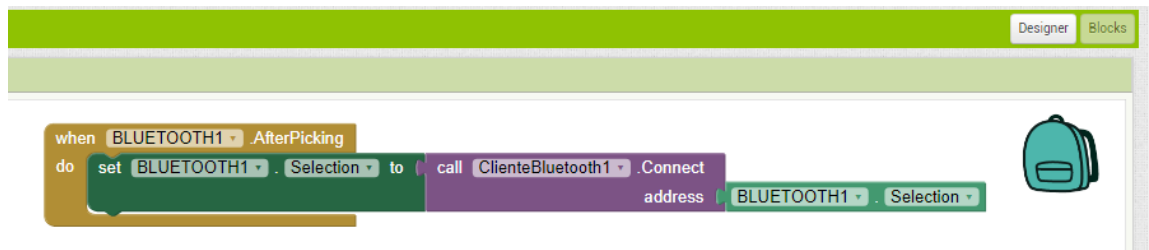


Figura 30. Botón de bluetooth selección de lista Screen 2.

Cuando se ejecuta el botón enfriar, envía un texto con el nombre de encender para que el módulo de Arduino pueda ejecutar la acción deseada que es la de hacer trabajar las celdas peltier al máximo.

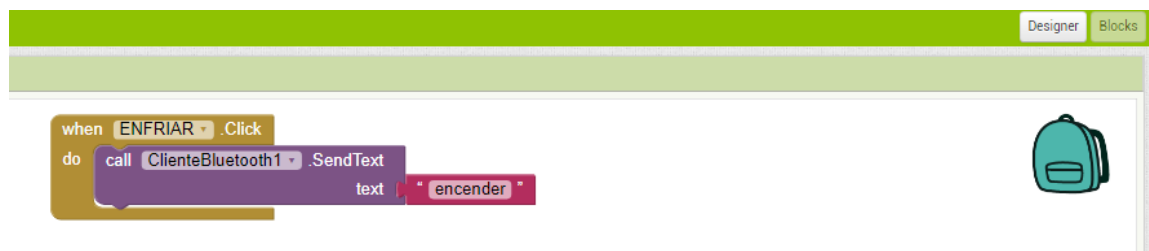


Figura 31. Botón enfriar Screen 2.

Cuando se ejecuta el botón apagar, envía un texto con el nombre apagar para que módulo Arduino pueda ejecutar la acción apagar.

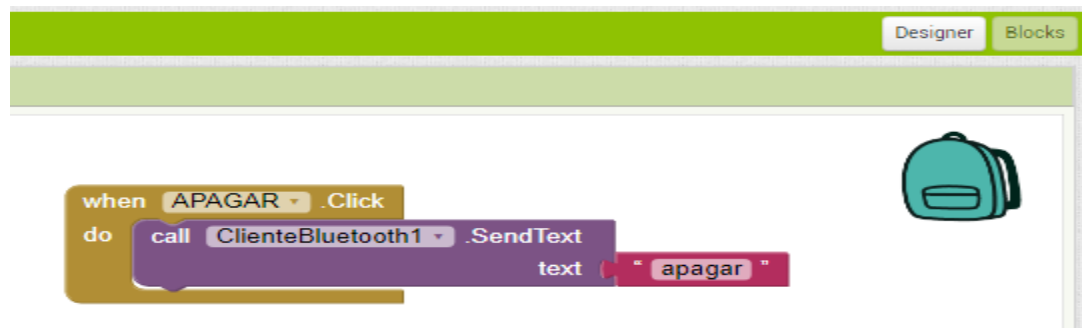


Figura 32. Botón apagar Screen 2.

4. IMPLEMENTACIÓN

4.1 primera parte de ensamblaje termo portátil 3D

La primera parte del ensamble consta del montaje de las celdas peltier al envase de acero inoxidable y su diseño creado en 3D.

El preparado de pegamento para colocar los disipadores de aluminio dentro del reservorio de acero inoxidable, se utilizó un pegamento llamado Home 2 Ron Epoxy clear de marca DECON, tiene un tiempo de secado de unas dos horas, una vez seco este pegamento soporta rangos de temperatura de -60°F a 180°F , es resistente al agua y anticongelantes esto lo hace ideal para pegar los disipadores de aluminio.



Figura 33. Ensamble 1.

Se debe mezclar el pegamento dentro de un envase por 1 minuto para que pueda pegar de manera eficiente los disipadores.



Figura 34. Ensamble 2.

Una vez mezclado por un minuto se debe colocar con cuidado el pegamento para poder pegar los disipadores de manera correcta.



Figura 35. Ensamble 3.

La espera de dos horas es esencial para manipular el recipiente y darle el uso para hacer los análisis y obtención de datos.



Figura 36. Ensamble 4.

Se debe verter agua en el recipiente de acero inoxidable para comprobar que no tenga ninguna fuga tanto por la parte de abajo como la parte donde va ir conectado el sensor de temperatura DS18B20 que estará ubicado en la parte lateral, antes de colocar la lana de vidrio que cubre el alrededor del recipiente se debe evitar los accidentes de fuga de líquido a las tarjetas electrónicas que se ubican en la parte de abajo del recipiente.



Esta lana de vidrio tiene diversas aplicaciones, se puede adaptar a la estructura que se desea usar, esta se caracteriza por ser ligera y como aislante térmico que no se corroe donde tiene una absorción de humedad de 20°C y 60% resistente a los rayos UV.



Figura 38. Ensamble 6.

Para poder envolver la lana de vidrio, se toma las medidas con una regla, estas nos van a permitir envolver de una forma correcta el recipiente, sin sobrantes de lana de vidrio.



Figura 39. Ensamble 7.

Una vez envuelto el recipiente de acero inoxidable con la lana de vidrio en la parte inferior donde se encuentran los disipadores, se coloca pasta disipadora para tener una buena disipación al momento de enfriar, la temperatura con la absorción de humedad 19°C y 59°C es el rango de humedad relativa que tiene la fibra de vidrio.



Figura 40. Ensamble 8.

La pasta disipadora se llama Dow Cornin 340 diseñada para proporcionar a los elementos de potencia alta como las celdas peltier TEC1-12706 está hace un puente térmico para su disipación eficaz porque tiene baja exudación, su evaporación es de 200°C y el rango de cambio en consistencia resiste 1770°C haciendo estable las temperaturas.



Figura 41. Ensamble 9.

Para aplicar la pasta disipadora de manera correcta se considerar verter en cantidades pequeñas, hasta que se vea una capa que cubra las celdas peltier dentro de su contorno, estas son estándar donde trabajan con 6 Amperios máximo y soportan un voltaje máximo de 15 V DC y su consumo es de 60 W.



Figura 42. Ensamble 10.

Al colocar la pasta disipadora por ambos lados se procede a colocar las celdas peltier de manera cuidadosa encima del envase donde está ubicado los disipadores, el lado que va enfriar debe ir a lado de los disipadores pequeños que están pegado dentro del reservorio y la parte de la cara a calentar, esta va ir adherida con el disipador más grande, se coloca un cooler encima este permite enfriar la parte caliente del disipador que esta adherida a las celdas peltier para darle más estabilidad.



Figura 43. Ensamble 11.

Se procede el ensamble del disipador de la parte de calor.

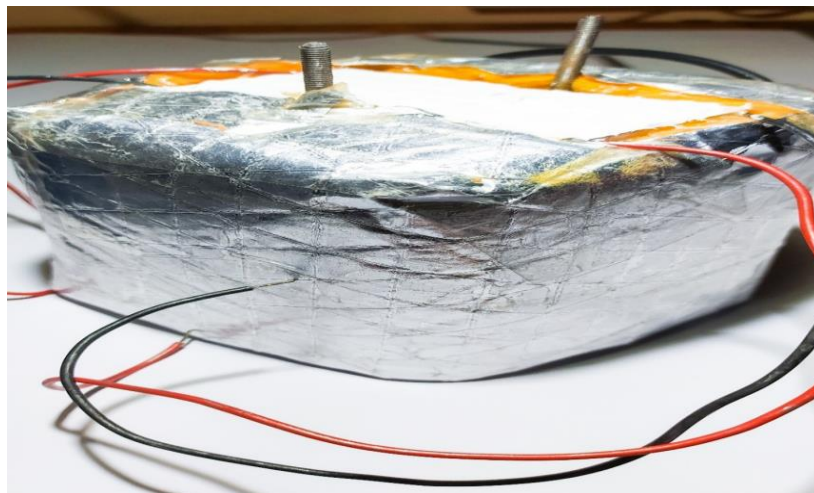


Figura 44. Ensamble 12.

Este disipador más grande permite enfriar la parte que calienta las celdas peltier de manera eficaz.

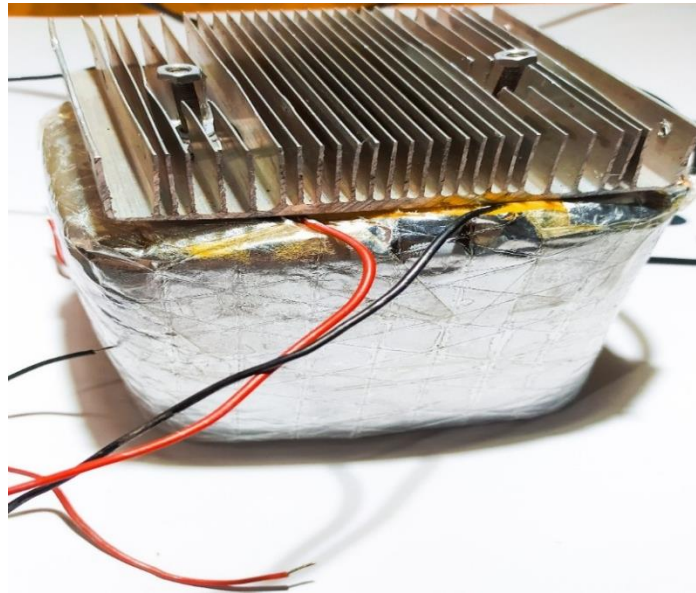


Figura 45. Ensamble 13.

Entre menos caliente este el lado que calienta las celdas peltier, va a permitir enfriar de mejor manera las dos caras para tener los mejores resultados de análisis.

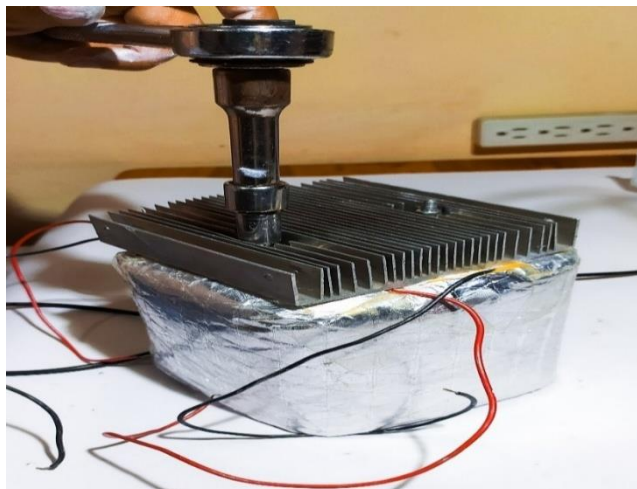


Figura 46. Ensamble 14.

Como refuerzo en la parte del disipador se debe añadir un cooler de 12V, su función va hacer enfría el disipador grande que calienta, a una temperatura considerable para proteger la durabilidad de las celdas peltier.



Figura 47. Ensamble 15.

El cooler va ir dentro de una carcasa de plástico, esta tiene como función sostener el cooler con el disipador para establecer una seguridad de ajuste y no se tienda a mover o salir del disipador.

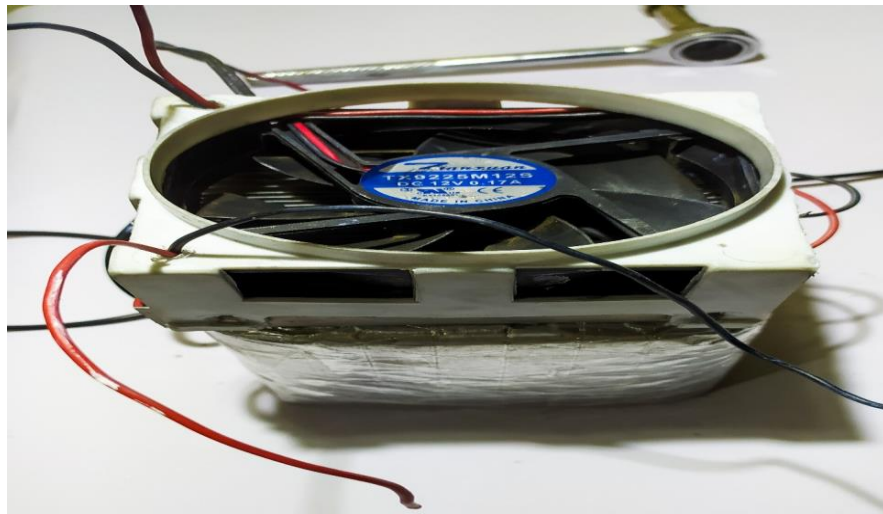


Figura 48. Ensamble 16.

Las celdas peltier con su respectivo disipador, se logra hacer la primera prueba, para revisar que los disipadores que estén dentro del reservorio enfríe de manera correcta.

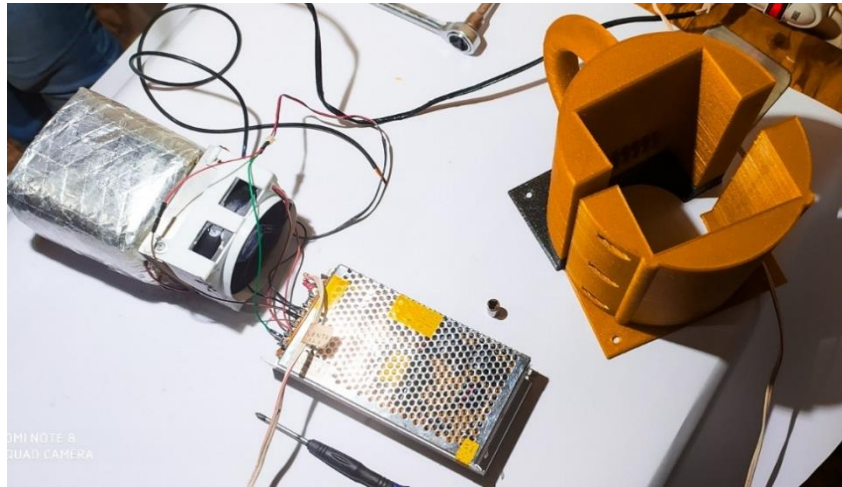


Figura 49. Ensamble 17.

Se ingresa el reservorio a la estructura realizada en 3D, esta estructura hecha de un material llamado PLA Shiny es resistente a la humedad al igual que la grasa, su inflamabilidad es baja siendo más resistente a otro tipo de filamento en 3D y tiene una temperatura de impresión de 190°C-200°C.

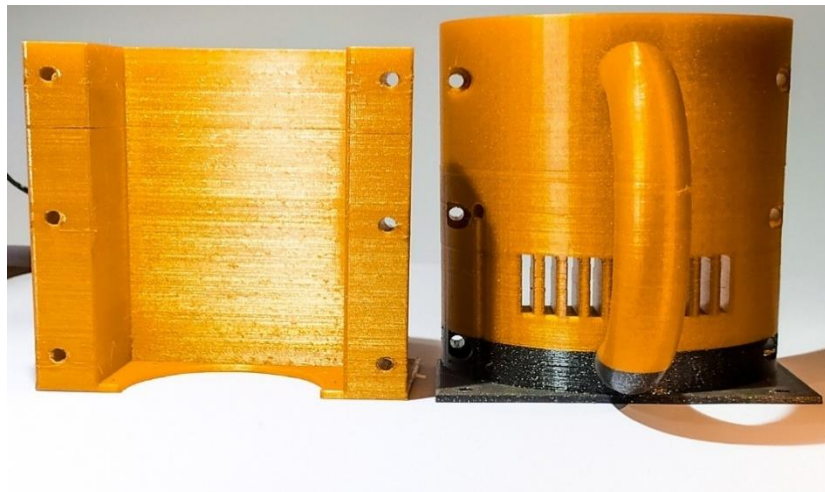


Figura 50. Ensamble 18.

Las partes de diseño en 3D se unen para hacer un solo ensamble, estas dos partes se unen con pernos para hacer más seguro el envase al momento de manipularlo.

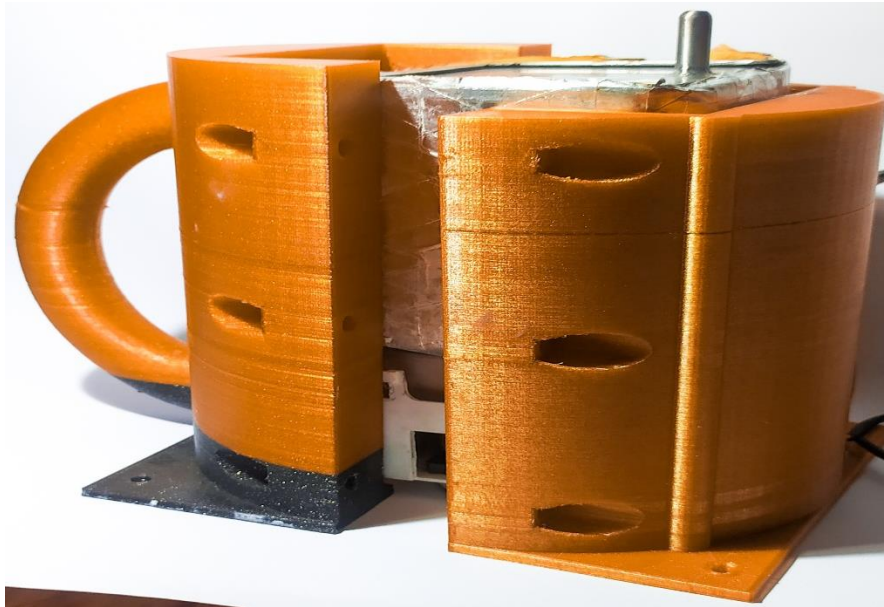


Figura 51. Ensamble 19.

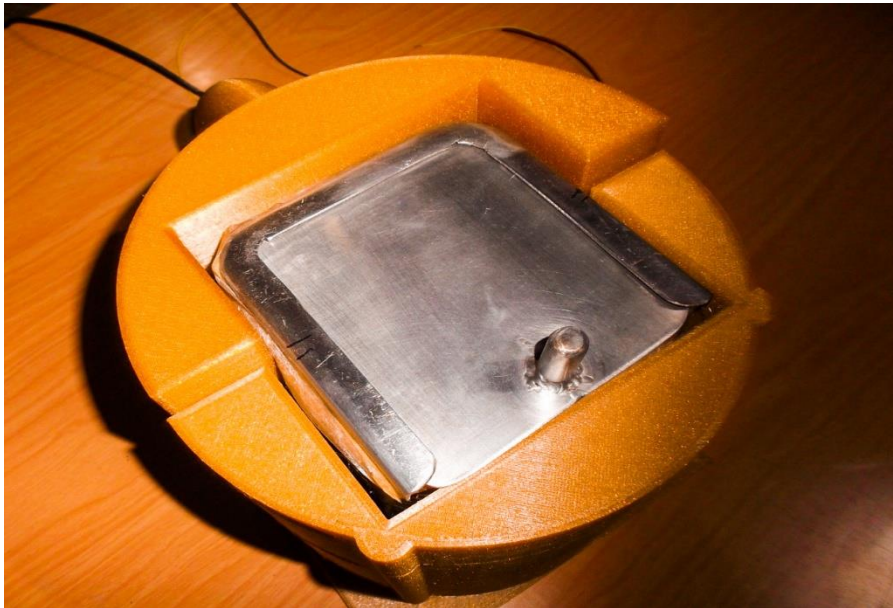


Figura 52. Ensamble 20.

La primera parte del ensamble es muy importante, porque esta nos permite analizar los datos de la planta para ver como se comporta el PID al momento de un cambio de temperatura hecho por el calentador eléctrico.

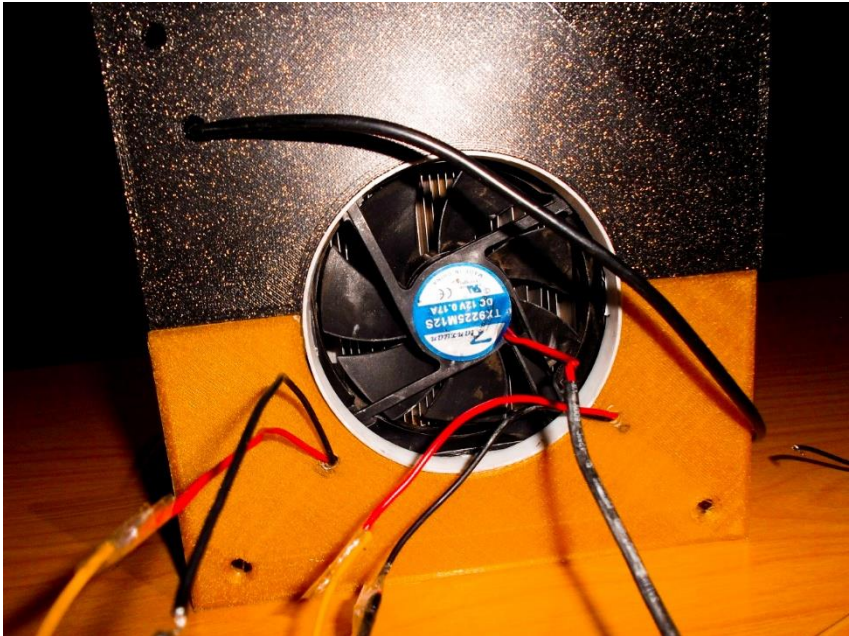


Figura 53. Ensamble 21.

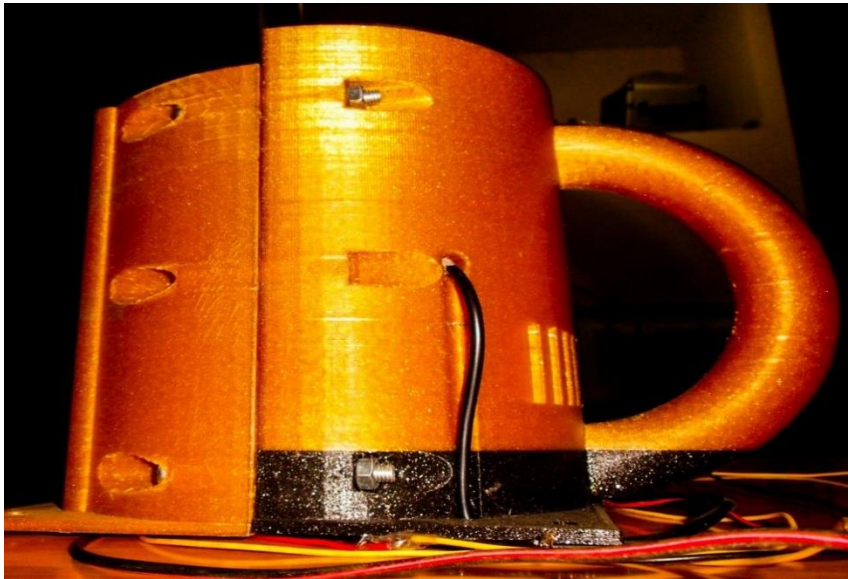


Figura 54. Ensamble 22.

4.2 Segunda Parte De Ensamble Termo Portátil

El diseño realizado en 3D, para tener una mayor seguridad de ajuste de ensamble con la base del termo se procede a utilizar pernos largos para facilitar un mayor agarre al momento de moverlo.



Figura 55. Ensamble 23.

Se utilizan cuatro pernos, uno para cada esquina con sus respectivas tuercas para cerrarlo por completo.

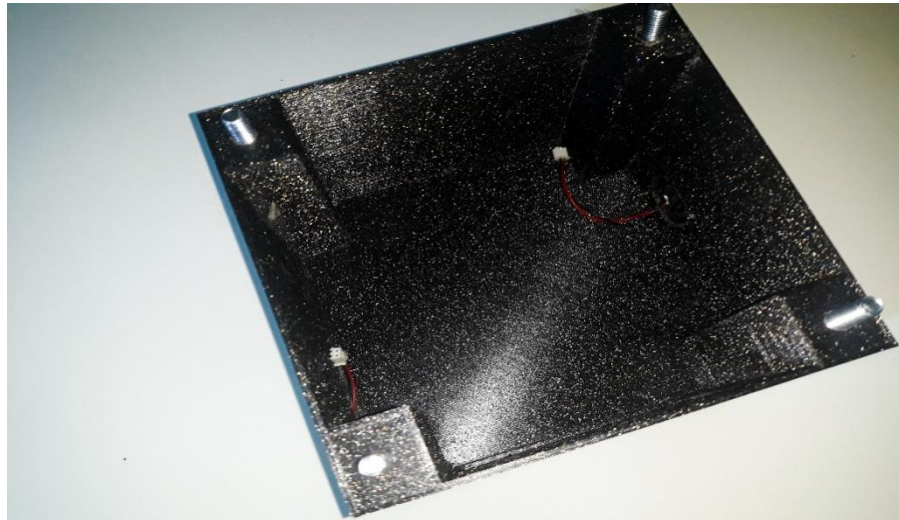


Figura 56. Ensamble 24.

El aire que produce la disipación del ventilador del termo donde se encuentra el primer ensamble, cae en las conexiones del segundo ensamble para evitar que se dañen los módulos por mucho calor o mala disipación.



Figura 57. Ensamble 25.

Se coloca dos extractores de aire el cual permite extraer el aire que cae y se mantiene dentro del segundo ensamble.

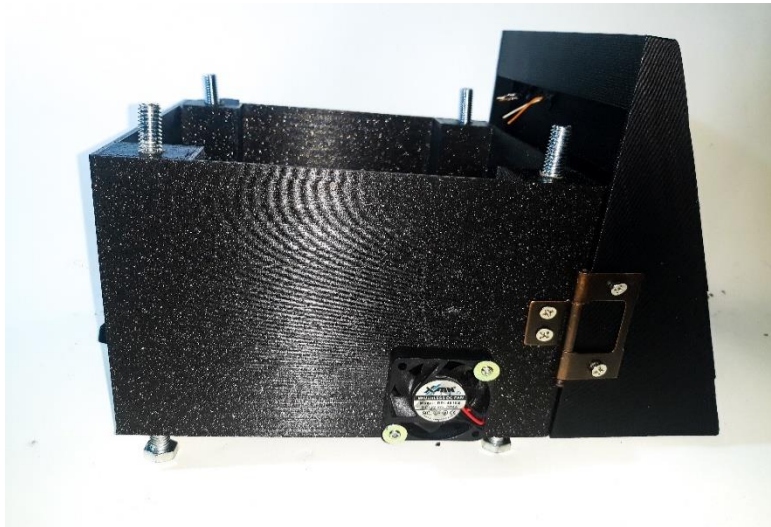


Figura 58. Ensamble 26.

En la parte de atrás se coloca los componentes de protección y alimentación del equipo donde:

- USB Mini-B que viene del Arduino nano a la computadora, para la adquisición de datos.
- Entrada de 12 V DC, su función es alimentar toda la planta.
- Llave de bloqueo, para seguridad al momento de encender el equipo.
- Fusible de protección de 30 A, para protección de sobre carga.
- Interruptor, su función es para elegir la manera manual o bluetooth.



Figura 59. Ensamble 27.

Se coloca una aldaba a la puerta en la parte lateral, esta nos ayuda abrir la parte frontal para poder visualizar las conexiones internas.



Figura 60. Ensamble 28.

La parte frontal se añade los siguientes componentes:

- Pantalla LCD 16x2, su función visualizar la temperatura y CP del Potenciómetro establecido.
- Potenciómetro de 10k, este permite manipular el CP deseado.
- Salida de 12 v para el calentador eléctrico.



Figura 61. Ensamble 29.

También consta de 4 diodos led donde cada uno cumple una función, el primero de color verde da el encendido del sistema, el segundo de color amarillo da la manera manual, el tercero de color rojo dará el modo bluetooth y el cuarto de color azul cuando el sistema este frío.



Figura 62. Ensamble 30.

Los componentes que se observan en la figura 63, son los que van a interactuar con la placa PBC principal, está hecha con anillos metalizados de doble cara, para garantizar las conexiones soldadas adheridas a ellas.

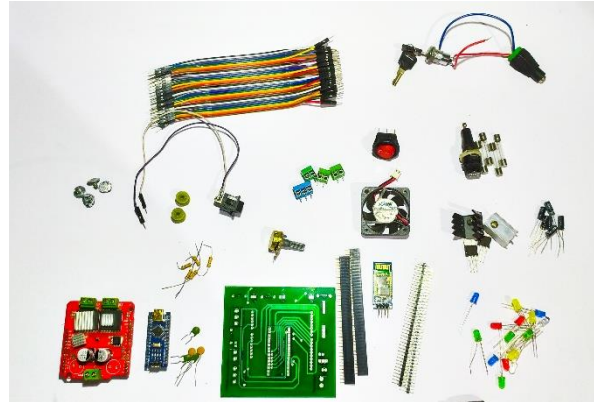


Figura 63. Ensamble 31.

La cara de arriba se ubica el módulo Arduino nano el cual contiene el PID programado y el módulo de bluetooth que interactúa con una aplicación para sistemas operativos Android, se añadió un regulador 7805 para no utilizar el voltaje de 5V DC del Arduino y causar alguna sobre carga, este voltaje regulado por el 7805 va a alimentar los componentes y módulos que trabajen con 5V DC.

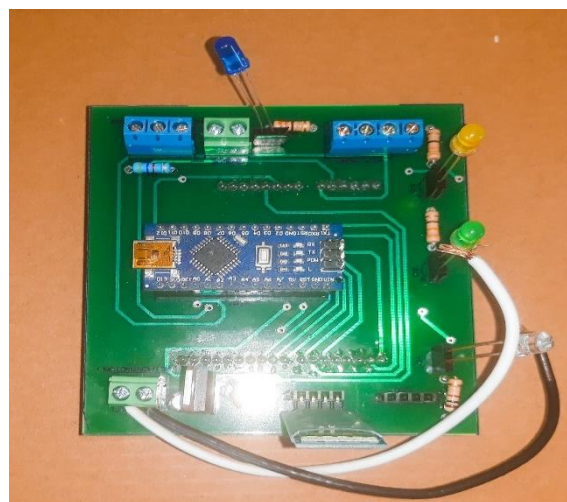
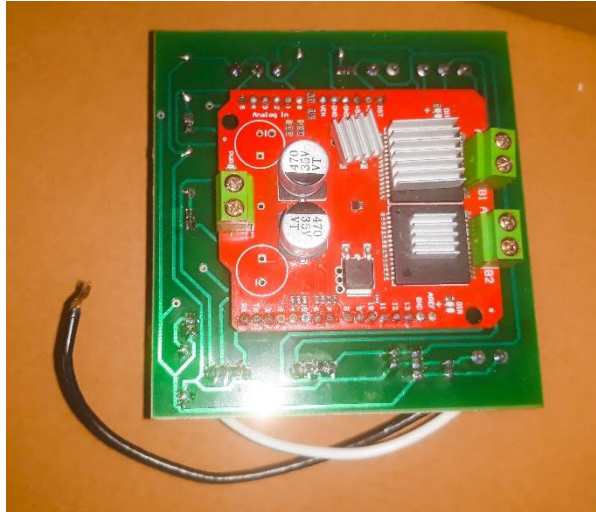


Figura 64. Ensamble 32.

La segunda cara de la placa diseñada en PBC reducen las conexiones para el módulo monster shield donde su función es amplificar la señal PWM generada por el Arduino y suministrar la corriente necesaria para que funcionen las celdas peltier TEC1-12706.



Antes de energizar el equipo se necesita verificar de manera segura las conexiones y módulos que estén insertado de manera correcta, para no causar un corto al momento de energizar todo el sistema.



Figura 66. Ensamble 34.

Las etiquetas para cables mangas termocontratiles son resistente a la abrasión y sustancias químicas, su tamaño es de 30x20 mm.



Figura 67. Ensamble 35.

Estas etiquetas permiten identificar los cables en caso de una desconexión de la tarjeta madre.



Figura 68. Ensamble 36.

5. ANÁLISIS DE RESULTADOS

5.1 Análisis Del Proyecto

Para lograr el proceso de análisis y tener una mejor planta primero se debe tener los cálculos matemáticos, se utiliza el PID para controlar en lazo cerrado esto nos va a permitir mejorar el tiempo de frío para estabilizar de manera rápida la planta.

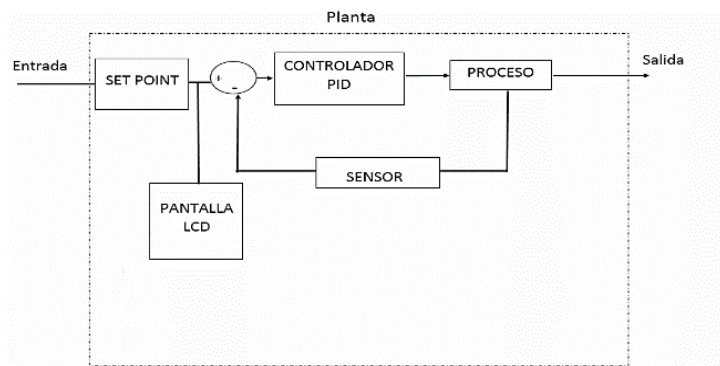


Figura 69. Planta

Los valores que se establecen el rango de la planta es de 23°C a 29°C, este rango se coloca después de analizar la planta en lazo abierto el análisis siempre se considera en los momentos iniciales donde se nota un gran cambio en la planta, donde se establece la primera temperatura de prueba en 28°C a 20°C esta se realiza porque yo quiero hacer que mi usuario solo espere 10 minutos y no una hora para el máximo tiempo de frío, al modificar la temperatura para establecer el rango se considera donde se optimiza mejor la planta con respecto al tiempo versus a la energía cuando este frío y las condiciones del material del envase.

En el PID no se va a ver el retardo ni la constante de tiempo o el tiempo de estabilización, eso se ve en la planta el PID es un controlador interno que está programado en Arduino donde tiene matemáticas para agilizar el proceso para enviar el error a 0 de una mejor forma.

Se toma en cuenta que el PID analiza el error en el momento que integra, deriva, y considera el tiempo de muestreo, el PID en control continuo se integra con respecto al tiempo, se deriva con respecto al tiempo, para crear la función transferencia de la planta en tiempo continuo, pero se discretiza en período de tiempo para tomar la muestra.

Una variable es la temperatura que desea el usuario menos la temperatura del sensor cuando se realiza esta resta, aquí aparece el error donde la $e(k)$ de manera discreta ingresa al PID para establecer los parámetros de control y sale a la planta, por eso es que se ingresa los parámetros como la ganancia, constante de tiempo, retardo y el tiempo de estabilización para que lo considere el PID.

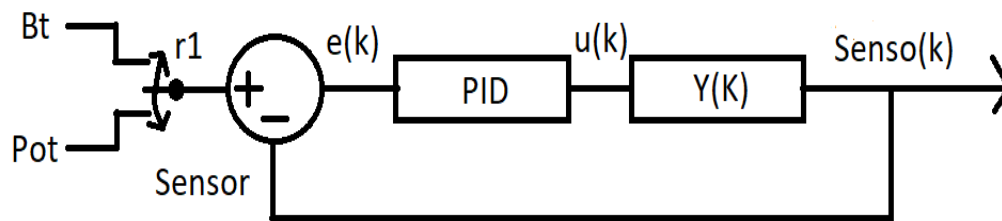


Figura 70. PID

Por eso se establece un tiempo de muestreo en base de lo que se requiera en el PID o al método que se utiliza, en este caso el método a usar es el de Ziegler – Nichol que

nos pide un período de muestreo donde se considera que sea de primer orden que se conozca la planta y que tenga la forma sigmoideal.

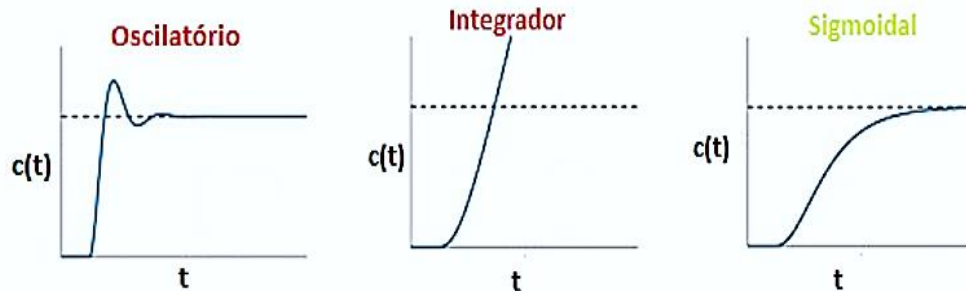


Figura 71. Gráfica Sigmoideal.

Este método también va a ayudar a encontrar los coeficientes faltantes para la función de transferencia obtenida matemáticamente haciendo una forma práctica experimental que analiza la gráfica, ya que en todas las plantas siempre surge este problema de manera matemática, por eso existen métodos que deben cumplir cierto requisitos para poder aplicarlos, al utilizar los coeficientes con el método práctico se analiza la planta en lazo abierto al plantear el método de Ziegler – Nichol de esta se obtiene los datos para la constante de tiempo τ y K que es la ganancia de proceso de manera discreta al analizar la planta, ya que si existiera todos los coeficientes de manera matemática en caso de una posible mejora, los cálculos matemáticos se deben acercar a los cálculos de manera práctica experimental obtenido con el método de Ziegler – Nichol.

Al utilizar un circuito digital no se puede analizar la planta en tiempo continuo porque un microcontrolador o un procesador consta de un reloj interno, que envía pulsos para que funcione y en algunos casos en cada segundo envía un proceso.

Ejemplo:

Se demora un ciclo por cada proceso, al analizar el diagrama desde tomar lectura en el SP, sensor, error y el PID entonces cada proceso que haga es un segundo, esto se hace en varios ciclos y no lo realiza todo a la vez.

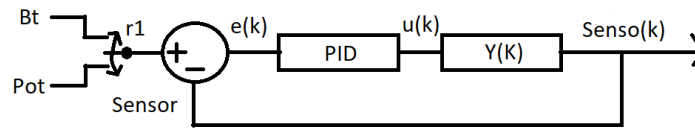


Figura 72. Lazo cerrado.

La ley de controlador PID discreto arroja valores entre 0 y 100 este valor se representa en 2.55 para que de el valor de la señal del PWM envía en los pines 5 y 6 del monster shield vnh2sp30 donde se manda el voltaje requerido para que funcione las celdas peltier, si se trabaja con varios valores de porcentaje de PWM se tendría de manera gráfica de como se comporta la señal PWM.

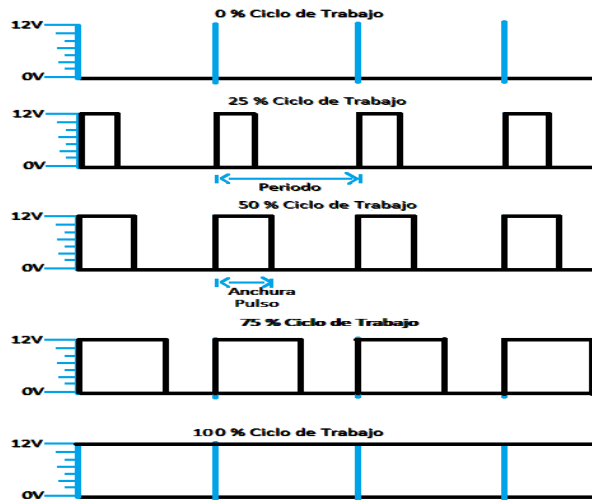


Figura 73.PWM

Formula	VELOCIDAD	Vout
$Y = (X \cdot VCC) / 255$	255	+VCC
X= Velocidad	X	Y
velocidad= $u \cdot 2,55$	x=255	Y=12v
y= Vout	x= 200	Y= 9,4V
	X=100	Y=4,71

TABLA 4. PWM cálculo 12v.

Voltaje Requerido 12v					
1)	255*12	2)	200*12	3)	100*12
	3060		2400		1200
	3060/255		2400/255		1200/255
Y=	12V	Y=	9,4V	Y=	4,71

TABLA 5. Voltaje requerido 12v.

El calor es un dato analógico que lo censa el sensor y hace digital la lectura, donde se pasa a analizar en tiempo discreto toda la planta para enviar una señal PWM al peltier que es un semiconductor analógico, por eso no envía una señal discreta si no una señal analógica y de eso se encarga el PWM, cuando el monster shield vnh2sp30 envía una señal de 0 a 100 se hace una gráfica de escalón unitario estando en lazo abierto, esto actúa como un switch y el PWM no varía en lazo cerrado dentro la programación del Arduino, esta va a depender de U donde se define como va a trabajar siendo esta una variable de control, se utiliza una señal PWM porque el Arduino necesita esa señal de salida analógica esta señal PWM no le hace nada a la planta lo que realiza es que la variable de control u que el PWM cumpla con la función de generar pulsos internos en el Arduino al monster shield vnh2sp30 y interprete que tiene que enviar el 100% a la variable de control para funcionar.

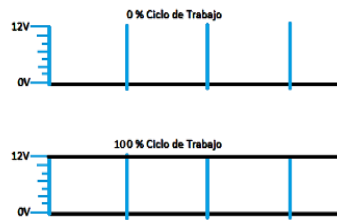


Figura 74. PWM ciclo de Trabajo.

5.2 Análisis Formula De La Segunda Ley De La Termodinámica Y Boltzmann.

$$\frac{\Delta Q}{\Delta t} = Q_{in} - Q_{out} \quad Q_{in} = \alpha Q_i \quad \text{Ec (12)}$$

T entrada (T. ambiente) 29°C.

T de salida convección y radiación.

Donde:

- **Q_i**= Calor de entrada.
- **α**= Factor relación 0 a 100%.

$$Q_{out} = \underbrace{KT(T - T_{\infty})}_{\text{Convección}} + \underbrace{\epsilon \sigma A(T^4 - T_{\infty}^4)}_{\text{Radiación}} \quad \text{Ec (13)}$$

Donde:

- **KT** : Constante de perdida por el ambiente.
- **T** : Temperatura del agua.
- **T_∞** : Temperatura ambiente.
- **ε** : Emisividad.
- **σ** : Constante de Stefan.
- **A** : Área de enfriamiento.

Variación:

$$\frac{dQ}{dt} = \alpha Q_i - K_T(T - T_\infty) - \varepsilon\sigma A(T^4 - T_\infty^4) \quad \text{Ec (14)}$$

$$K_T = UA \quad \text{Ec (15)}$$

- **U=** Coeficiente de transferencia de calor.
- **A=** Es el área.

$$\frac{dQ}{dt} = \alpha Q_i + UA(T_\infty - T) + \varepsilon\sigma A(T_\infty^4 - T^4) \quad \text{Ec (16)}$$

Se puede calcular el calor liberado o absorbido utiliza el calor específico Cp. Donde:

- Masa= **m**.
- Cambio de temperatura=**(T-Tref)**.
- Calor específico=**CP**.

$$Q = mC_p(T - T_{ref}) \quad \text{Ec (17)}$$

$$\frac{dmC_p(T - T_{ref})}{dt} = \alpha Q_i + UA(T_\infty - T) + \varepsilon\sigma A(T_\infty^4 - T^4) \quad \text{Ec (18)}$$

$$mC_p \frac{\partial T}{\partial t} - mC_p \frac{\partial T_{ref}}{\partial t} = \alpha Q_i + UA(T_\infty - T) + \varepsilon\sigma A(T_\infty^4 - T^4) \quad \text{Ec (19)}$$

$$\frac{\partial T_{ref}}{\partial t} = 0 ; \text{Puesto que la } T_{ref} \text{ es constante.} \quad \text{Ec (20)}$$

$$mC_p \frac{\partial T}{\partial t} = \alpha Q_i + UA(T_\infty - T) + \varepsilon\sigma A(T_\infty^4 - T^4) \quad \text{Ec (21)}$$

5.2.1 Cálculos Linealización Por Serie De Taylor

Primero hay que hallar el punto de Equilibrio:

$$\frac{\partial T}{\partial t} = \frac{\alpha Q_i}{mCp} + \frac{UA}{mCp}(T_\infty - T) + \frac{\varepsilon\sigma A}{mCp}(T_\infty^4 - T^4) \quad \text{Ec (22)}$$

El punto de Equilibrio (tiempo de estabilidad) $\frac{dT}{dt} = 0$, donde T_{SS} es la temperatura de estabilidad o estado estacionario.

$$0 = \alpha Q_i + UA(T_\infty - T_{SS}) + \varepsilon\sigma A(T_\infty^4 - T_{SS}^4) \quad \text{Ec (23)}$$

$$0 = \alpha Q_i + UAT_\infty - UAT_{SS} + \varepsilon\sigma AT_\infty^4 - \varepsilon\sigma AT_{SS}^4 \quad \text{Ec (24)}$$

Se agrupa términos semejantes:

$$0 = \alpha Q_i + UAT_\infty + \varepsilon\sigma AT_\infty^4 - UAT_{SS} - \varepsilon\sigma AT_{SS}^4 \quad \text{Ec (25)}$$

Recordemos que: $\alpha Q_i + UAT_\infty + \varepsilon\sigma AT_\infty^4$ es una constante y multiplicando por (-1), por lo tanto:

$$\varepsilon\sigma AT_{SS}^4 + (UA)T_{SS} - (\alpha Q_i + UAT_\infty + \varepsilon\sigma AT_\infty^4) = 0 \quad \text{Ec (26)}$$

Por consiguiente, queda una ecuación de cuarto orden si es posible tendrá raíces imaginarias y reales, en este caso T_{SS} deberá ser la raíz real y positiva.

5.2.2 Linealizando El Sistema Por Series De Taylor:

Se linealizará el sistema de acuerdo a la temperatura y el calor de entrada.

$$f(T, Qi) = \frac{\alpha}{mCp} Qi + \frac{UA}{mCp} (T_{\infty} - T) + \frac{\epsilon \delta A}{mCp} (T_{\infty}^4 - T^4) \quad \text{Ec (27)}$$

$$\frac{\partial T}{\partial t} = \frac{\partial f}{\partial T} \Big|_{T_{ss} Qi} (T - T_{ss}) + \frac{\partial f}{\partial Qi} \Big|_{T_{ss} Qi} (Qi - Q_{is}) + f(T_{ss}, Q_{is}) \quad \text{Ec (28)}$$

Se realiza las derivadas parciales con relación a la temperatura y al calor de entrada, evaluando todo en los puntos de equilibrio encontrados anteriormente se llega a la siguiente ecuación:

Parte 1:

$$\frac{\partial f}{\partial T} \Big|_{T_{ss} Qi} (T - T_{ss}) = \frac{\partial \left[\frac{\alpha}{mCp} Qi + \frac{UA}{mCp} (T_{\infty} - T) + \frac{\epsilon \delta A}{mCp} (T_{\infty}^4 - T^4) \right]}{\partial T} \Big|_{T_{ss} Qi} (T - T_{ss}) \quad \text{Ec (29)}$$

$$\frac{\partial f}{\partial T} \Big|_{T_{ss} Qi} (T - T_{ss}) = \frac{\partial \left[\frac{\alpha}{mCp} Qi + \frac{UA}{mCp} T_{\infty} - \frac{UA}{mCp} T + \frac{\epsilon \delta A}{mCp} T_{\infty}^4 - \frac{\epsilon \delta A}{mCp} T^4 \right]}{\partial T} \Big|_{T_{ss} Qi} (T - T_{ss}) \quad \text{Ec (30)}$$

$$\frac{\partial f}{\partial T} \Big|_{T_{ss} Qi} (T - T_{ss}) = \left(-\frac{UA}{mCp} - \frac{4\epsilon \delta A}{mCp} T_s^3 \right) \Delta T \quad \text{Ec (31)}$$

Parte 2:

$$\frac{\partial f}{\partial Qi} \Big|_{T_{ss} Qi} (Qi - Q_{is}) = \frac{\partial \left[\frac{\alpha}{mCp} Qi + \frac{UA}{mCp} (T_{\infty} - T) + \frac{\epsilon \delta A}{mCp} (T_{\infty}^4 - T^4) \right]}{\partial Qi} \Big|_{T_{ss} Qi} (Qi - Q_{is}) \quad \text{Ec (32)}$$

$$\frac{\partial f}{\partial Qi} \Big|_{T_{ss} Qi} (Qi - Q_{is}) = \frac{\alpha}{mCp} \Delta Q \quad \text{Ec (33)}$$

Uniendo:

$$\frac{dT}{dt} = \left(-\frac{UA}{mCp} - \frac{4\epsilon\sigma A}{mCp} T_{ss}^3 \right) \Delta T + \frac{\alpha}{mCp} \Delta Q + \frac{\partial T}{\partial t} |_{T_{ss} Q_{is}} \quad \text{Ec (34)}$$

$$\frac{\partial \Delta T}{\partial t} = \left(-\frac{UA}{mCp} - \frac{4\epsilon\sigma A}{mCp} T_{ss}^3 \right) \Delta T + \left(\frac{\alpha}{mCp} \right) \Delta Q \quad \text{Ec (35)}$$

Función de Transferencia

Se aplica la transformada de Laplace:

$$s[\Delta T(s)] = \left(-\frac{UA}{mCp} - \frac{4\epsilon\sigma A}{mCp} T_{ss}^3 \right) \Delta T(s) + \left(\frac{\alpha}{mCp} \right) \Delta Q(s) \quad \text{Ec (36)}$$

$$s\Delta T(s) + \left[\frac{UA}{mCp} + \frac{4\epsilon\sigma A}{mCp} T_{ss}^3 \right] \Delta T(s) = \left(\frac{\alpha}{mCp} \right) \Delta Q(s) \quad \text{Ec (37)}$$

$$\frac{\Delta T(s)}{\Delta Q(s)} = \frac{\left(\frac{\alpha}{mCp} \right)}{s + \left(\frac{UA}{mCp} + \frac{4\epsilon\sigma A}{mCp} T_{ss}^3 \right)} \quad \text{Ec (38)}$$

Dejando la funcion de transferencia de la forma:

$$\frac{\Delta T(s)}{\Delta Q(s)} = \frac{\frac{\alpha}{mCp}}{\frac{UA}{mCp} + \frac{4\epsilon\sigma A}{mCp} T_{ss}^3} \quad \text{Ec (39)}$$

$$\frac{\Delta T(s)}{\Delta Q(s)} = \frac{s}{\frac{UA}{mCp} + \frac{4\epsilon\sigma A}{mCp} T_{ss}^3 + \frac{UA}{mCp} + \frac{4\epsilon\sigma A}{mCp} T_{ss}^3}$$

$$\frac{\Delta T(s)}{\Delta Q(s)} = \frac{\frac{\frac{\alpha}{mC_p}}{\frac{UA}{mC_p} + \frac{4\epsilon\sigma A T_{ss}^3}{mC_p}}}{\frac{s}{\frac{UA}{mC_p} + \frac{4\epsilon\sigma A T_{ss}^3}{mC_p}} + 1} \quad \text{Ec (40)}$$

Reemplazando:

$$\frac{\frac{\alpha}{mC_p}}{\frac{UA}{mC_p} + \frac{4\epsilon\sigma A T_{ss}^3}{mC_p}} = K \quad \text{Ec (41)}$$

$$\frac{1}{\frac{UA}{mC_p} + \frac{4\epsilon\sigma A T_{ss}^3}{mC_p}} = \tau \quad \text{Ec (42)}$$

Obtenemos:

$$\frac{\Delta T(s)}{\Delta Q(s)} = \frac{K}{\tau s + 1} \quad \text{Ec (43)}$$

En los sistemas térmicos la temperatura no aumenta instantáneamente, demora un tiempo y ese tiempo se conoce como retardo.

$$G(s) \cdot e^{-Ls} \text{ retardo se puede tomar} \quad \text{Ec (44)}$$

5.3 Final módulo general.

$$G(s) = \frac{k}{\tau s + 1} \cdot e^{-Ls} \quad \text{Ec (45)}$$

$$G(s) = \frac{-0.6}{175s+1} \cdot e^{-48s} \quad \text{Ec (46)}$$

Donde L es el tiempo de retardo que se obtendrá de forma experimental en la gráfica de la función de transferencia.

En fines prácticos para τ es la constante de tiempo que se puede tomar encontrando el tiempo de estabilidad t_{ss} y dividirlo para 4 de esta forma lo obtenemos experimentalmente.

5.4 Análisis Método De Ziegler-Nichols

$$\tau = \frac{t_{ss}}{4} = \frac{700}{4} = 175 \quad \text{Ec (47)}$$

Donde K es la ganancia del proceso, los 23°C corresponde a la temperatura final y el 29°C a la temperatura inicial y el 100% que corresponde al escalón unitario de mi señal PWM al máximo, esta es u_{final} mientras $u_{inicial}$ mi temperatura es 0.

$$K = \frac{T_{final} - T_{inicial}}{U_{final} - U_{inicial}} = \frac{23 - 29}{100} = -0.6C/\% \quad \text{Ec (48)}$$

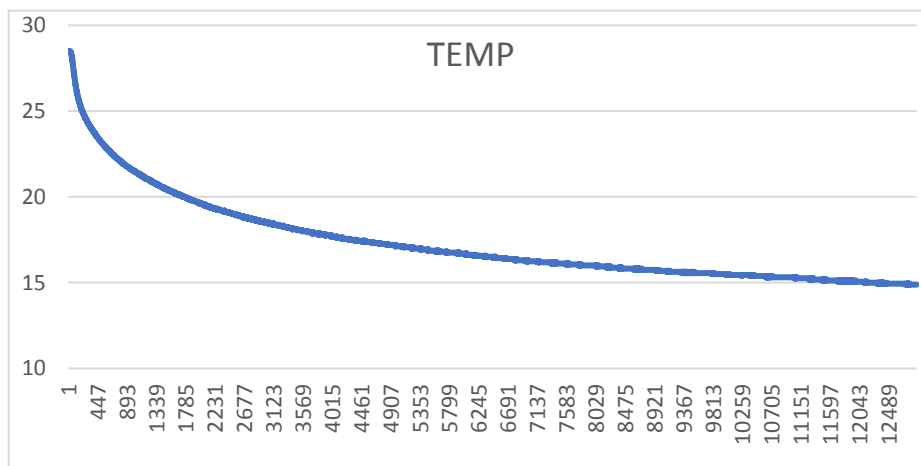


Figura 75. Análisis Planta lazo abierto.

El PID $u(t)$ tiene la ganancia proporcional, el tiempo integral y derivado, donde $e(k)$ es el error y la diferencia se encuentra entre el set point que manipula el usuario menos el valor dado por el sensor.

$$e(k) = r(k) - y(k) \quad \text{Ec (49)}$$

$$u(t) = K_p e(t) + \frac{K_p}{t_i} \int_0^t e(t) dt + K_p t_d \frac{de(t)}{dt} \quad \text{Ec (50)}$$

En tiempo discreto t_s es el tiempo de muestra donde $e(k)$ es el error presente.

$$\int_0^t e(t) dt \rightarrow \sum \left[\frac{e(k)+e(k-1)}{2} \right] T_s \quad \text{Ec (51)}$$

$$\frac{de(t)}{dt} \rightarrow \frac{e(k)+e(k-1)}{T_s} \quad \text{Ec (52)}$$

El período de muestreo T_s , se utiliza uno de los dos métodos que fueron propuestos por Ziegler-Nichols:

Donde T_s es el theta θ que corresponde al retardo donde se obtiene del lazo abierto de la gráfica y t es la constante de tiempo.

$$T_s < \frac{\theta}{4} = \frac{48}{4} = 12 \quad \text{Ec (53)}$$

$$\frac{t}{20} \leq T_s \leq \frac{t}{10} \quad \text{Ec (54)}$$

Porque t es la constante de tiempo donde tiene un valor de 175 y se reemplaza para el segundo método de Ziegler- Nichols.

$$\frac{t}{20} = \frac{175}{20} = 8,75 \quad \text{Ec (55)}$$

$T_s = 8,75$ es nuestro período de muestreo.

$$\frac{t}{10} = \frac{175}{10} = 17,5 \quad \text{Ec (56)}$$

$T_s = 17,5$ es nuestro período de muestreo.

$$8,75 \leq T_s \leq 17,5 \quad \text{Ec (57)}$$

El valor utilizado en el código de Arduino es de 9 que redondea el valor de 8,75 y también se utiliza el 12 del primer método para hacer las primera pruebas. Ver anexo G

$$\theta = L + T_s/2 \quad \text{Ec (58)}$$

Se utiliza el retardo $T_s/2$ para el control discreto PID de esta manera se obtiene la función de transferencia:

Se discretiza en período de tiempo para tomar la muestra:

$$C(z^{-1}) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \quad \text{Ec (59)}$$

$$q_0 = kp \left[1 + \frac{T_s}{2ti} + \frac{td}{T_s} \right] \quad \text{Ec (60)}$$

$$q_0 = -kp \left[1 - \frac{T_s}{2ti} + \frac{2td}{T_s} \right] \quad \text{Ec (61)}$$

$$q_2 = \frac{kptd}{T_s} \quad \text{Ec (62)}$$

$$u(k)(1-z^{-1}) = q_0e(k) + q_1z^{-1}e(k) + q_2z^{-2}e(k) \quad \text{Ec (63)}$$

$$u(k) - u(k)z^{-1} = q_0e(k) + q_1z^{-1}e(k) + q_2z^{-2}e(k) \quad \text{Ec (64)}$$

$$u(k) = u(k)z^{-1} + q_0e(k) + q_1z^{-1}e(k) + q_2z^{-2}e(k) \quad \text{Ec (65)}$$

La transformada inversa Z es:

$$u(k) = u(k-1) + q_0e(k) + q_1e(k-1) + q_2e(k-2) \quad \text{Ec (66)}$$

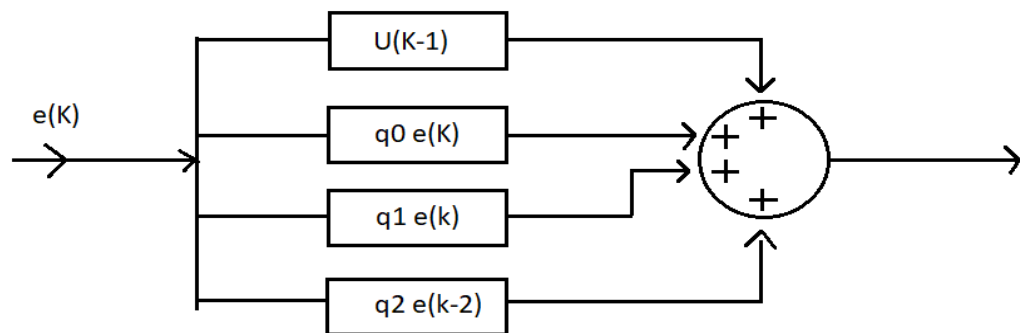


Figura 76. La transformada inversa Z.

Para la sintonía por el método de Ziegler y Nicols se utiliza el PID:

Controlador	k_p	T_i	t_d
P	$\frac{t}{k\theta}$	-	-
PI	$\frac{0.9t}{k\theta}$	3.33θ	-
PID	$\frac{1.2t}{k\theta}$	2θ	0.5θ

TABLA 6. Sintonía por el método de Ziegler y Nicols.

Los parámetros de Ziegler y Nicols y la transformada z del PID que se está ingresando en una sola ecuación la que sería la variable de control ingresando los datos y valores tomados que se justifican con el método de Ziegler – Nichol este dice que se puede usar un PID en tiempo discreto y se debe modificar el theta=48, esto es igual al tiempo muerto o retardo prima que se considera para el PID discreto.

Esta afirmación del método 1 es para control analógico y no para un control digital. Por consiguiente, el factor (L/S) es de 0.3 y 0.1 sin retardo dominante, una elección es incrementar el retardo siendo esta la mitad del período de muestreo $L'=L+T_s/2$.

$$L' = 48 + \frac{9}{2} \quad \text{Ec (67)}$$

5.5 Análisis De La Planta

La variación del tiempo con la variación de la temperatura al momento de encender el Arduino nano se debe de tomar en cuenta el tiempo de inicio del monster shield vnh2sp30 su temperatura inicial y el tiempo inicial.

- **t inicial**= Tiempo inicial = 27 segundos.
- **T inicial**= temperatura inicial =28,5°C.


	A	B	C	D	E	F
35		Monster prendido				
36		27	28,5	-18,5	100	
37		28	28,5	-18,5	100	
38		29	28,5	-18,5	100	
39		29	28,5	-18,5	100	
40		30	28,5	-18,5	100	
41		31	28,5	-18,5	100	
42		32	28,44	-18,44	100	
43		33	28,5	-18,5	100	
44		33	28,5	-18,5	100	
45		34	28,5	-18,5	100	

Figura 77. Análisis tiempo de variación 1.

Para calcular la variación de temperatura en los primeros 7 minutos se toma en cuenta:

- **1 minuto**= 60 segundos.
- **Tiempo**= 7min*(60 segundo) = 420 segundos.

Este valor se reemplaza para calcular el tiempo final, donde:

- **tfinal**= inicial+ Tiempo.
- **tfinal**= 27 segundo + 420 segundo.
- **tfinal**= 447 segundos.

El tiempo final es de 447 segundos, estos datos son generados por el Arduino nano por el monitor serie en tiempo real de la planta en lazo abierto para graficar como se comporta y se utiliza un documento creado en Excel.

Tomando en cuenta que el monitor serie nos da 4 valores de lectura, la primera corresponde al tiempo en segundo, el segundo valor corresponde a la temperatura ambiente que nos da el sensor, el tercero corresponde al error y el ultimo es la velocidad de PWM que es el 100%.

Donde:

- **Tfinal**= 447 segundos.
- **Tfinal** = 23°C.


	A	B	C	D	E	F
574		446	23,06	-13,06	100	
575		447	23	-13	100	
576		447	23	-13	100	
577		448	23	-13	100	
578		449	23	-13	100	
579		450	23	-13	100	
580		450	23	-13	100	
581		451	23	-13	100	
582		452	23	-13	100	
583		453	23	-13	100	

Figura 78. Análisis tiempo de variación 1.

Se concluye que en 7 minutos hay una variación de temperatura donde se reemplaza:

- **tvariación** = tfinal – tinicial
- **tvariación** = 28,5°C- 23°C
- **tvariación**= 5,5°C

El análisis dos con otro tiempo se toma el valor de tiempo que es de 7000 segundos dentro de 7 minutos.

- **tinicial**= 7000 segundos.
- **t**= 7 minutos.


	A	B	C	D	E	F
8990		7000	15,75	-5,75	100	
8991		7000	15,69	-5,69	100	
8992		7001	15,75	-5,75	100	
8993		7002	15,69	-5,69	100	
8994		7003	15,75	-5,75	100	
8995		7004	15,69	-5,69	100	
8996		7004	15,75	-5,75	100	
8997		7005	15,75	-5,75	100	
8998		7006	15,75	-5,75	100	

Figura 79. Análisis tiempo de variación 2.

El 7000 segundo se considera es la temperatura de 15,69°C.


	A	B	C	D	E	F
8985		6996	15,75	-5,75	100	
8986		6997	15,75	-5,75	100	
8987		6997	15,75	-5,75	100	
8988		6998	15,75	-5,75	100	
8989		6999	15,75	-5,75	100	
8990		7000	15,75	-5,75	100	
8991		7000	15,69	-5,69	100	
8992		7001	15,75	-5,75	100	
8993		7002	15,69	-5,69	100	
8994		7003	15,75	-5,75	100	
8995		7004	15,69	-5,69	100	
8996		7004	15,75	-5,75	100	

Figura 80. Análisis tiempo de variación 3.

- **1 minuto**= 60 segundos.
- **Tiempo**= 7min*(60 segundo) = 420 segundos.

Los dos datos iniciales es para sacar el tiempo final.

- **t_{final}**= 7000 + 420= 7420 segundos

En Excel se busca el tiempo de 7420 y la temperatura en dicho rango es de 15,56°C

- **t**= 7000 segundo
- **T**= 15,56°C

Al analizar de cuanta es la variación de temperatura que hay en un tiempo de 7 minutos será:

- **t_{variación}** = t_{final} – t_{inicial}
- **t_{variación}** = 15,69°C- 15,56°C
- **t_{variación}** = 0,13°C

En 7 minutos se analiza que hay una variación de 0,13°C por lo que se concluye que para fines prácticos la planta está estabilizada.

El análisis en 16°C en 7 minutos:

- **t inicial**=5000
- t=16.5°C

- **t_{final}**= 5420
- t=16.31°C

Donde la variación va hacer:

- **t_{variación}** = t_{final} – t_{inicial}
- **t_{variación}** = 16,31°C- 16,5°C
- **t_{variación}** = 0,19°C

5.6 Matlab

Se utiliza Matlab para graficar la función de transferencia en tiempo continuo y convertirla a tiempo discreto esta función es obtenida por los cálculos matemáticos.

Donde ingresamos datos:

- `% función tiempo continuo a discreta`
- `% Gráfica tiempo continuo`
- `%Michael velasquez & joselyn carvajal`

- `H = tf(-0.6,[175 1], 'InputDelay', 48)`
- `figure(1)`
- `step(H)`
- `title('Funcion Trasferencia Continua')`
- `grid on`

- `Hz = c2d(H,8)`
- `figure(2)`
- `step(Hz)`
- `title('Funcion Transferencia Discreto T=8')`
- `grid on`

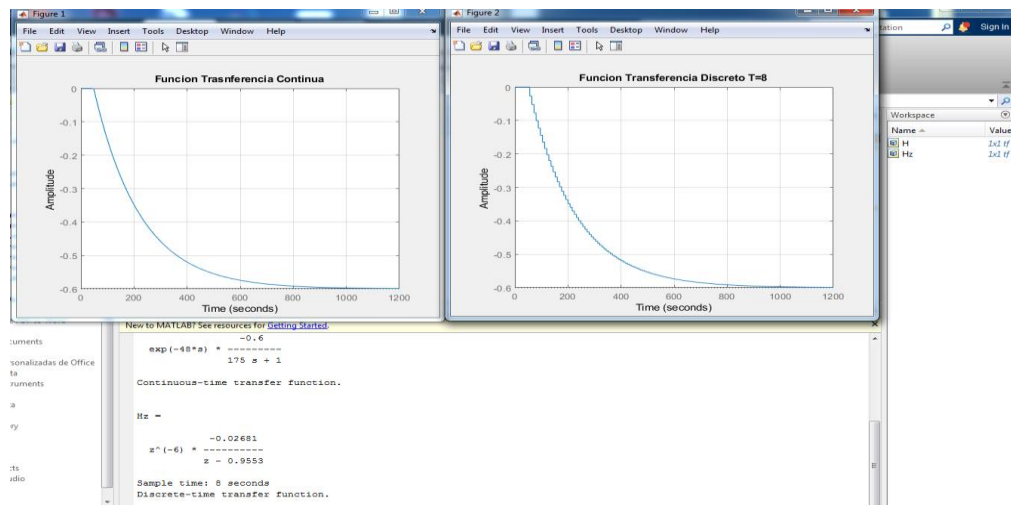


Figura 81. Matlab Gráfica ft en tiempo continuo.

La función de transferencia en discreto es:

$$z = z^{-6} \chi \frac{-0.02681}{z-0.9553} \quad \text{Ec (67)}$$

Una función discreta toma valores enteros donde su variable es con respecto al tiempo, la gráfica a continuación representa las muestras de la función de transferencia en tiempo discreto de la planta, donde:

```
clear all
close all
clc
t=(0:0.001:0.05);
k= -0.02681;
%q=175; % ciclo por muestra
%f=20; %muestras por ciclos
L= -6;
Ts= 0.9553;
theta= L +Ts/2;
z=1;
y= z^L;
r= ((k)/(Ts))*(y); %señal discreta
H= stem( r,'b','LineWidth' ,1)
j= stem( t,'b','LineWidth' ,1)
grid on
```

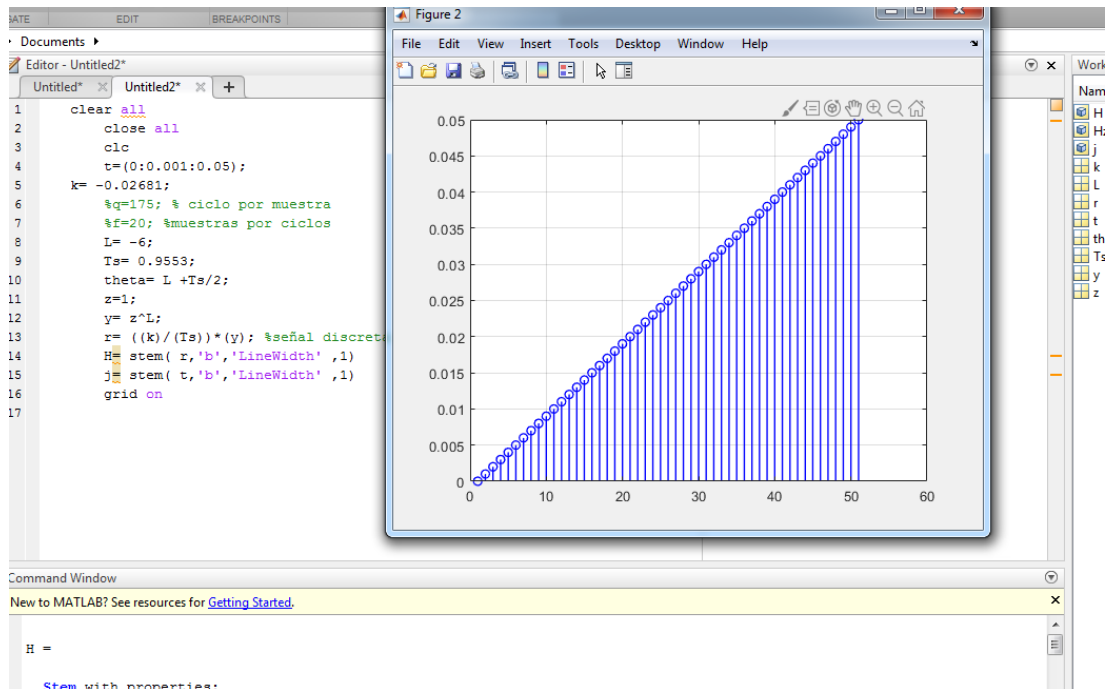


Figura 82. Matlab Gráfica en tiempo discreto.

La siguiente Configuración corresponde a la función de transferencia en tiempo continuo, se utiliza simulink para ver su comportamiento estableciendo esto es un análisis con fines práctico.

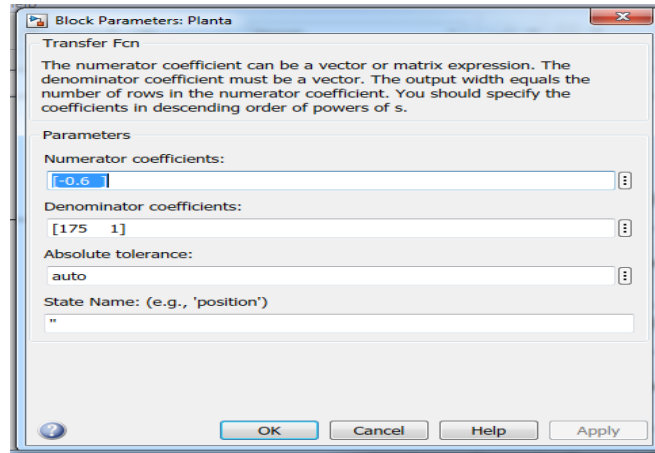


Figura 83. Simulink Gráfica ft en tiempo continuo.

La simulación de planta en simulink es la misma creada en el código de Matlab, pero sin retardo.

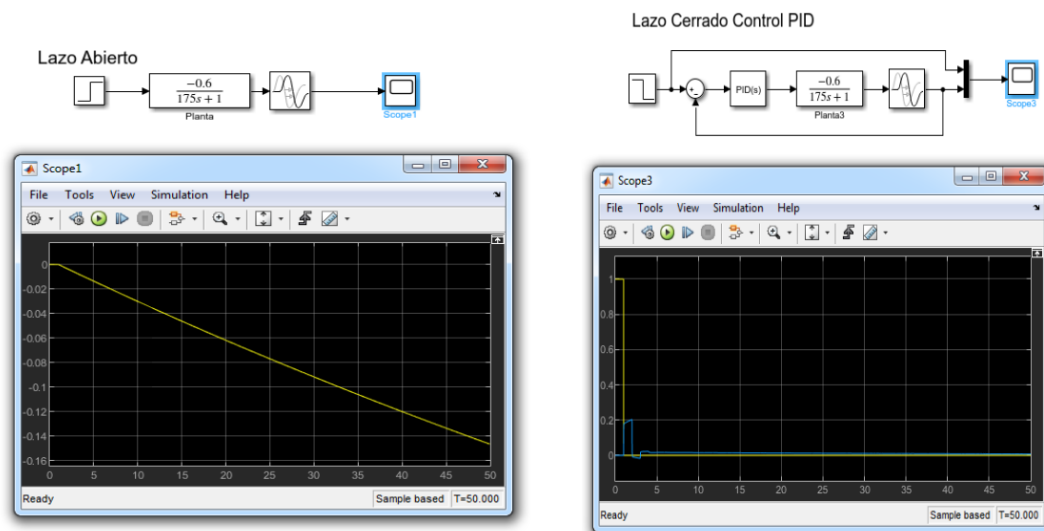


Figura 84. Simulink Gráfica ft en tiempo continuo.

Se selecciona las delimitadas donde se inserta clic en siguiente.

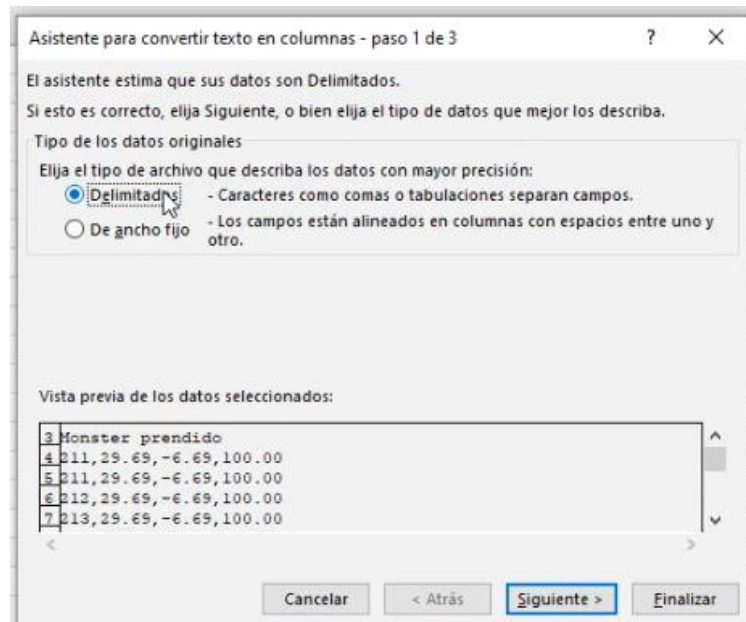


Figura 87. Configuración Excel parte 3.

Clic en coma, clic en siguiente.

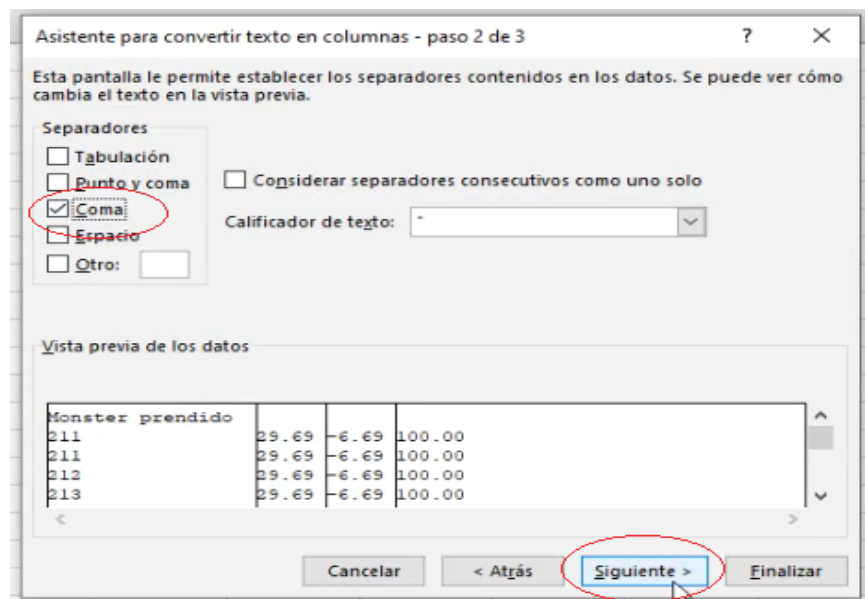


Figura 88. Configuración Excel parte 4.

Clic en general, clic en finalizar.

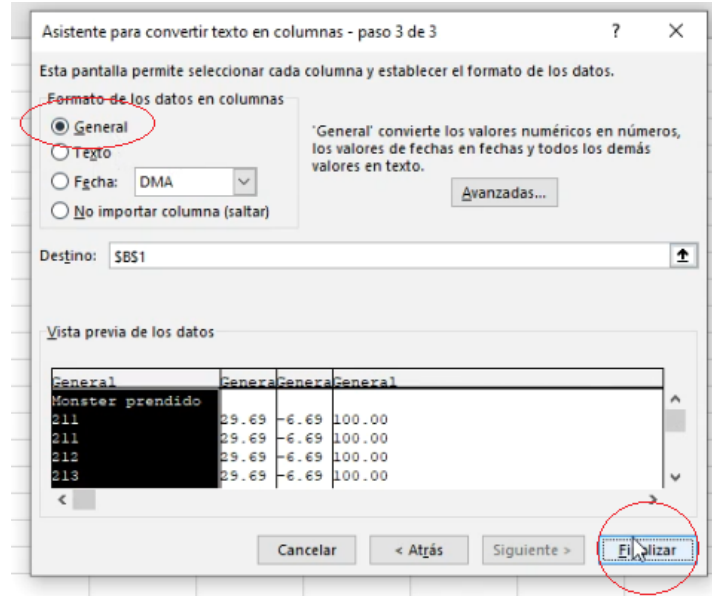


Figura 89. Configuración Excel parte 5.

Una vez separados los datos se reemplazan los puntos por las comas donde se selecciona las columnas y se presiona la tecla Ctrl + B, para poder reemplazar.

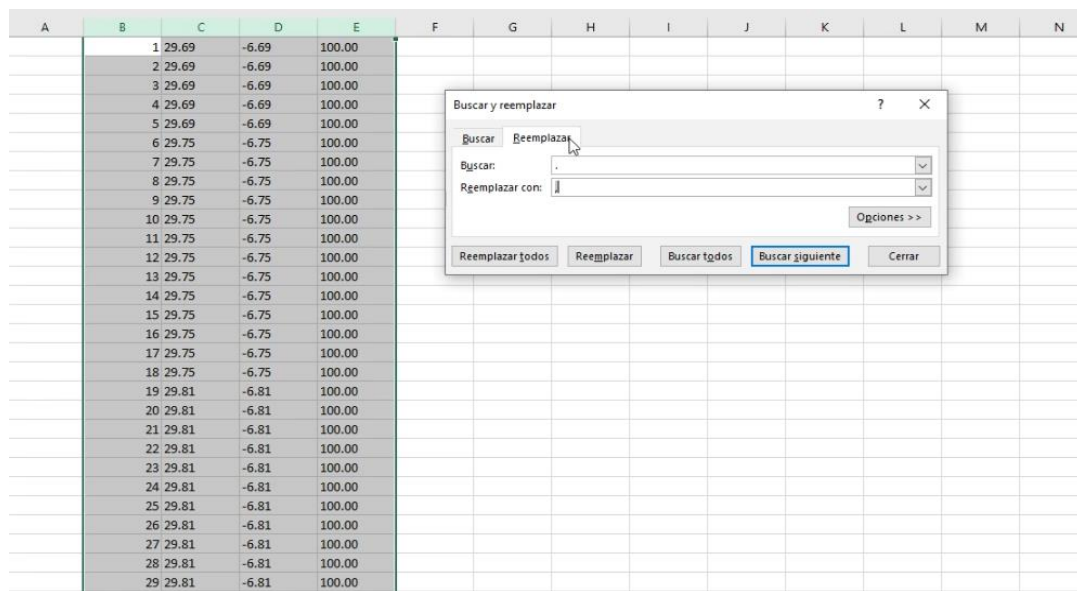


Figura 90. Configuración Excel parte 6.

Se reemplaza todos los valores.

	A	B	C	D	E	F	G	H
1			29,69	-6,69	100			
2			29,69	-6,69	100			
3			29,69	-6,69	100			
4			29,69	-6,69	100			
5			29,69	-6,69	100			
6			29,75	-6,75	100			
7			29,75	-6,75	100			
8			29,75	-6,75	100			
9			29,75	-6,75	100			
10			29,75	-6,75	100			
11			29,75	-6,75	100			
12			29,75	-6,75	100			
13			29,75	-6,75	100			
14			29,75	-6,75	100			
15			29,75	-6,75	100			
16			29,75	-6,75	100			
17			29,75	-6,75	100			
18			29,75	-6,75	100			
19			29,81	-6,81	100			
20			29,81	-6,81	100			
21			29,81	-6,81	100			
22			29,81	-6,81	100			

Figura 91. Configuración Excel parte 7.

Para graficar se selecciona las 4 columnas se dará:

- Clic en gráfica línea.
- Clic en insertar.

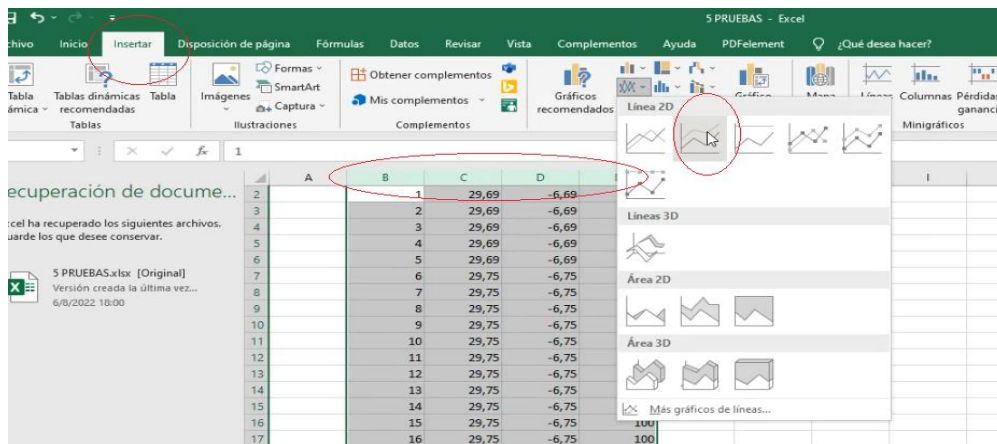


Figura 92. Configuración Excel parte 8.

Ahora se ajusta, clic en seleccionar datos.

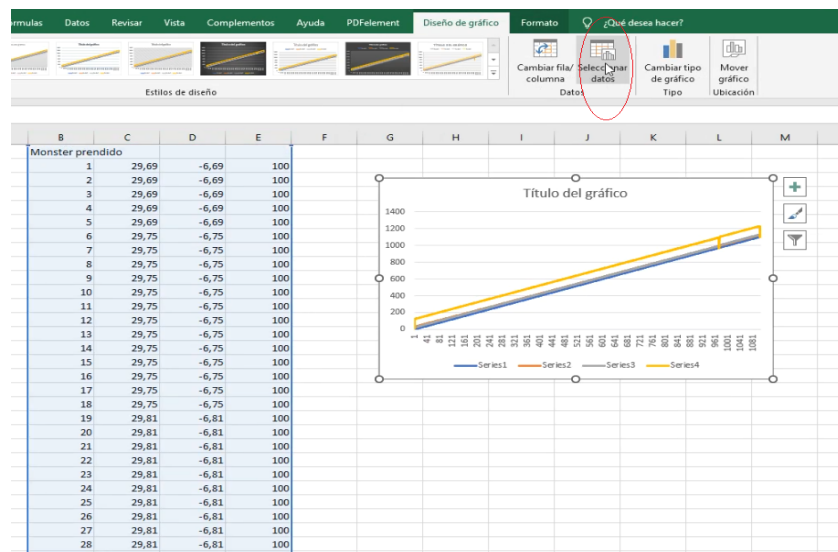


Figura 93. Configuración Excel parte 9.

- clic agregar.

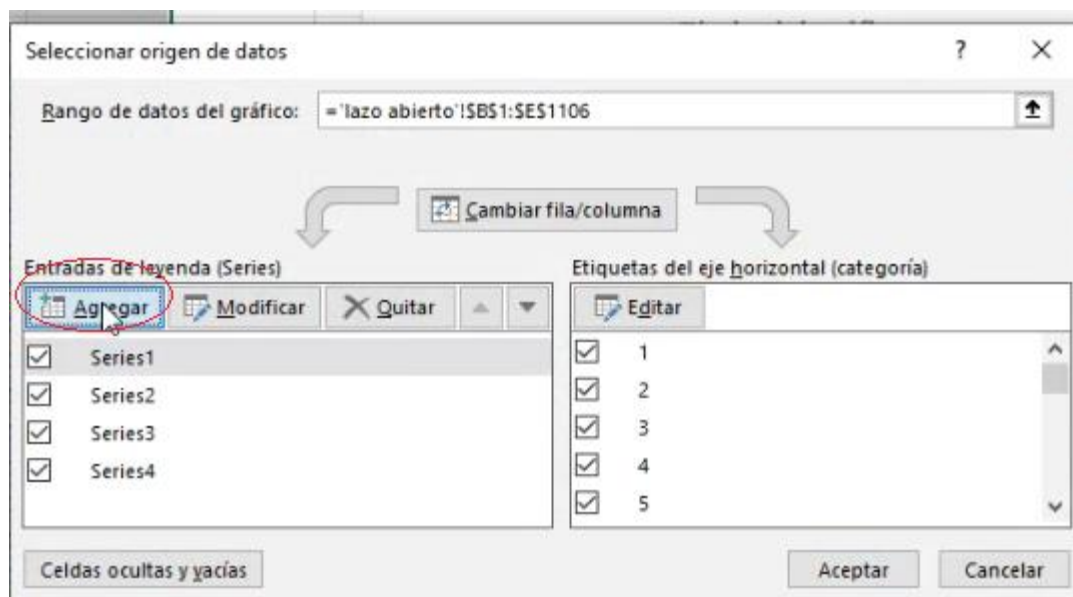


Figura 94. Configuración Excel parte 10.

Se escribe el nombre de la serie nueva y se selecciona el primer valor de la serie referente a la temperatura.

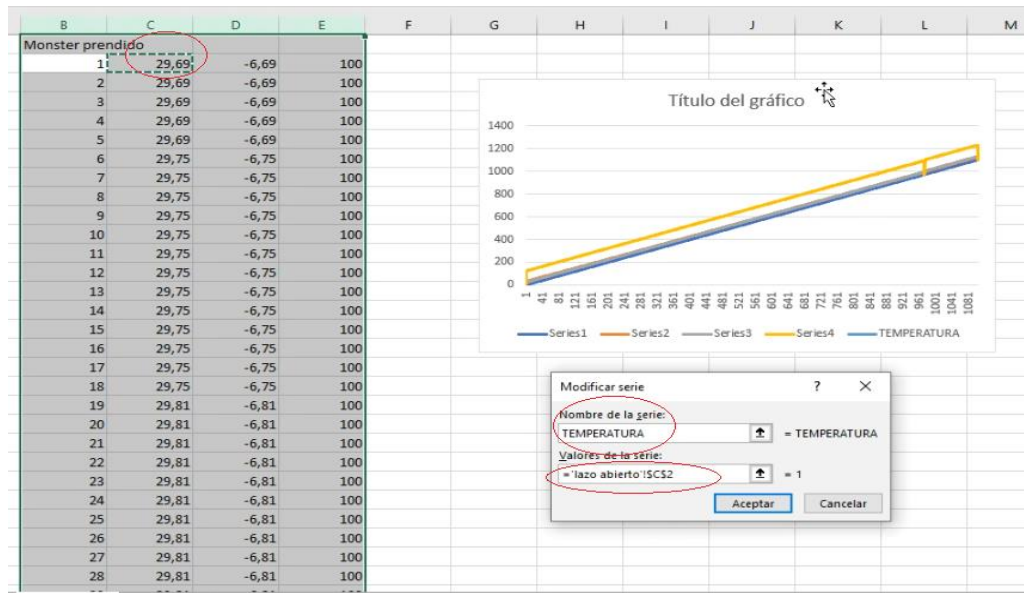


Figura 95. Configuración Excel parte 11.

Se selecciona el ultimo valor cuando se estabiliza la planta para dar clic en aceptar.

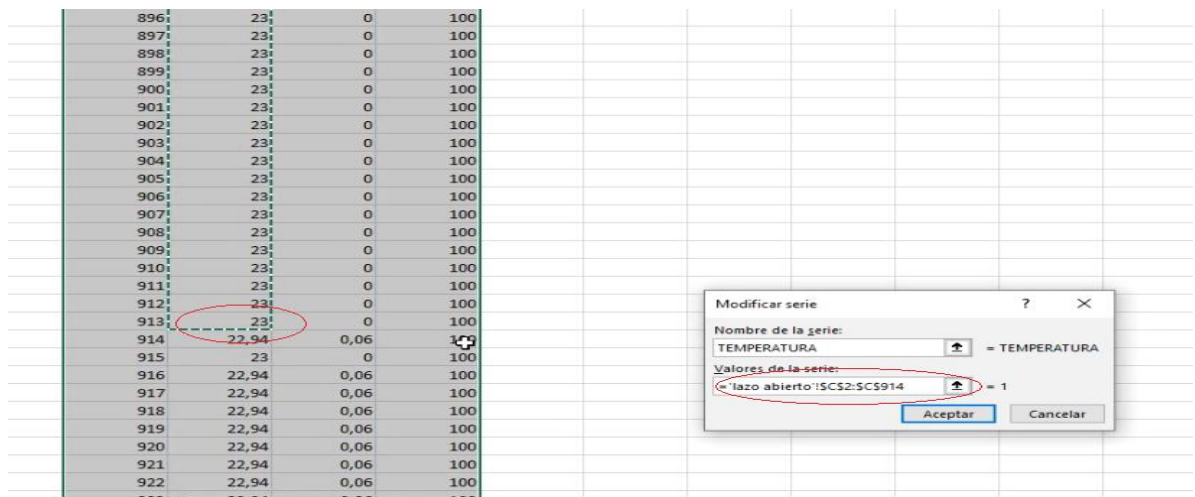


Figura 96. Configuración Excel parte 12.

Para terminar, se elimina todas las series anteriores, porque no son acorde al sistema.

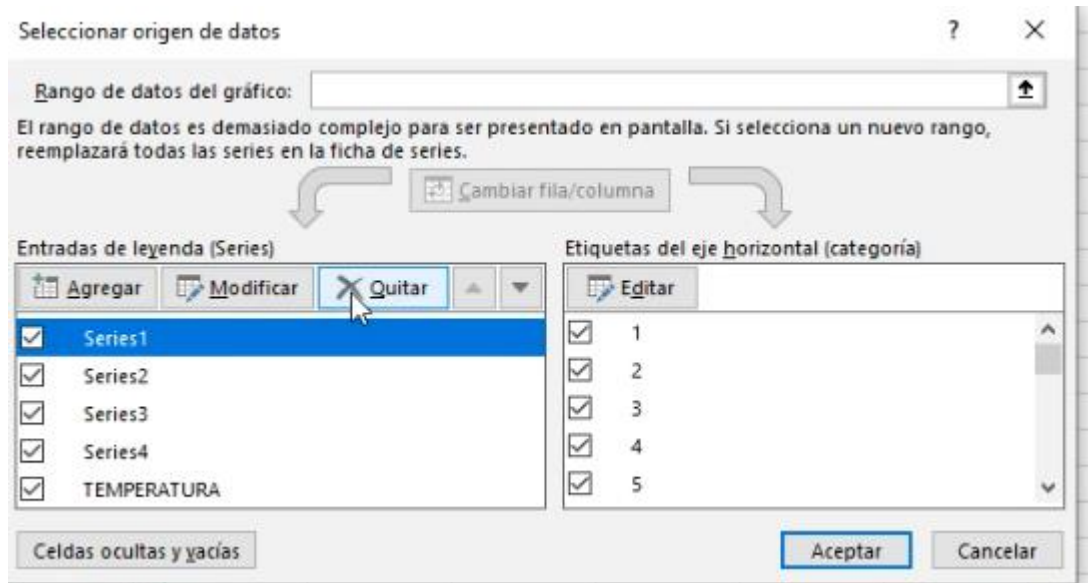


Figura 97. Configuración Excel parte 13.

Solo se deja la serie creada con el nombre temperatura.

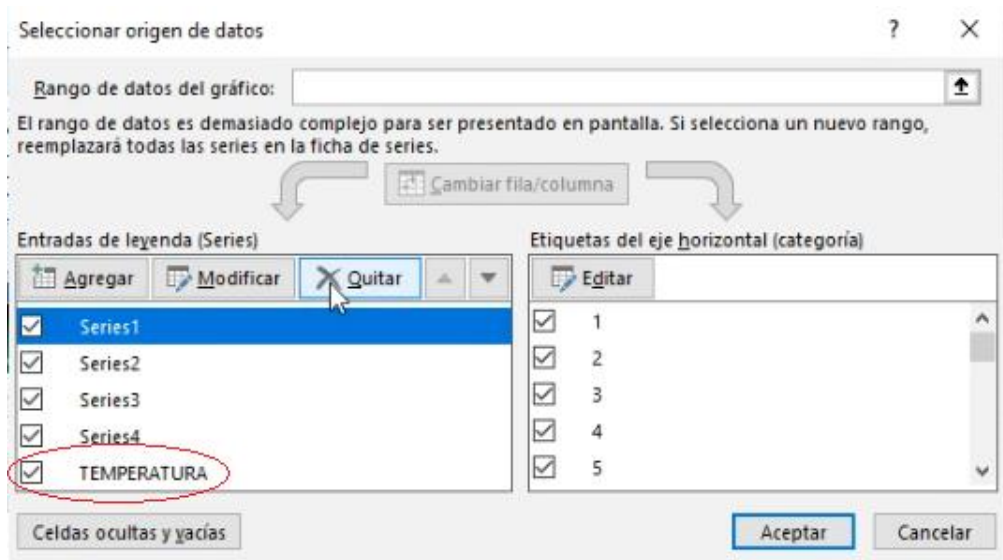


Figura 98. Configuración Excel parte 14.

Esta es la manera correcta para poder visualizar la planta y su comportamiento.

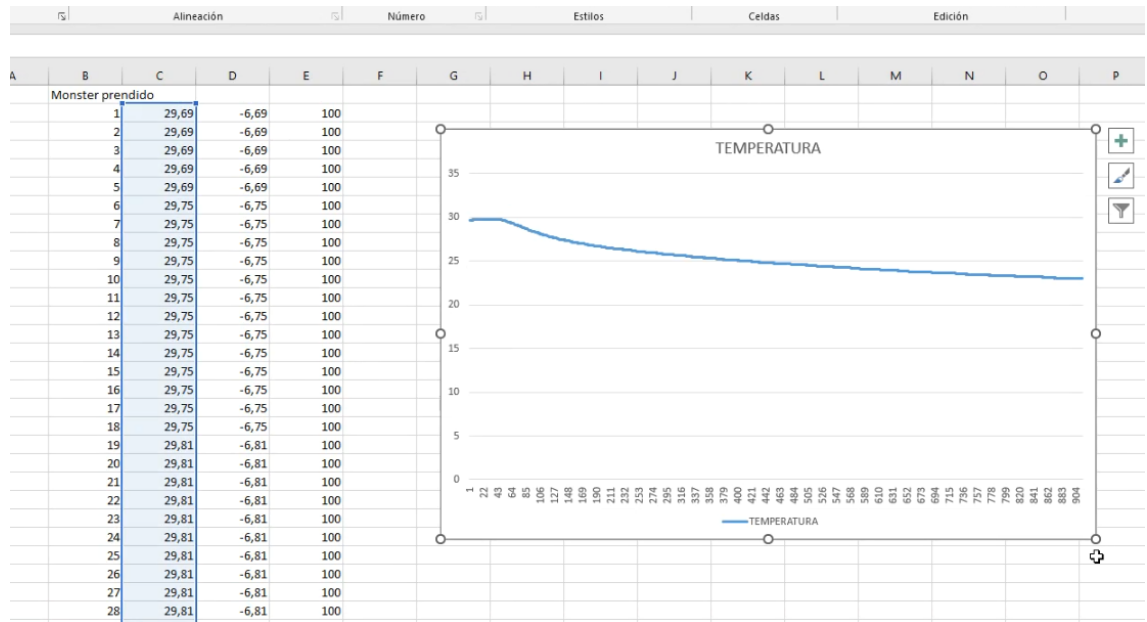


Figura 99. Configuración Excel parte 15.

6. CONCLUSIONES

- Al utilizar las celdas peltier TEC1-12706 se comprueba que para sistemas pequeños por su temperatura de -50°C hasta una temperatura máxima de 69°C es eficaz.
- Se trabaja con las celdas peltier TEC1-12706 de una mejor forma con una buena disipación utilizando un disipador grande y un cooler como extractor esto garantiza la durabilidad.
- La señal que se establece en lazo abierto se comporta como un escalón unitario, en lazo cerrado depende siempre de u .
- La señal PWM generada por el Arduino al Monster shield vnh2sp30 es necesaria para enviar el voltaje que requiere las celdas peltier TEC1-12706 para trabajar porque son analógicas.
- Al comprobar el método de Ziegler y Nichols se debe cumplir varias cosas como la función de transferencia sea de primer orden, la planta tiene una forma sigmoideal y si no cumple esos requisitos se debe utilizar otro método para poder estabilizar la planta.
- El método de Ziegler y Nichols nos ayudó hallar los valores de los coeficientes faltantes y discretizar la planta de manera más rápida y segura para colocar los parámetros calculados dentro de la programación de Arduino.
- El mejor rango regulado para la planta es de 29°C y 23°C porque trabaja de manera más eficiente esto depende del material del envase.
- El envase no era el adecuado para enfriar en su totalidad se establece pérdidas y genera un gradiente mayor.
- En 7 minutos hay una variación de $0,13^{\circ}\text{C}$ por lo que se concluye que para fines prácticos la planta está estabilizada.
- Se discretiza en período de tiempo para tomar la muestra en Arduino y luego ver como se comporta.

7. RECOMENDACIONES

- Cambiar el tipo de envase porque el de la planta esta hecho de acero inoxidable al analizar la planta contiene mucha perdida de enfriamiento y un mejor material de conducción para el frío es el aluminio.
- El envase sea totalmente cerrado, para no perder el mínimo tiempo de frío.
- En el ensamblaje se debe tener precaución aislando bien los cables donde cada conductor debe mantener distancia para al final revisar los malos contactos para evitar corto circuito y dañar el equipo.
- Etiquetar los cables o conductores para identificar las conexiones.
- Se utiliza una tierra común en la tarjeta PBC creada para evitar malos funcionamiento en los módulos utilizados y el Arduino.
- Al energizar correctamente la planta y verificar que al momento de funcionar monster moto shield vnh2sP30 el ventilador de disipación debe estar encendido, porque de no ser así las celdas peltier llegan a un punto frío máximo pero se comienzan a calentar de ambo lados si no tiene una buena disipación causando que se dañen.
- Dentro de la tarjeta PBC se debe medir el regulador 7805 este permite alimentar los módulos y componentes que trabajan a 5v DC.
- Utilizar un ventilador o disipadores en el módulo Monster Moto shield vnh2sP30 ya que estas están conectadas a su salida con las celdas peltier y esto puede dañar los microcontroladores de la tarjeta ubicados a la salida.
- El Arduino suele fallar al no tener actualizado el programa porque lo detecta como un controlador antiguo, para corregir eso es bueno tener la última versión de Arduino.
- Antes de hacer funcionar el proyecto se debe hacer una prueba sin agua para ver si las celdas peltier están trabajando de manera correcta.

8. PROYECTOS DE INVESTIGACIÓN VINCULADOS

Termoelectricidad: uso de las celdas peltier en el campo de la refrigeración y sus principales aplicaciones.

Autores: Cristian Rubio Ramón, Guillermo Martheyn Lizarazo.

Universidad: Corporación Universitaria Minuto de Dios.

Sistema De climatización Basado En Células Peltier.

Autor: Sergio López Campos.

Universidad: Universidad Carlos III De Madrid.

Diseño E Implementación Del Sistema De Climatización Para Un Automóvil Honda Civic Por Efecto Peltier.

Autor: Eliud Romero Delgadillo.

Universidad: Universidad Católica De Santa María.

Smart Home Automation System Monitor and Control Via Bluetooth And Internet Connectivity by Android Application

Autor: Nikitha Tipparti, K Suresh.

Universidad: International Journal Of Advanced Technology and Innovative Research.

9. REFERENCIAS BIBLIOGRÁFICAS

- Alfaro, V. (2011). Actualización Del Método De Sintonización De Controladores De Ziegler Y Nichols. *Revista Ingeniería*, 15(1–2). <https://doi.org/10.15517/ring.v15i1-2.7739>
- Alibaba. (2022). *120w fuente de alimentación regulable 12v 15 amp s-120-24 24vdc 12 volt power supply 24v 5a 12v 15a*. https://www.alibaba.com/pla/120w-fuente-de-alimentación-regulable-12v_60223452584.html?mark=google_shopping&biz=pla&pcy=US&searchText=Power+Supply&product_id=60223452584&src=sem_ggl&from=sem_ggl&cmpgn=16513912513&adgrp=139882389568&fditm=&tgt=pla-29394677
- amazon. (2022). *Ovente Calentador de inmersión eléctrico para coche, para té, café y sopa, 150 W, portátil, sumergible, agua caliente instantánea, calentador de acero inoxidable, peso ligero y fácil de limpiar, color negro, paquete de 1 CH1121: Industrial y Científico*. <https://www.amazon.com/Ovente-CH112-CH1121B-Immersion-Liquid/dp/B00USFWQEA?th=1>
- appinventor. (2022). *MIT App Inventor*. <https://appinventor.mit.edu/>
- Arduino. (2022). *Arduino Nano*. <https://arduino.cl/arduino-nano/>
- Campos, S. (2014). *Sistema de climatización basado en células peltier*. 150. https://e-archivo.uc3m.es/bitstream/handle/10016/24294/PFC_Sergio_Rafael_Lopez_Campos.pdf
- Cuyubamba, S. (2018). *Diseño óptimo de un sistema de iluminación LED aplicando el efecto Seebeck en celdas Peltier*. Distrito de Yanahuanca, Región Pasco, 2018. 7(2), 44–68.
- Daier. (2017). *Datasheet Switch* (pp. 1–7). <https://www.chinadaier.com/wp-content/uploads/2017/08/Rocker-switch.pdf>
- Daiertek. (2022). *Datasheet Switch de Bloqueo*. 0, 1–5. <https://es.aliexpress.com/item/32953431766.html>
- Delta. (2022). *Datasheet Power Suppli 12vDC-15A*. 1–3. https://shopdelta.eu/pdf.php?page=shop/datasheet&product_id=3962
- Elecok. (2022). *Datasheet Cooler 12 V*. 1. [https://www.elecok.com/es/data_sheet/190014/FENGHAO_HL9225S12H_12V_0%2C20A_2_Cable_Enfriamiento_Ventilador_\(ES\).pdf](https://www.elecok.com/es/data_sheet/190014/FENGHAO_HL9225S12H_12V_0%2C20A_2_Cable_Enfriamiento_Ventilador_(ES).pdf)
- Etter, D. M. (2011). *Introduction to MATLAB ® Second Edition*. 1–197.

<http://dl.icdst.org/pdfs/files3/6c7882dfb3a85367f9c3410b72588d3a.pdf>

everlight. (2007). *LED_DATASHEET*. <http://www.datasheet.es/PDF/1087073/334-15-T1C1-4WYA-pdf.html>

Fezari, M., & Dahoud, A. Al. (2019). *Exploring One-wire Temperature sensor “DS18B20” with Microcontrollers*. February. https://www.researchgate.net/profile/Mohamed-Fezari-2/publication/330854061_Exploring_One-wire_Temperature_sensor_DS18B20_with_Microcontrollers/links/5c58388d92851c22a3a832d2/Exploring-One-wire-Temperature-sensor-DS18B20-with-Microcontrollers.pdf

García, P. (2012). *Estudio y cuantificación de la demanda energética en una vivienda aislada*. https://riunet.upv.es/bitstream/handle/10251/15137/proyecto_definitivo_14.12.2011.pdf?sequence=1&isAllowed=y

GCE. (2022). *Datasheet Resistencia*. https://moam.info/tabla-de-resistencias_5a04d0361723dd3827e2eff5.html

giraldo, juan. (2018). *Datasheet Módulo BLUETOOTH HC-06*. <http://juangiraldo0.blogspot.com/2018/07/configuracion-módulo-bluetooth-hc-06.html>

GONZÁLEZ, I. (2014). DIAGNÓSTICO DEL SISTEMA DE CONTROL DE TEMPERATURA Y HUMEDAD DEL LABORATORIO DE METROLOGÍA DEL CDT DE GAS, UBICADO EN EL MUNICIPIO DE PIEDECUESTA, SANTANDER Y DESARROLLO DE LA METODOLOGÍA PARA LA SINTONIZACIÓN DE UN SISTEMA DE CONTROL DE TEMPERATURA. *Paper Knowledge . Toward a Media History of Documents*, 7(2), 107–115.

GUALPA, R. (2018). *DISEÑO DE UN PROTOTIPO DE MONITOREO DE LA CALIDAD DEL AIRE*. http://repositorio.ug.edu.ec/bitstream/redug/40636/1/TESIS_COMPLETA_DARIO_GUALPA.pdf

Guierrez, G. (2017). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CLIMATIZACIÓN AUTOMATIZADA PARA PERSONAS CON DISCAPACIDADES FÍSICAS” PRESENTADA*. 1–110.

Herraiz, G. (2012). Upv - hdi ANDROID. *Historia de La Informática*, 40. <http://dl.icdst.org/pdfs/files3/6c7882dfb3a85367f9c3410b72588d3a.pdf>

innovationM. (2021). *Android - Apps en Google Play*. https://play.google.com/store/apps/details?id=com.innovationm.myandroid&hl=es_PA&gl=US

- Kaith, D., B. Patel, J., & Gupta, N. (2015). An Introduction to Functional Verification of I2C Protocol using UVM. *International Journal of Computer Applications*, 121(13), 10–14. <https://doi.org/10.5120/21599-4703>
- KEC. (2010). datasheet regulador 7805. *Control*, 1–9. <https://www.mantech.co.za/Datasheets/Products/KIA7805-24AP.pdf>
- kiwi. (2022). *Waterbestendige DS18B20 digitale temperatuursensor + weerstand - 5 meter*. https://www.kiwi-electronics.com/nl/waterbestendige-ds18b20-digitale-temperatuursensor--weerstand-5-meter-3709?language=nl-nl¤cy=EUR&gclid=CjwKCAjwwdWVBhA4EiwAjcYJECEQ3dlc2LzYZYMZOu6KIWvIQ-B15BS3o6dw67_M-cwSfM2ID7LS0hoCSAgQAvD_BwE
- LAMPIRAN. (2022). *Datasheet Arduino Nano*. <https://repo.undiksha.ac.id/1980/9/1705031010-LAMPIRAN.pdf>
- Márquez, V. E., Valencia-López, J. J., García-Aranda, O., & Heard, C. (2016). Determinación computacional del coeficiente de transferencia de calor en calentadores eléctricos de flujo continuo, mediante dinámica de fluidos computacional. *Informacion Tecnologica*, 27(5), 151–162. <https://doi.org/10.4067/S0718-07642016000500017>
- Martek. (2022). *Datasheet Conector de Voltaje CON-00 2 - M. 1*. <http://www.electrosistemas.com.ar/datasheets/accesorios/CON-002-M.pdf>
- Maximintegrated. (2015). DATASHEET Ds18B20. *DATASHEET*, 92(1 (35)), 1–20. <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- Mohammad, A. (2020). *Interfaz VNH2SP30 Monster Motor Shield con Arduino - Electropeak*. <https://electropeak.com/learn/interfacing-vnh2sp30-monster-motor-driver-shield-with-arduino/>
- Moler, C. (2018). *MATLAB - El lenguaje del cálculo técnico - MATLAB & Simulink*. <https://la.mathworks.com/products/matlab.html>
- Mouser. (2022). *datasheet Potenciometro*. http://www.datasheet.es/PDF/866982/RV24AF-10-15R1-B2K-pdf.html#google_vignette
- Mucientes, S. J. D. (2021). *Implementación de un entorno de comunicación Bluetooth basado en el módulo HC-06*. <https://uvadoc.uva.es/handle/10324/48095>
- Nangoi, A. D. (2015). *APPLIANCE SWITCH UTILIZED BY LIGHT SENSOR Andrew Darren Nangoi President University DECLARATION OF ORIGINALITY*. April. <http://repository.president.ac.id/bitstream/handle/123456789/1407/002201100017>.

pdf?sequence=1&isAllowed=y

NIKITHA, T., & SURESH, K. (2016). *Smart Home Automation System Monitor and Control Via Bluetooth And Internet Connectivity by Android Application*. 08(14), 2758–2763. www.ijergs.org

Ogata, K. (2010). *Ingeniería de Control Moderna*.

Planas, carla. (2016). *PARA LA DIFERENCIACIÓN DE CÉLULAS MADRE Trabajo de Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona (Universidad Politécnica de Cataluña) por Carla Alonso Planas En cumplimiento parcial de los requisi.* <https://upcommons.upc.edu/handle/2117/89426>

POTINGO, L. (2011). *REDISEÑO Y CONSTRUCCIÓN DE UN TERMO-ESTIMULADOR*. <http://cybertesis.uach.cl/tesis/uach/2011/bmfcp816r/doc/bmfcp816r.pdf>

Ramírez, C., Lizarazo, G., & Vera, E. (2017). Termoelectricidad: uso de las celdas peltier en el campo de la refrigeración y sus principales aplicaciones. *Inventum*, 12(22), 9–16. <https://doi.org/10.26620/uniminuto.inventum.12.22.2017.9-16>

rlocman. (2022). *Datasheet TEC1-12706*. <https://www.rlocman.es/datasheet/pdf.html?di=173597>

Romero, E. (2017). *“Diseño E Implementación Del Sistema De Climatización Para Un Automóvil Honda Civic Por Efecto Peltier.”* <https://tesis.ucsm.edu.pe/repositorio/bitstream/handle/UCSM/6360/4A.0242.IM.pdf?sequence=1&isAllowed=y>

salamanca, jose. (2017). *Ley de Stefan-Boltzmann*. <https://sites.google.com/site/201702joselsalamanca/apuntes/semana-2>

Serrano, M. (2019). *SISTEMA DE ADQUISICIÓN DE DATOS EN LA GENERACIÓN DE ENERGÍA MEDIANTE EL USO DE CELDAS PELTIER*.

Soto, C. A. (2015). *BRAZO ROBÓTICO DE 5GDL CON SISTEMA DE CONTROL MODIFICABLE POR EL USUARIO PARA FINES DE INVESTIGACIÓN EN INGENIERÍA ROBÓTICA Tesis*. <https://uvadoc.uva.es/handle/10324/48095>

st. (2022). *Datasheet MONSTER_SHIELD_vnh2sp30*. <https://www.st.com/en/automotive-analog-and-power/vnh3asp30-e.html>

Sycelectronica. (2022). *Datasheet Bornera. 500*. <https://www.sycelectronica.com.ar/borneras/EK500V-XXP.pdf>

Type, R. (2022). *datasheet capacitor electrolitico*. 0–1.
<https://www.electron.com/media/14677/datasheet-123-338.pdf?&key=ZGpmlyQwNUZfMTQ2Nzc=>

Ufoaroh, S. U., Oranugo, C. O., & Uchechukwu, M. E. (2015). Heartbeat Monitoring and Alert System Using Gsm Technology. *International Journal of Engineering Research and General Science*, 3(4), 26–34. www.ijergs.org

Vishay. (2013). Datasheet LCD 16X2. *Datasheet*, 1–3.
<https://www.vishay.com/docs/37481/lcd016n002mttiet.pdf>

Vishay. (2021). *datasheet capacitor ceramico*.
<https://www.mouser.com/datasheet/2/427/564r565r-1762222.pdf>

Watson, D. (2018). *Introduction to Arduino Nano - The Engineering Projects*.
<https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html>

10. ANEXOS

ANEXO A. CRONOGRAMA DE DURACIÓN DEL PROYECTO.

Actividad	Meses												
	1	2	3	4	5	6							
Adquisición de equipos y componentes.													
Diseño termo.													
Instalación- Energización de los Equipos.													
Pruebas de Equipos.													
Programación Etapa de pruebas Finales.													
Escritura del Documento.													

TABLA 7. Cronograma de proyecto de tesis.

ANEXO B. PRESUPUESTO DEL PROYECTO

Descripción	Cantidad	Costo unitario	Costo final
Aldaba	1	\$1,50	\$1,50
Arduino nano	1	\$25,00	\$25,00
Bluetooth hc-06	1	\$10,00	\$10,00
Bisagra	1	\$2,00	\$2,00
Borneras doble	5	\$0,75	\$3,75
Borneras triple	2	\$1,00	\$2,00
Cable dúplex eléctrico calibre 16	1	\$3,50	\$3,50
Cable mini usb b transmisión de datos	1	\$4,00	\$4,00
Cable pla shiny	2	\$28,99	\$57,98
Cable pla simple	1	\$27,00	\$27,00
Cable utp	1	\$1,40	\$1,40
Candado	1	\$1,60	\$1,60
Celda peltier	2	\$6,10	\$12,20
Condensador 100µf/25v	1	\$0,15	\$0,15
Condensador cerámico 10nf/25v	1	\$0,15	\$0,15
Conector de corriente plug macho y hembra	4	\$1,20	\$4,80
Cooler pequeño 12v 0.06A	2	\$2,25	\$4,50
Cooler grande 12v 0.20a	1	\$5,00	\$5,00
Diodo leds	4	\$0,10	\$0,40
Disipador para Monster shield	4	\$0,50	\$2,00
Disipador grande para celda peltier	1	\$7,00	\$7,00
Disipadores pequeños para celda peltier	2	\$4,00	\$8,00
Envase acero inoxidable	1	\$85,00	\$85,00
Etiqueta	14	\$2,00	\$28,00
Fusible 30 A	1	\$0,35	\$0,35
I2c lcd 1602 16x02	1	\$4,00	\$4,00
Impresión 3D	1	\$55,00	\$55,00
Interruptor de 2 posiciones	1	\$0,85	\$0,85
Interruptor de bloqueo	1	\$4,45	\$4,45
Lcd 16x2	1	\$8,00	\$8,00
Monster shield vnh2sp30	1	\$38,00	\$38,00
Pernos 4cm para disipadores pequeños	2	\$0,15	\$0,30
Pernos 16 cm parte cuadrada	4	\$0,45	\$1,80
Pernos parte frontal 6cm	4	\$0,35	\$1,40
Placa de reconocimiento 10x7cm	1	\$14,00	\$14,00
Placa pbc doble cara "anillos metalizados"	1	\$75,00	\$75,00
Porta fusible	1	\$0,65	\$0,65
Potenciómetro 10kΩ	1	\$0,35	\$0,35
Power suppli 12vdc-15a	1	\$35,00	\$35,00
Regulador 7805	1	\$0,65	\$0,65
Resistencia 1kΩ,330Ω,4.6Ω	5	\$0,10	\$0,50
Rollo de estaño	1	\$5,00	\$5,00
Rollo de etiqueta	1	\$3,25	\$3,25
Sensor de temperatura ds18b20	1	\$4,90	\$4,90
Tuerca parte cuadrada	4	\$0,20	\$0,80
Tuercas para disipadores pequeños	2	\$0,15	\$0,30
Tuercas parte frontal	4	\$0,15	\$0,60
Total			\$552,08

TABLA 8. PRESUPUESTO DEL PROYECTO.

ANEXO C. DATASHEET MÓDULOS

DATASHEET MÓDULO ARDUINO NANO

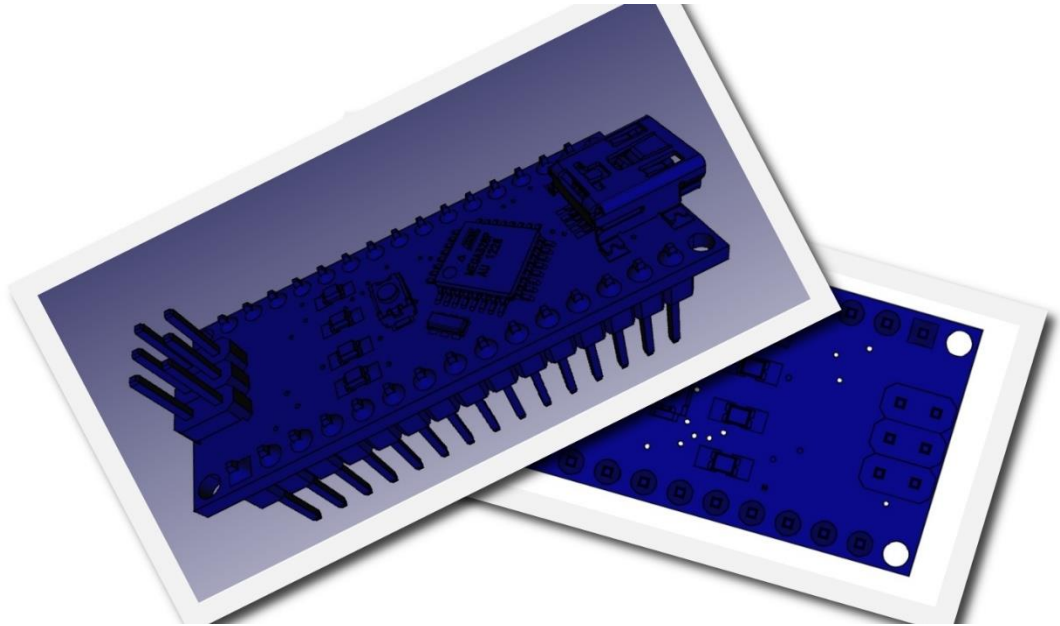


Figura 100. Módulo Arduino nano.

CATEGORÍA DE PINES	NOMBRE DE PIN	DETALLES
ENERGÍA	Vin, 3.3v, 5v, Tierra	Vin: (12V). 5V: utiliza para energizar los componentes del microcontrolador en la placa.
REINICIAR	REINICIAR	Restablece.
PINES ANALÓGICOS	A0-A7	Mide voltaje analógico de 0-5v
PIN DE ENTRADA Y SALIDA	DIG D0-D13	Se puede utilizar como pines de entrada o salida. 0v(bajo(y 5v(alto).
SERIAL	Rx, Tx	Recibir y transmitir datos en serie TTL.
INTERRUPCIONES EXTERNAS	2,3	Para activar una interrupción.
PWM	3,5,6,9,11	Salida por pin 8 bits.
SPI	10(ss), 11(MOSI), 13(SCK)	comunicación SPI.
LED INCORPORADO	13	Enciende el LED
CII	A4(SDA), A5(SCA)	Se utiliza para la comunicación TWI.
AREF	AREF	voltaje de referencia de v de entrada.

TABLA 9. Datasheet Arduino nano.(LAMPIRAN, 2022)

DATASHEET MÓDULO BLUETOOTH HC-06

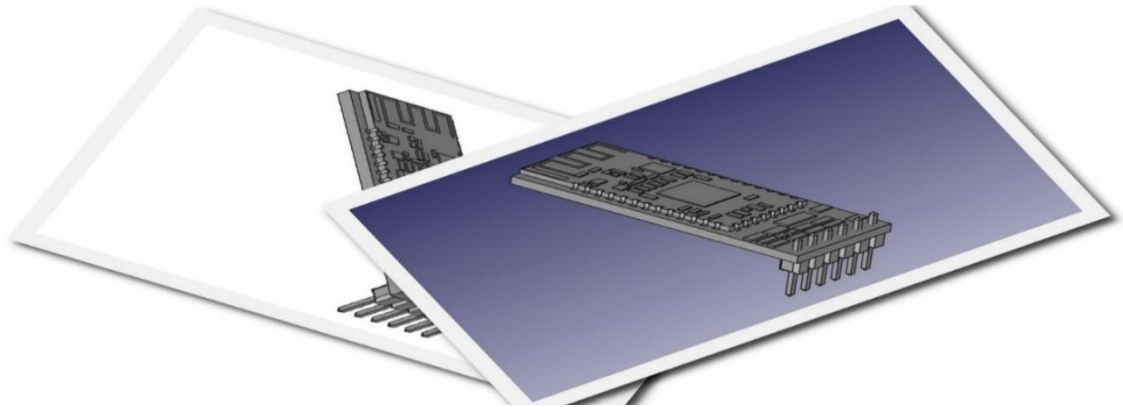


Figura 101. Módulo BLUETOOTH HC-06.

<r>	Observaciones
1200	configurado a 1200bps
2400	configurado a 2400bps
4800	configurado a 4800bps
9600	configurado en 9600bps (predeterminado)
19200	configurado a 19200bps
38400	configurado a 38400bps
57600	configurado a 57600bps
115200	configurado a 115200bps

TABLA 10. Configuración Predeterminada BLUETOOTH HC-06. (giraldo, 2018)

DATASHEET MÓDULO MONSTER SHIELD

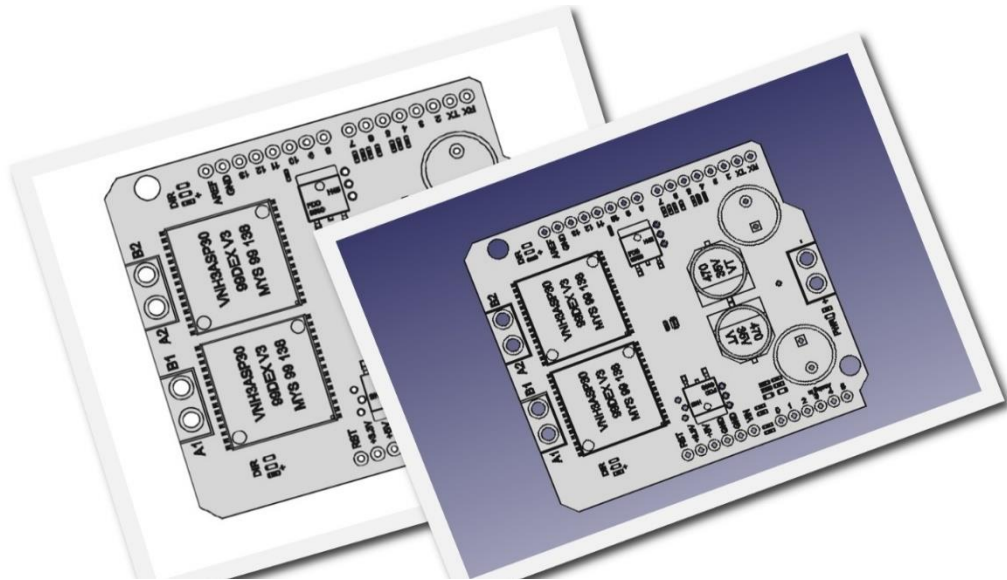


Figura 102. Módulo MONSTER SHIELD.

Símbolo	Parámetro	Valor	Unidad
VCC	Tensión de alimentación	+41	V
I _{max}	Corriente máxima de salida (continua)	30	A
I _R	Corriente de salida inversa (continua)	-30	
I _{IN}	Corriente de entrada (INAY enBpatas)	±10	mA
I _{EN}	Activar corriente de entrada (DIAGA/ESAY DIAGNÓSTICOB/ESB pin)	±10	
I _{pw}	Corriente de entrada PWM	±10	
VCS	Tensión máxima de detección de corriente	-3/+15	V
VESD	Descarga electrostática (R = 1.5k-, C = 100pF)		
	– Pasador CS	2	kV
	– Pines lógicos	4	kV
	– Pines de salida: SALIDA A, AFUERA B, VCC	5	kV
T _j	Temperatura de funcionamiento de la unión	limitada internamente	°C
T _c	Temperatura de funcionamiento de la carcasa	- 40 a 150	
T _{STG}	Temperatura de almacenamiento	- 55 a 150	

TABLA 11. Datasheet Módulo MONSTER SHIELD(st, 2022)

DATASHEET MÓDULO LCD 16X2

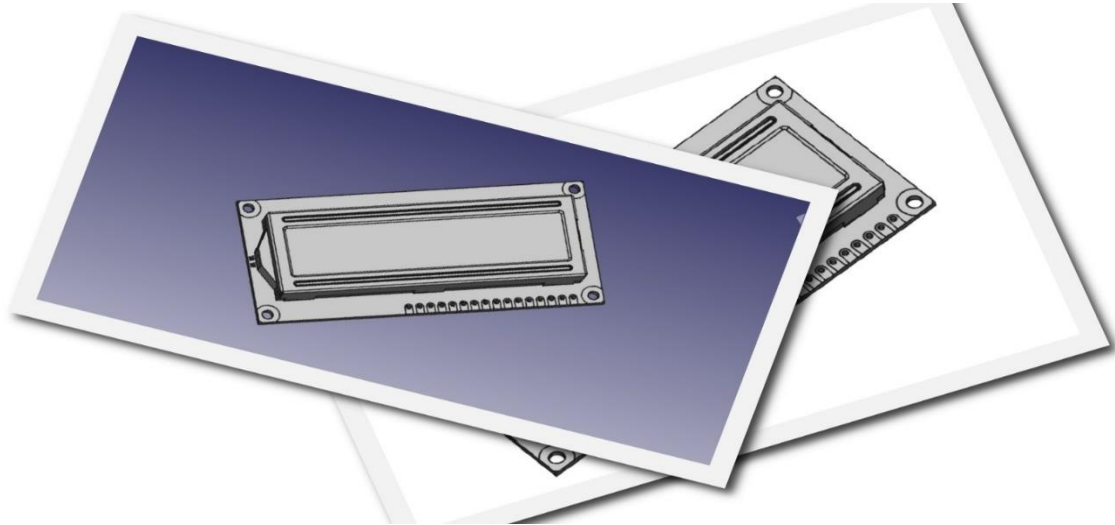


Figura 103. Módulo LCD 16x2.

CARACTERÍSTICAS ELÉCTRICAS						
ARTÍCULO	SÍMBOLO	CONDICIÓN	VALOR ESTÁNDAR			UNIDAD
			MÍN.	TIP.	MÁX.	
Voltaje de entrada	VDD	VDD = + 5 V	4,7	5.0	5,3	V
Corriente de suministro	IDD	VDD = + 5 V	-	1,6	1,5	mA
Conducción LC recomendada Módulo de versión de voltaje para Temperatura normal	VDD to V0	- 20 °C	-	-	5,2	V
Conducción LC recomendada Módulo de versión de voltaje para Temperatura normal		0 °C	-	-	4,5	
Conducción LC recomendada Módulo de versión de voltaje para Temperatura normal		25 °C	4,2	4,2	-	
Conducción LC recomendada Módulo de versión de voltaje para Temperatura normal		50 °C	3,8	-	-	
		70 °C	3,5	-	-	
Voltaje directo LED	VF	25 °C	-	4,2	4,6	V
Corriente directa LED - Matriz	IF	25 °C	-	260	520	mA
Corriente de la fuente de alimentación	IEL	VEL = 110 VAC, 400 Hz	-	-	5.0	mA

TABLA 12. Datasheet Módulo LCD 16x2(Vishay, 2013)

DATASHEET CELDA PELTIER

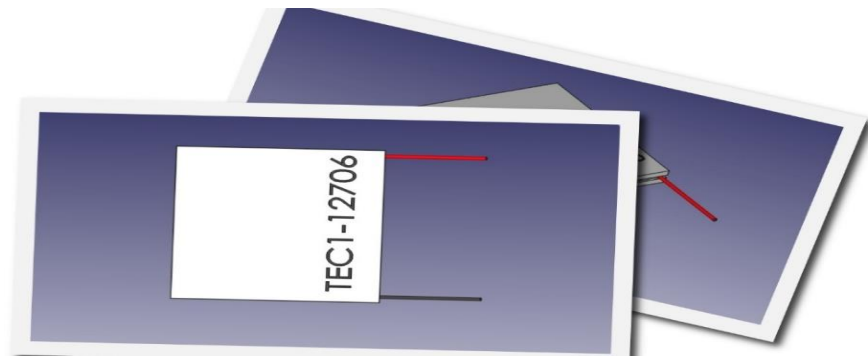


Figura 104. Celda Peltier.

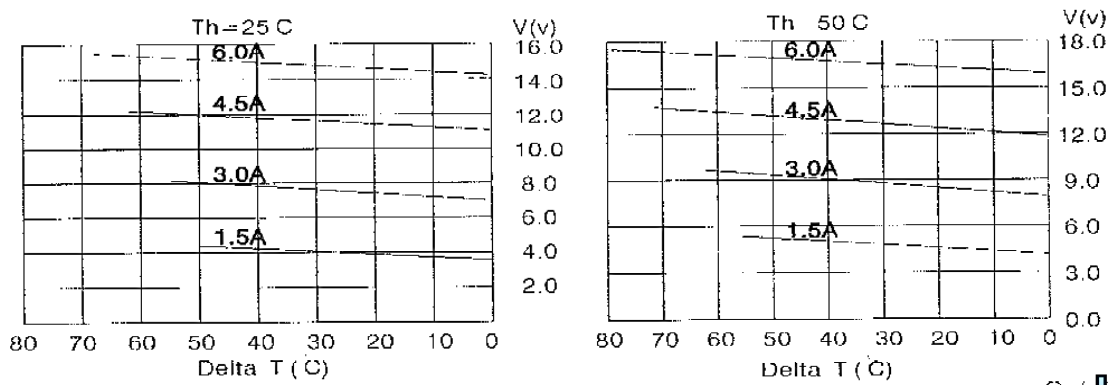


Figura 105. Gráfica de Comportamiento Celda Peltier.(rlocman, 2022)

Temperatura del lado caliente (°C)	24°C	49°C
Qmáx (Wattios)	49	56
Delta Tmáx (°C)	65	74
Imax (Amperios)	6,3	6,4
Vmax (Voltios)	14,3	16,3
Resistencia del módulo (Ohmios)	1,97	2,2

TABLA 13. Datasheet Celda Peltier.(rlocman, 2022)

DATASHEET SENSOR DS18B20



Figura 106. Sensor DS18B20.

PARÁMETRO	SÍMBOLO	CONDICIONES		MÍN.	TÍPICO	MÁX.	UNIDADES
Voltaje de suministro	VDD	Energía local (Nota 1)		+3.0		5,5	V
Tensión de alimentación pull-up	VPU	Poder parásito	(Notas 1, 2)	+3.0		5,5	V
		Poder local		+3.0		VDD	
Error de termómetro	TERR	- 10°C a +85°C	(Nota 3)			±0,5	°C
		- 30°C a +100°C				±1	
		- 55°C a +125°C				±2	
Lógica de entrada baja	VIL	(Notas 1, 4, 5)		-0,3		0,8	V
Lógica de entrada alta	VIH	Poder local	(Notas 1,6)	+2.2		El más bajo de 5,5 o VDD + 0,3	V
		Poder parásito		+3.0			
Sumidero actual	IL	IV/S = 0,4 V		4.0			mA
Corriente de espera	IDDS	(Notas 7, 8)			750	1000	nA
Corriente activa	IDD	VDD = 5V (Nota 9)			1	1,5	mA
Corriente de entrada DQ	IDQ	(Nota 10)			5		µA
Deriva		(Nota 11)			±0.2		°C

TABLA 14. Datasheet Sensor DS18B20.(Maximintegrated, 2015) .

DATASHEET POWER SUPPLI 12VDC-15A

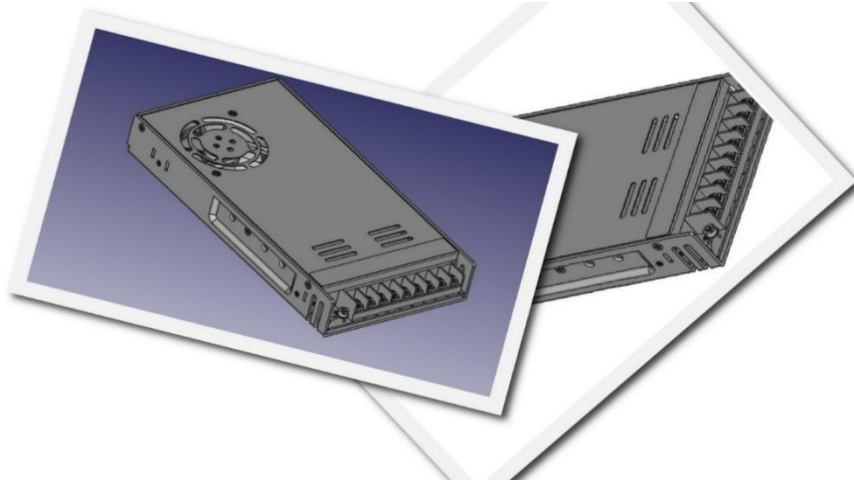


Figura 107. Power Suppli 12vDC-15A.

Tipo de fuente de alimentación:	Conmutada
Tensión de alimentación:	120/230 V AC
Voltaje de salida:	12 V DC
Tensión de salida:	9.. 12 V
Carga máxima alimentación:	15 A
Potencia:	190 W
Número de salidas:	2 uds.
Protecciones:	<ul style="list-style-type: none">• Contra cortocircuitos• Contra sobrecargas
Temp / humedad relativa:	-10 °C .. 50 °C / 20 % .. 90 %
Peso:	0.56 kg
Dimensiones:	200 x 100 x 43 mm

TABLA 15. Datasheet Power Suppli 12vDC-15A.(Delta, 2022)

DATASHEET DIODO LED

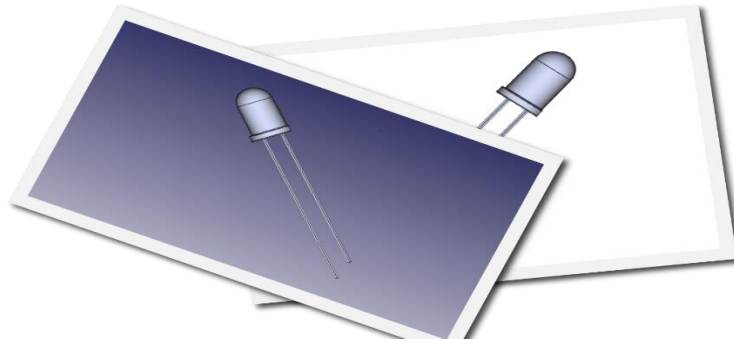


Figura 108. Diodo LED.

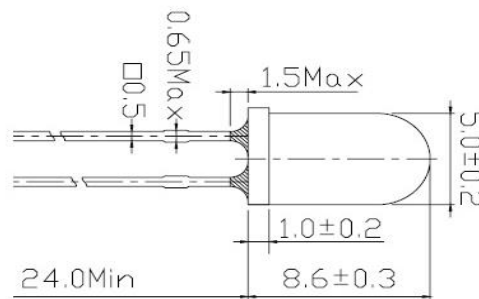


Figura 109. Dimensiones Diodo LED (everlight, 2007)

Parámetro	Símbolo	Clasificación	Unidad
Corriente directa continua	IF	30	mA
Corriente directa máxima (servicio / 10 a 1 KHZ)	IFP	100	mA
Tensión inversa	VR	5	V
Temperatura de funcionamiento	Topr	-40 ~ +85	°C
Temperatura de almacenamiento	Tstg	-40 ~ +100	°C
Temperatura de soldadura (T=5 seg)	Tsol	260 ± 5	°C
Disipación de potencia	Pd	100	mW
Corriente inversa Zener	Iz	100	mA
Descarga electrostática	ESD	4K	V

TABLA 16. Datasheet Diodo LED. (everlight, 2007)

DATASHEET RESISTENCIA

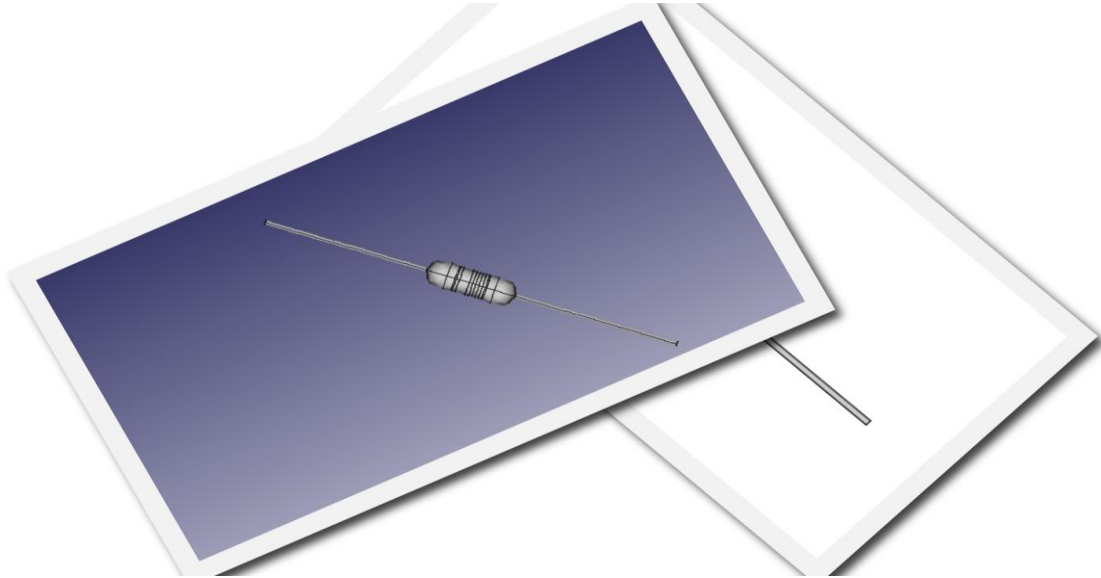


Figura 110. Resistencia.

CARACTERÍSTICAS	
Rango de valores	10A 1m
Disipación	1/4W a 70°C
Tolerancia	1% y 2%
Temperatura de operación	-65°C a 165°C
Coefficiente de Temperatura	100 ppm/°C
Máxima tensión de trabajo	250 VRMS
Marcaje	estándar de 5 bandas

TABLA 17. Datasheet Resistencia.(GCE, 2022)

DATASHEET BORNERA

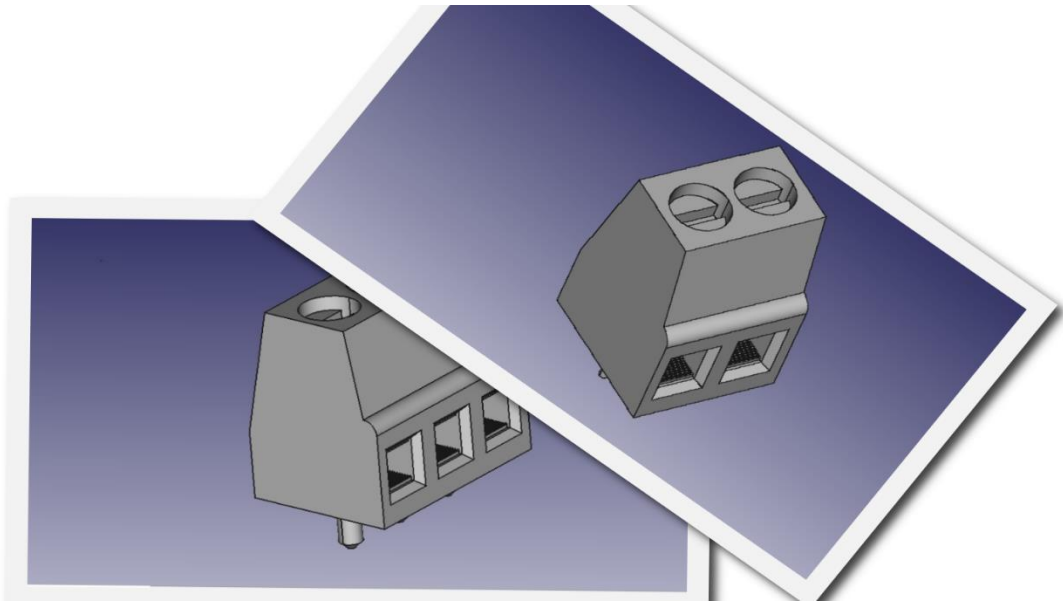


Figura 111. Bornera.

Clasificación	300V 20A (EC250V 10A)
Dimensión	5mm
Tensión soportada de aislamiento	AC 2000V/MIN
Resistencia de aislamiento	500V Ω o mas de DC500V
Rango de temperatura usado	-40°C ~ + 105°C
Cable eléctrico adecuado	24~12AWG(IEC 0.5 ~ 2.5 mm ²)

TABLA 18. Datasheet Bornera. (Sycelectronica, 2022)

DATASHEET REGULADOR DE VOLTAJE 7805

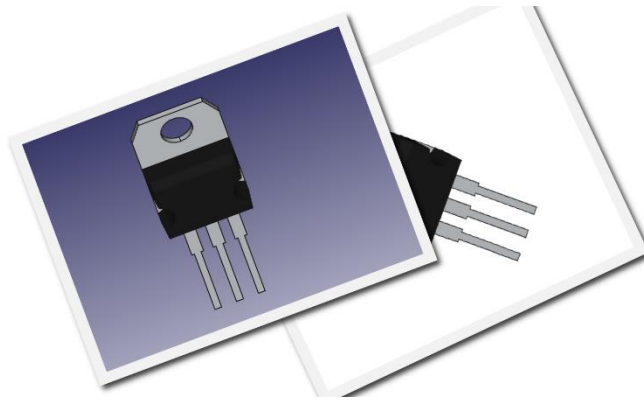


Figura 112. Regulador de voltaje 7805.

Característica	Símbolo	CONDICIÓN DE PRUEBA	MIN	TYP	MAX.	UNIT.	
Voltaje de salida	Vout	Tj=25°C , IOU=100mA	4.8	5.0	5.2	V	
Regulación de entrada	Línea de registro	Tj=25°C	7.0V ≤ VIN ≤ 25V	-	3	100	Mv
			8.0V ≤ VIN ≤ 12V	-	1	50	
Regulación de carga	Carga de registro	Tj=25°C	5mA ≤ Iout ≤ 1.5A	-	15	100	Mv
			249mA ≤ Iout ≤ 749mA	-	4	49	
Voltaje de salida	VOUT	7.0V ≤ VIN ≤ 20V	4,75	-	5.25	V	
Corriente de reposo	IB	Tj=25°C , IOU=5mA	-	4.2	8.0	mA	
Cambio de corriente de reposo	IB	7.0V ≤ VIN ≤ 25V	-	-	1.3	mA	
Voltaje de ruido de salida	VNO	Ta=25°C , 10Hz ≤ f ≤ 100kHz	-	50	-	μVrms	
Relación de rechazo de ondulación	RR	f=120Hz, 8.0V ≤ VIN ≤ 18V	62	78	-	dB	
Voltaje de caída	VD	Iout=1.0A, Tj=25°C	-	2.0	-	V	
Límite de corriente de cortocircuito	ISC	Tj=25°C	-	1.6	-	A	
Temperatura media coeficiente de voltaje de salida	TCVO	IOU=5mA, 0°C ≤ Tj ≤ 125°C	-	-0.6	-	Mv/°C	

TABLA 19. Datasheet Regulador de voltaje 7805.(KEC, 2010)

DATASHEET POTENCIÓMETRO

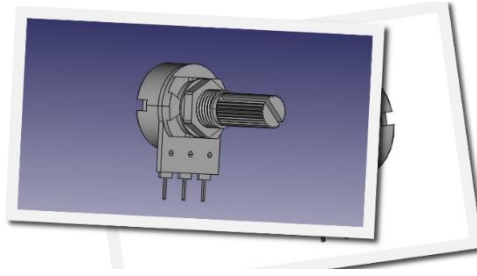


Figura 113. Potenciómetro.

Tolerancia de resistencia:	±20%
Afilas:	lineal (serie 31VA, serie 31VC), audio (serie 31VJ)
Tensión máxima de trabajo:	500V (cono lineal) 250 V (cono de audio)
Disipación de potencia:	.5W (cono lineal) .25W (cono de audio)
Resistencia de aislamiento:	> 100MΩ @ 500VDC
Vida de rotación:	15.000 ciclos
Elemento resistivo:	composición de carbono
Ángulo de rotación eléctrica:	285° ± 5°C
Ángulo de rotación:	300° ± 5°C
Par de rotación:	20~200 gf.cm
Fuerza de tope del eje:	> 6Kgf.cm/min

TABLA 20. Datasheet Potenciómetro.(Mouser, 2022)

DATASHEET CAPACITOR ELECTROLÍTICO

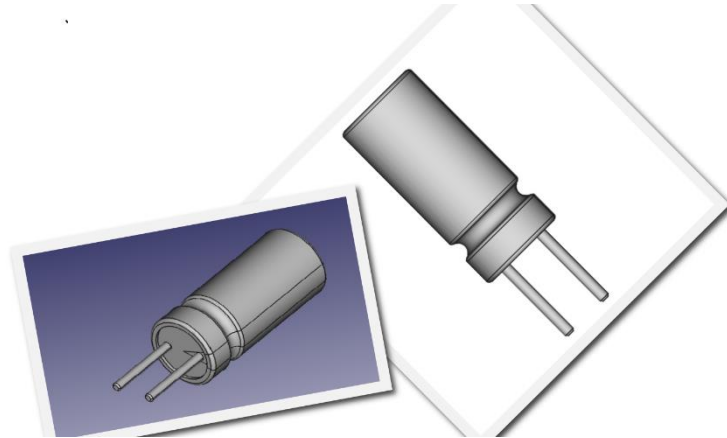


Figura 114. Capacitor electrolítico.

ESPECIFICACIONES								
Rango de operación de temperatura	-40 ~ +85°C							
Voltaje de trabajo nominal	4~ 63VDC							
Tolerancia de capacitancia (120Hz 20°C)	±20%(M)							
Estabilidad a baja temperatura.	Relación de impedancia a 120Hz							
	Tensión Nominal (V)	4	6.3	10	16	25	35	50 ~ 63
	-25°C / +20°C	6	4	3	2	2	2	2
-40°C / +20°C	12	8	6	4	4	3	3	
Carga de vida.	Después de 1000 horas de aplicación de W.V. a +85oC, el capacitor deberá cumplir con los siguientes límites.							
	Cambio de capacitancia	≤±20% del valor inicial						
	Factor disip.	≤200%						
Corriente de Fuga	≤valor especificado inicial							

TABLA 21. Datasheet Capacitor electrolítico.(Type, 2022)

DATASHEET CAPACITOR CERÁMICO

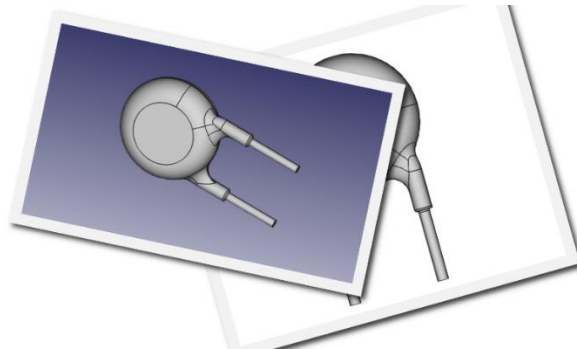


Figura 115. Capacitor Cerámico.

DESCRIPCION	
Tolerancia en capacitancia	$\pm 10\%$, $\pm 20\%$, -20% a $+80\%$
Factor de Disipación	Clase 1: 0,2 % máx. a 1 MHz; 1 V
	Clase 2: 2,0 % máx. a 1kHz; 1 V
Categoría rango de Temperatura	- 25 °C a +85 °C
Categoría climática ACC. EN 60068-1	25/085/21
Rango de temperatura de Funcionamiento	25 °C a +105 °C

TABLA 22. Datasheet Capacitor Cerámico.(Vishay, 2021)

DATASHEET CONECTOR DE ALIMENTACIÓN 12V

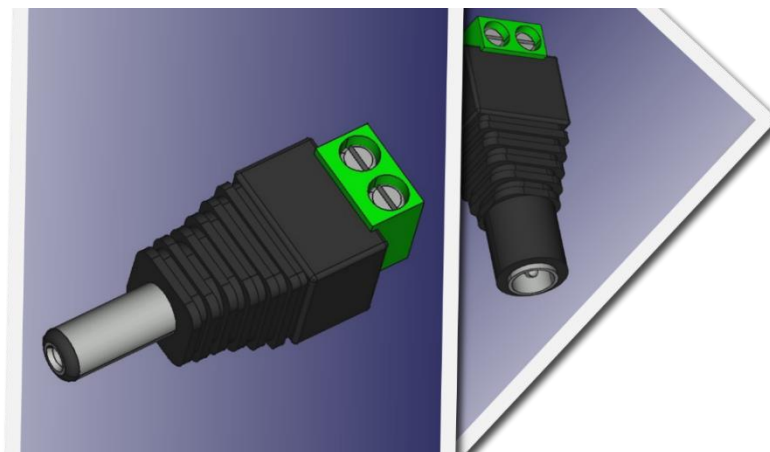


Figura 116. Conector de alimentación 12v.

Numero de modelo: CON-002-M	
TIPO:	Adaptador, enchufe.
APLICACIÓN:	Poder
CONECTOR:	Macho y Hembra
PIN:	Cobre (Cu)
Tipo de Cable:	Cable de alimentación.
Duración:	5000 veces.
Tamaño:	5,5*2,1mm
Color:	Negro y verde.

TABLA 23. Datasheet Conector de Voltaje CON-00 2 – M. (Martek, 2022)

DATASHEET COOLER 12 V

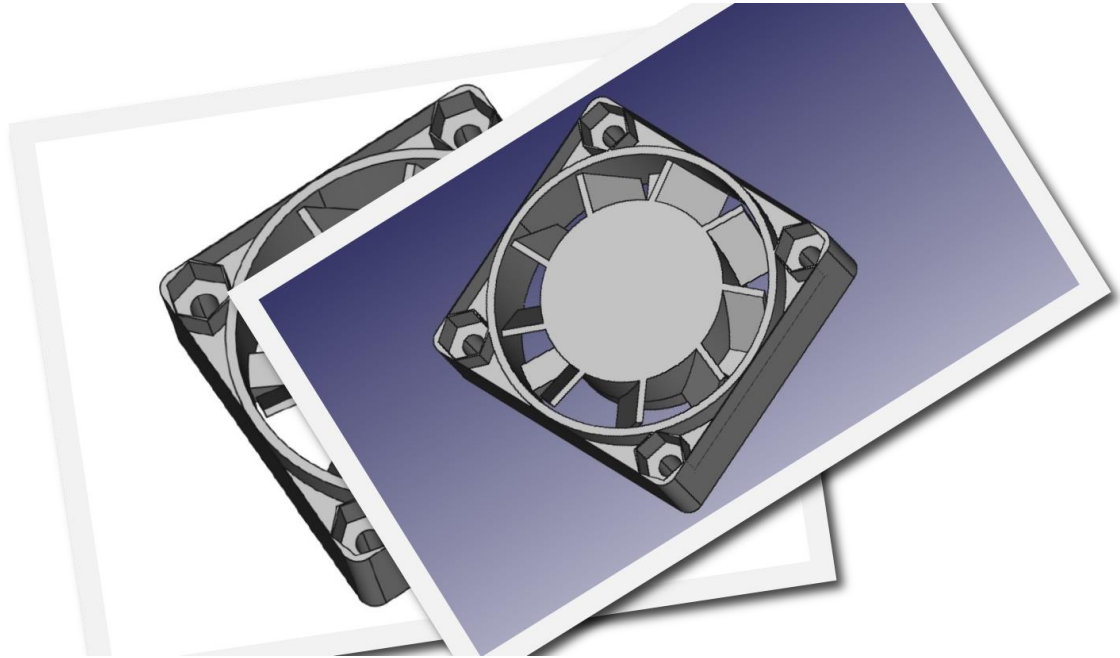


Figura 117. Cooler 12 V.

MODELO	HL9225S12H
MARCA	FENGHAO
LENGTH	92
WIDTH	92
ALTURA	25
VOLTAJE	12V
CURRENT	0.20 ^a
CABLE	2

TABLA 24. Datasheet Cooler 12 V.(Elecok, 2022)

DATASHEET SWITCH DE BLOQUEO

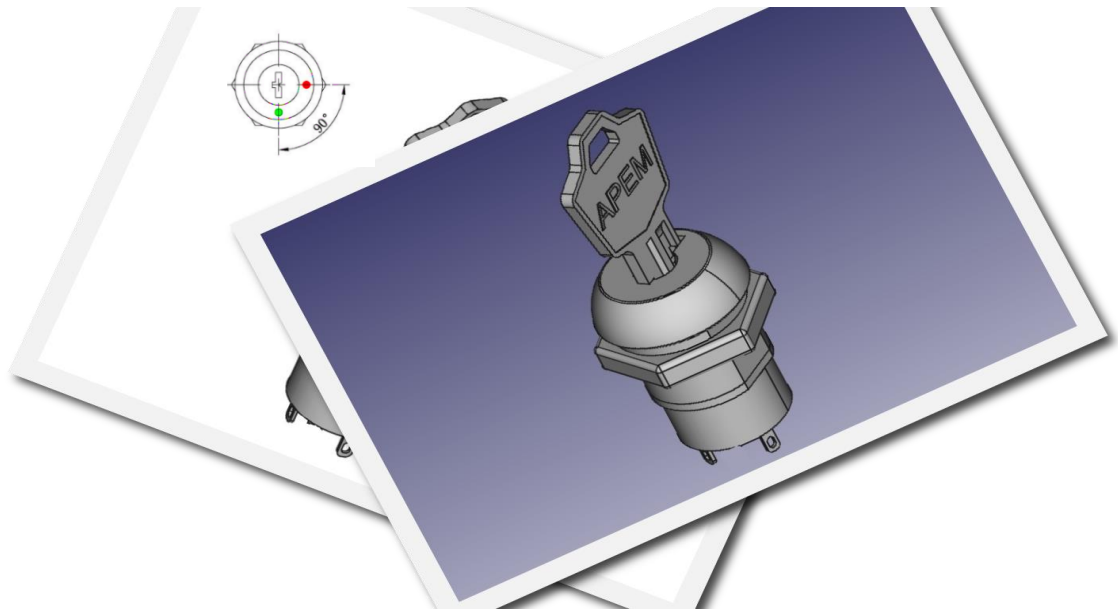


Figura 118. Switch de Bloqueo.

Clasificación:	0,5 a 250v ca/0,5 a 24v cc
Material:	acero inoxidable, plástico.
Pin:	2 pines
Clave:	llave extraíble en ambas posiciones
Posición:	2 posiciones
Resist. De aislamiento:	$\geq 500\text{m}\omega(500\text{vac})$
Resist. De contacto:	$\leq 100\text{ m}\omega$
Soporta voltaje	ca 500v(50hz) 1min 2ma
Temp. De funcionamiento:	25 °C ~ + 85 °C llave eléctrica

TABLA25. Datasheet Switch de Bloqueo. (Daiertek, 2022)

DATASHEET SWITCH

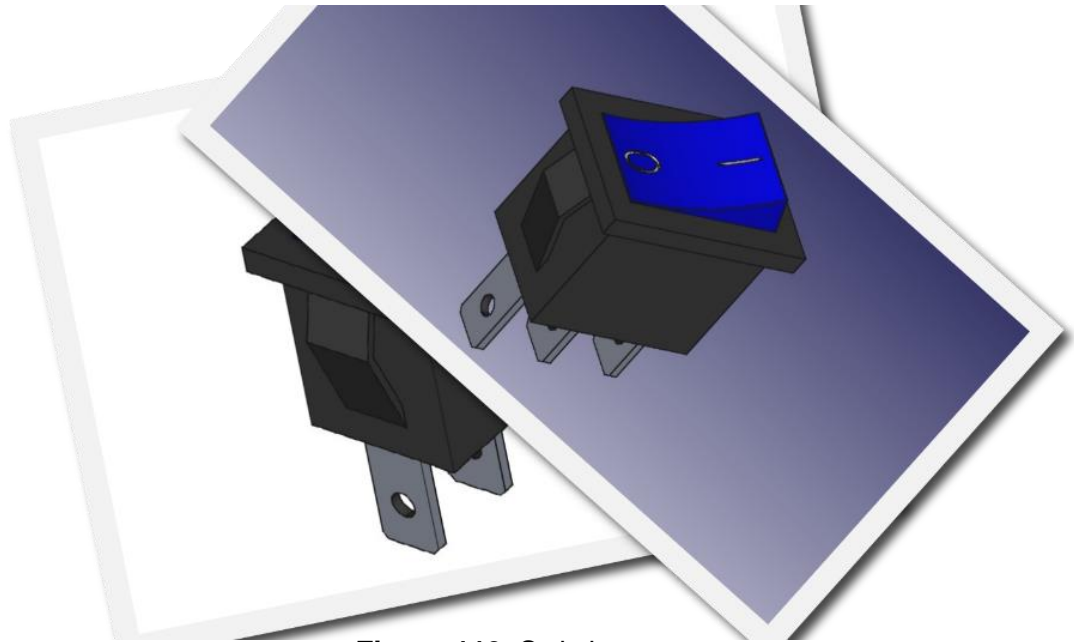


Figura 119. Switch.

Indicador:	Luz de neón roja integrada.
Pin:	un solo polo (SPST), terminales de 3 pines.
Resistencia de aislamiento:	500VDC 100MΩ min.
Resistencia de contacto:	35mΩ max.
Resistencia dieléctrica:	1500VAC, 1 minuto.
Temperatura de operación:	-25°C- +85°C
Vida eléctrica:	≥ 10.000 ciclos.
Voltaje de la lampara:	ca 220V.
Voltaje nominal:	6A 250 VAC, 10A 125VAC.

TABLA26. Datasheet Switch.(Daier, 2017)

ANEXO D. PROTOTIPOS

PRIMER PROTIPO

El primer diseño está basado en un termo donde se monitorea por un sensor que dará respuesta a los cambios de temperatura al momento del uso del mismo, estos cambios se visualizan en una pantalla LCD.

En la parte superior se encuentra la tapa que contiene las celdas peltier TEC- 12706 que es un termoconductor para refrigerar.

Para poder enfriar el agua con ayuda de un cooler se pretende analizar como se comporta al momento de enfriar y ver si se podría aprovechar el PID con este primer prototipo.

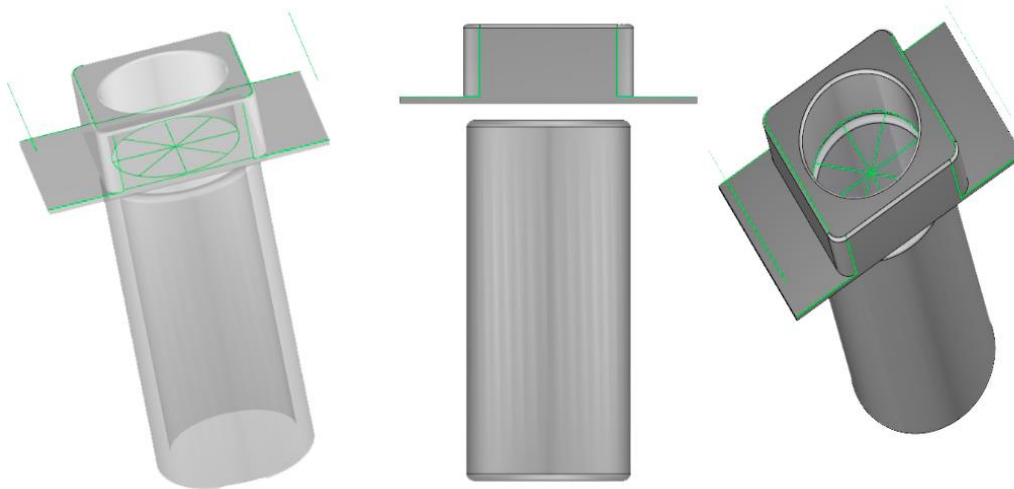


Figura 120. Diseño de termo portátil 1 en software inventor.

LISTA DE MATERIALES:

- Hiletgo relé estado sólido hiletgo ssr-40da, 250 v.
- Celdas peltier tec1-12706.
- Display lcd 16x2.
- Arduino nano.
- Protoboar.
- Fuente de poder 12v- 30 - 360w.
- Monster Moto Shield Vnh2sp30.
- Sensor Ds18b20.
- Pasta Térmica.
- cooler 12v.
- Vaso Precipitado De 500ml.
- Cables cobre duplex calibre 12, 14,16.
- calentador eléctrico ovente CH112.
- Potenciómetro.



Figura 121. Materiales termo portátil 1.

Las pruebas del prototipo 1 se tomó en cuenta el consumo de las celdas peltier de 60w por cada una, viendo que su consumo es alto y la corriente máxima es de 6A, por cada celda para poder enfría al 100% de su capacidad, dicho amperaje fue dado por la fuente de poder utilizada en las pruebas, para el tiempo de CP se regula la temperatura deseada con un potenciómetro de manera manual.

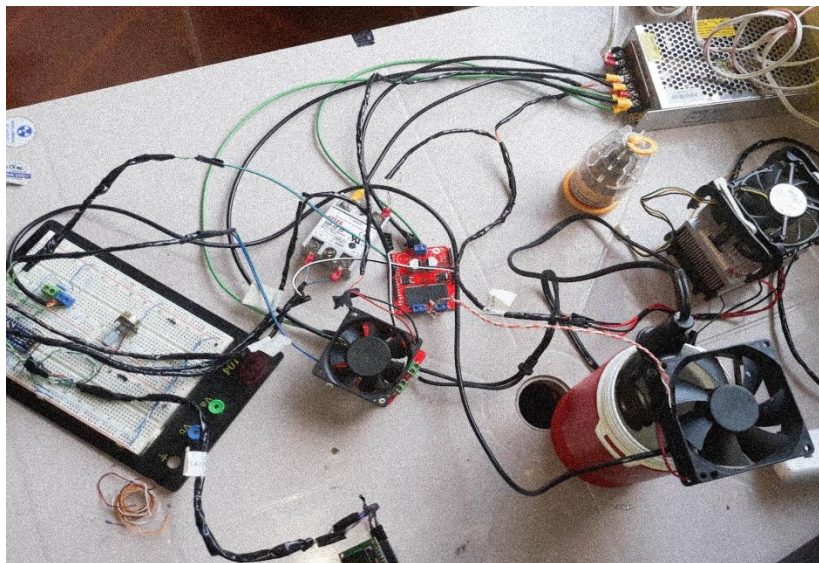


Figura 122. Conexiones termo portátil 1.

Al momento de tapar el vaso para poder enfriar el líquido suministrado, no era muy eficaz ya que demoraba mucho, aunque se modificó varias veces el PID no se tuvo los resultados deseados y se concluye que no se aprovecha de manera correcta las celdas peltier para poder enfriar el líquido del envase.

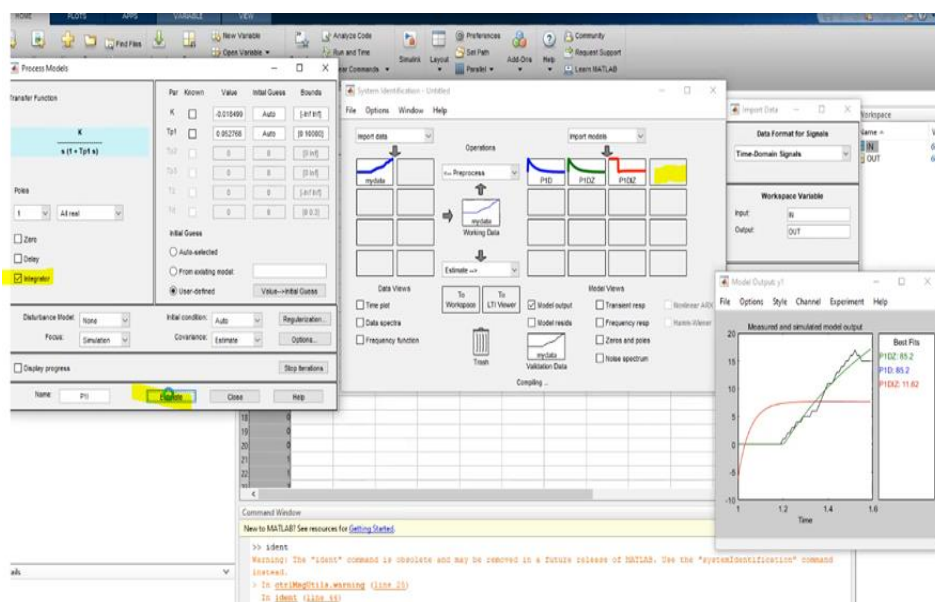


Figura 123. Herramienta ident Software Matlab.

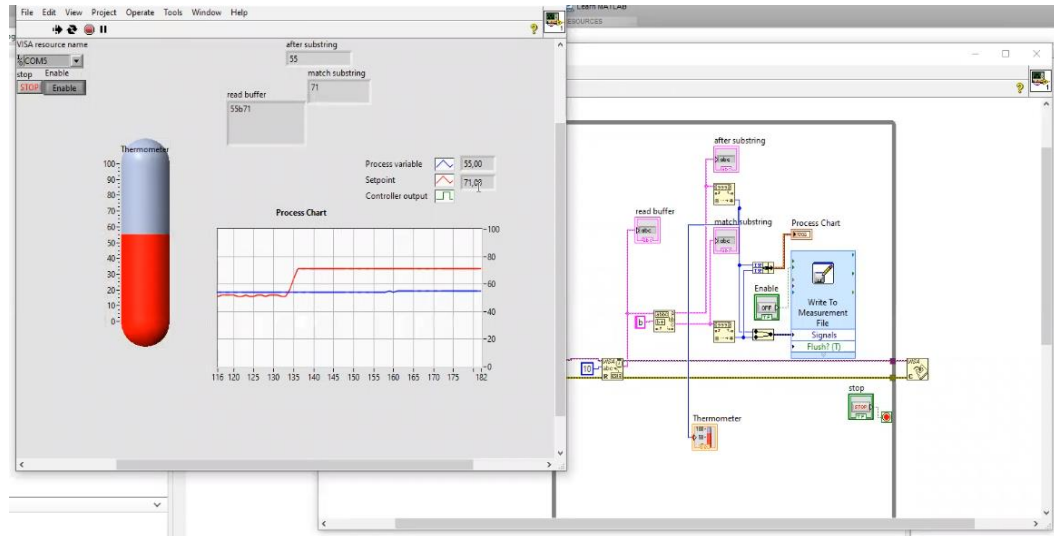


Figura 124. Prueba comunicación serial software LabVIEW.

El LCD muestra la temperatura en tiempo real, mientras que CSP es el set point el cual le dice al calentador a que temperatura quiero llegar y el output es la comunicación serial entre la tarjeta Arduino y el LabView para guardar dichos datos obtenidos de manera real en un Excel y ver su comportamiento para ajustar mejor el PID On-Off deseado.



Figura 125. Datos visual LCD 16x2.

El circuito realizado de manera esquemática es para visualizar las conexiones que corresponde al proyecto del termo portátil 1.

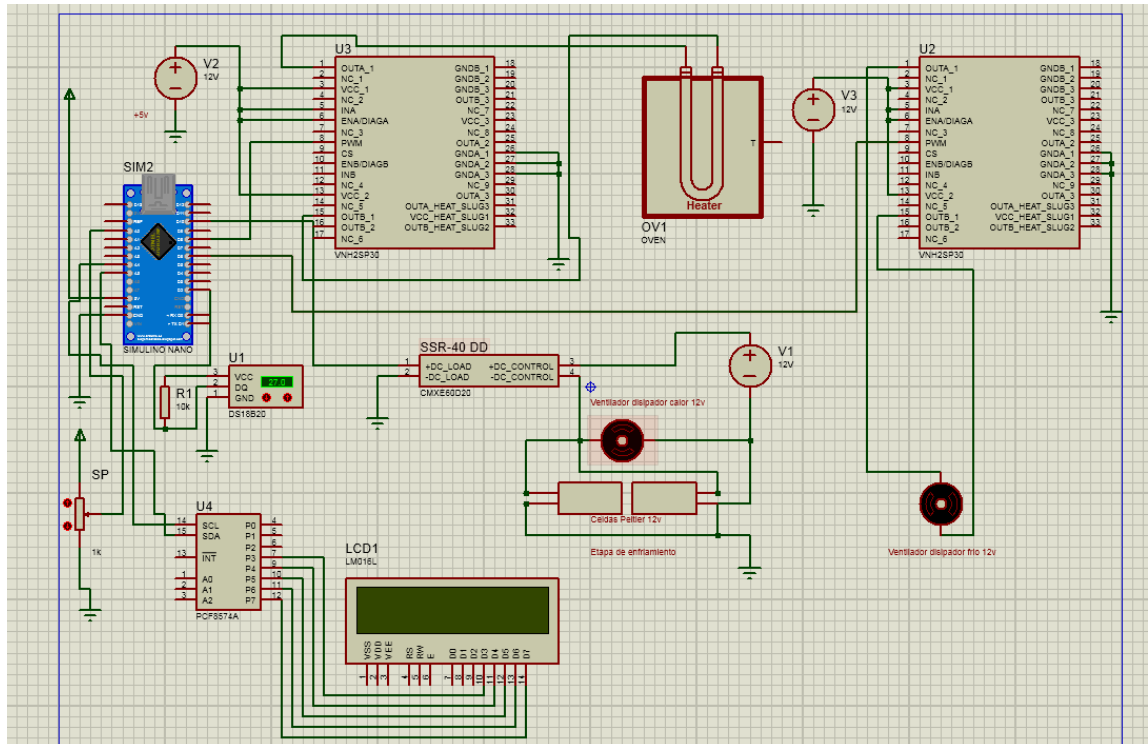


Figura 126. Esquemático proteus termo portátil 1.

SEGUNDO PROTOTIPO

En el segundo prototipo muestra la señal de entrada de referencia con respecto al tiempo, esta será la temperatura deseada y el proporcional con respecto al tiempo donde nuestra una constante, siendo un sistema retroalimentado para comparar la temperatura deseada con la temperatura del sensor, estas se van a restar para obtener la variable del error dando paso al sistema PID y estabilizar la planta.

Se va a usar un vaso precipitado de 1000ml, estos envases soportan $-250\text{ }^{\circ}\text{C} / 270\text{ }^{\circ}\text{C}$.

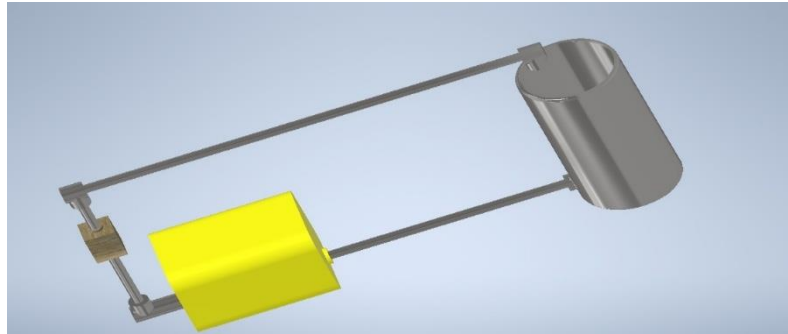


Figura 127. Prototipo 2 en software inventor.

El líquido dentro del envase va a pasar por los módulos peltier las veces que sean necesario para lograr nuestro CP deseado con ayuda de la bomba brushless water pump que trabaja con 12v DC.



Figura 128. Bomba Brushless Water Pump.

lista de Materiales:

- Celdas peltier tec1-12706.
- Display lcd 16x2.
- Arduino nano.
- Protoboar.
- Fuente de poder 12v- 30 - 360w.
- Monster Moto Shield Vnh2sp30.
- Sensor Ds18b20.
- Pasta Térmica.
- cooler 12v.
- Cables cobre duplex calibre 12, 14,16.

- DC12V Brushless Water Pump.

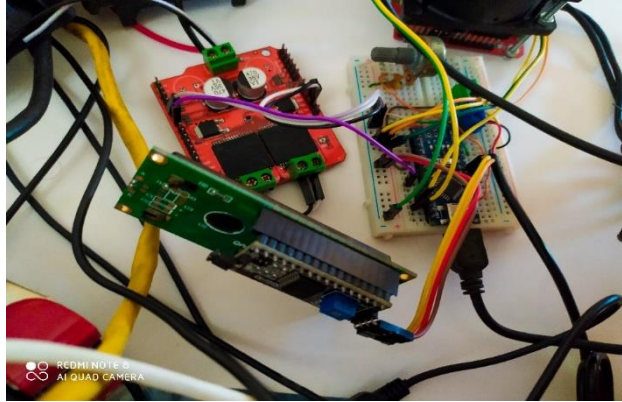


Figura 129. Conexiones Prototipo 2.

Para esta idea se enfoca con el sistema de un dispensador de agua ya que usa un sistema de refrigeración.

Las pruebas hechas en el segundo prototipo se debe enfocar más la investigación en el material del envase para tener un mejor resultado.



Figura 130. Envase de prueba.

TERCER PROTOTIPO

En este sistema va constar de dos celdas peltier independientes, el primero es para enfriar y el otro es para calentar están conectadas en paralelo, cuando se presiona el primer switch va a energizar el sistema y luego se energiza para seleccionar la posición del segundo switch para enfriar o calentar, con el potenciómetro le damos el CP deseado donde luego se introduce el vaso con el líquido para ver como se comporta el sistema de control.

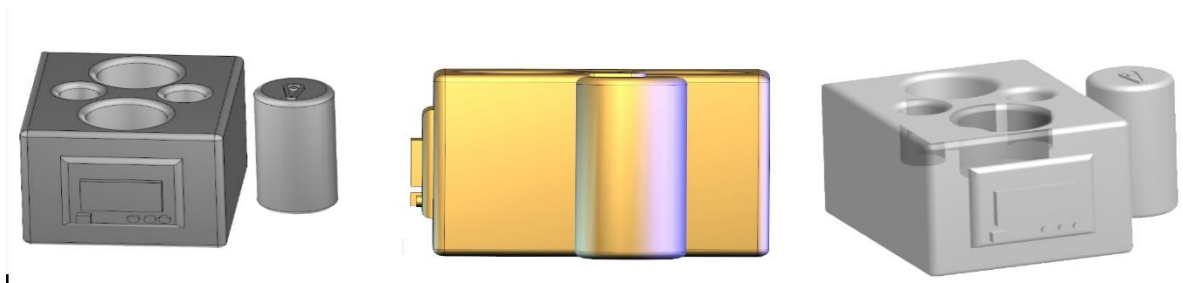


Figura 131. Prototipo 3 en software inventor.

El problema de este diseño es que ocupaba mucho espacio y los resultados obtenido no fueron los mejores, ya que se sobre calentaba las celdas peltier por no tener una buena disipación de flujo de aire.



Figura 132. Caja de madera Prototipo 3.

CUARTO PROTOTIPO

Se diseña un envase metálico para hacer dos retroalimentaciones de flujo, la cual almacena el agua en dos reservorios de aluminio que va a pasar por dos tuberías de cobre individual cada una va a tener una bomba Brushless Water Pump para el líquido que se mantiene en constante movimiento y estos reservorios van a ir en cada cara de las celdas peltier donde un reservorio va a calentar mientras el otro va a enfriar.

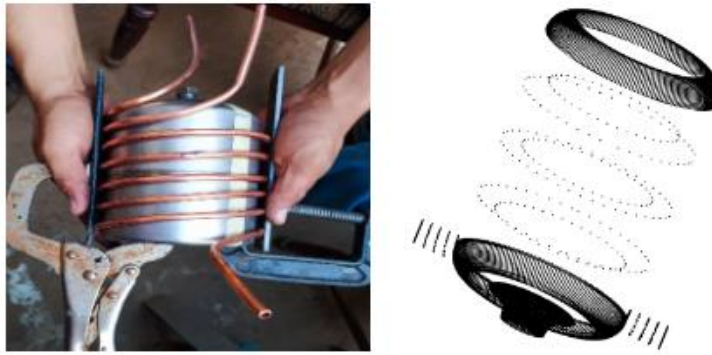


Figura 133. Diseño envase metálico prototipo 4.

El sistema retroalimentado por cada tubería va a ir alrededor del envase, donde se tiene un switch de encendido y apagado otro para enfriar o calentar este tiene contacto con el envase con ambas tuberías de cobre para establecer la transferencia de energía (frío o calor) al momento que elija el usuario.

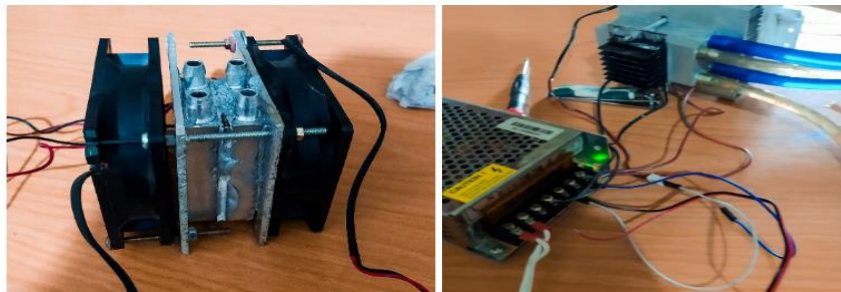


Figura 134. Reservorios de aluminio.

El CP se manipula por un potenciómetro donde se establece los grados centígrados, si se desea enfriar o calentar, se acciona el switch de la bomba requerida para que mantenga el flujo del agua en movimiento y transferir de manera constante la transferencia de energía en el líquido dentro del envase.

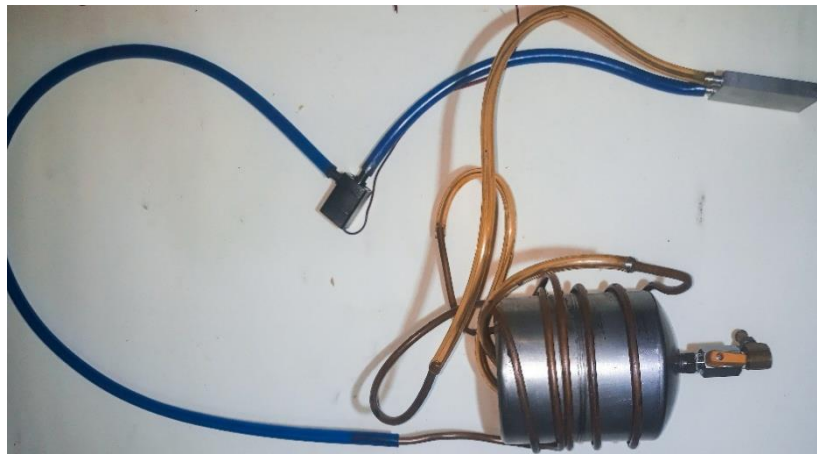


Figura 135. Sistema retroalimentado.

Los parámetros estables dentro de las pruebas que se quiere llegar son -15°C a -20°C para enfriar mientras que para calentar será de 30°C a 40°C .

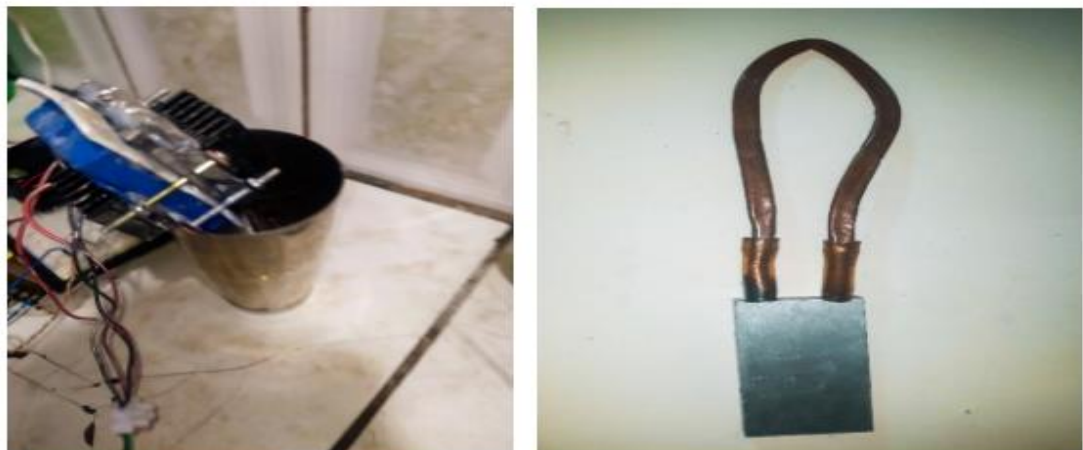


Figura 136. Prueba prototipo 4.

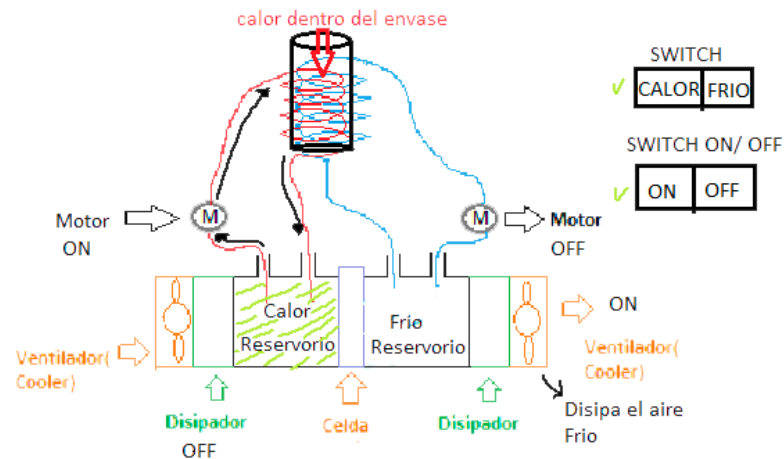


Figura 137. Conexiones prototipo 4.

Este análisis se basa en los equipos de computadora porque este tipo de sistema es similar y usa el sistema que utiliza un gas para hacer que la temperatura del procesador no se caliente y tenga una temperatura ideal para poder trabajar en la computadora, por ello se ha realizado este mismo sistema con un líquido o fluido para ver como se comporta la transferencia de energía, donde se concluye los resultados no fue favorable la de transferencia en el envase donde se considera que demora mucho tiempo en poder enfriar o calentar el recipiente porque es de metal muy grueso.

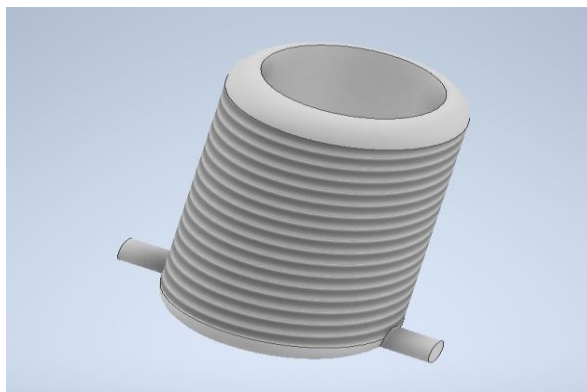


Figura 138. Prototipo 4 en software inventor.

ANEXO E. DISEÑO FINAL

La impresión posterior del envase final de acero inoxidable se realizó una prueba en un recipiente creado en 3D impreso PLA.

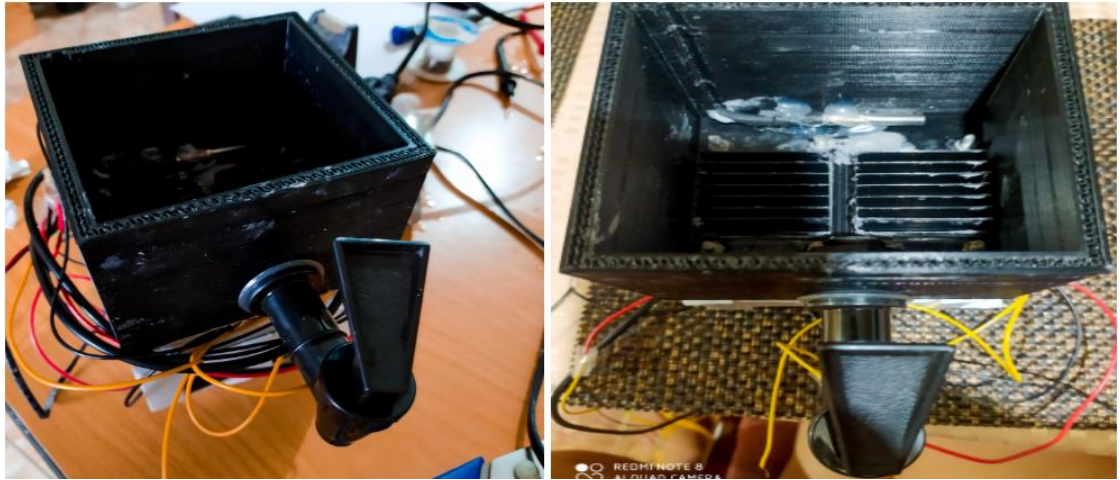


Figura 139. Envase de PLA.

El envase final fue creado en AutoCAD con medidas específicas de ancho 9.5 cm x10.2 cm de largo hecho de acero inoxidable.



Figura 140. Envase final AutoCad.

La placa PBC fue creada en el software Proteus esta placa es necesaria para poder reducir conexiones con los módulos y actuadores a utilizar:

- módulo Bluetooth hc-06.
- módulo Arduino Nano.
- módulo monster shield VNH2SP30.

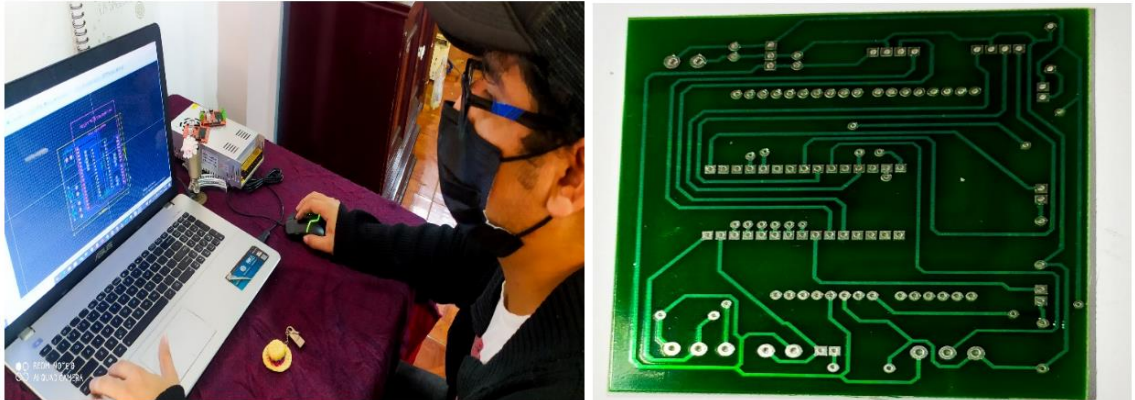


Figura 141. Diseño PBC.

Los diseños creados para imprimir, se utiliza 3 softwares Inventor, FreeCad para diseños en 3D y ultimaker cura para imprimir los diseños creados en 3D.

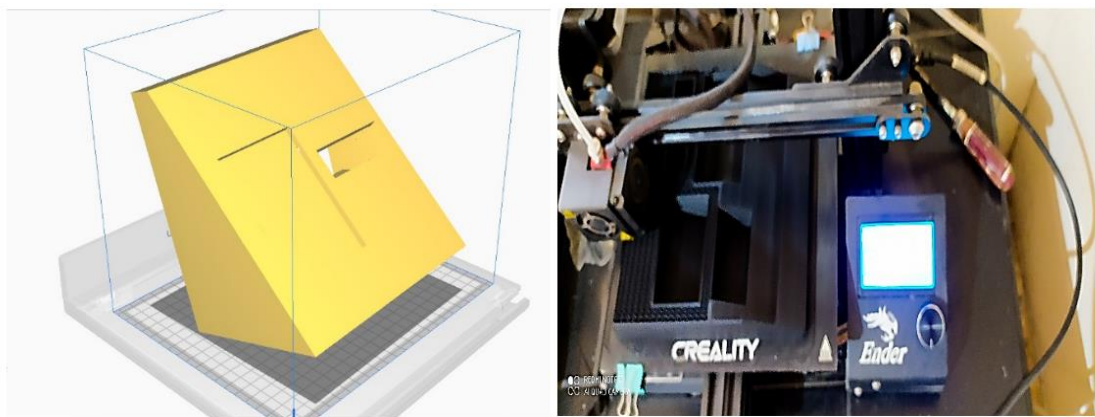


Figura 142. Impresión PLA software ultimaker cura.

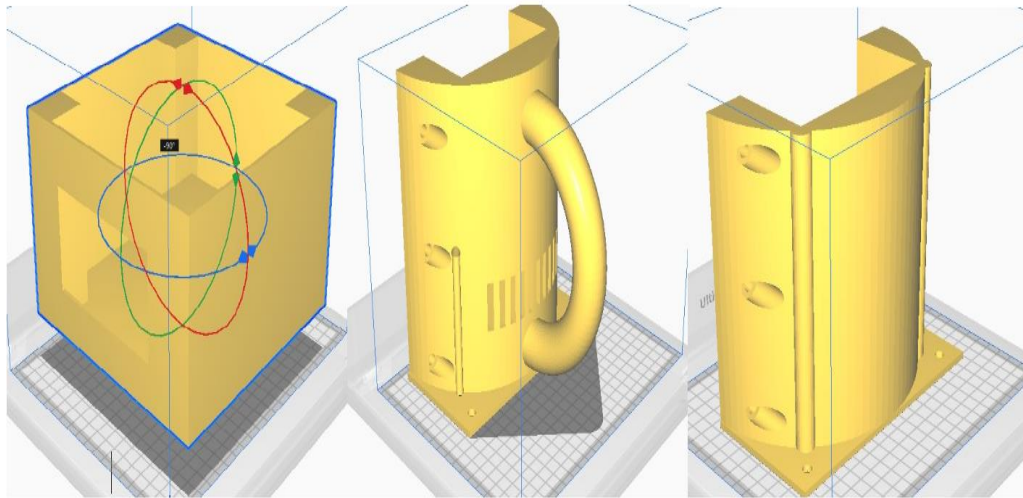


Figura 143. Impresión PLA software ultimaker cura.

La programación de Arduino es la más importante ya que va a contener los cálculos matemáticos, este código fue modificado varias veces hasta tener una programación optima.

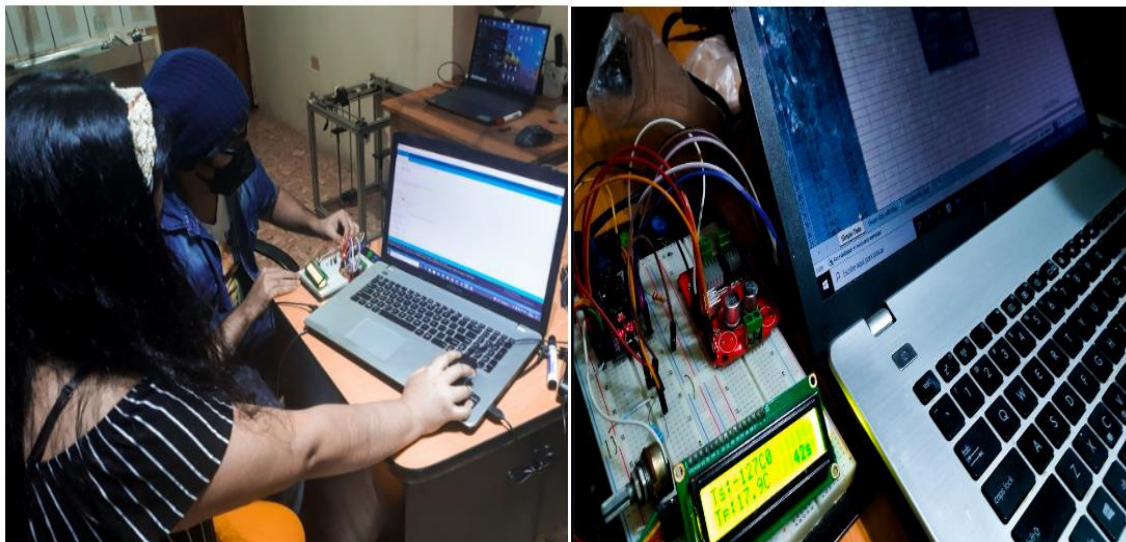


Figura 144. Foto de Programación Arduino.

La aplicación se creó en App Inventor para ayudar enfriar al máximo la planta.



Figura 145. Programación APK en APP Inventor.

Paleta. -Hay varios elementos estos se arrastran al espectador o visor para añadir a la aplicación. La Interfaz de usuario permitirá crear:

Botones, campo de texto, etiquetas, visor web, visor de lista, selector de fecha, selector de hora, casilla de verificación, selector de lista, campo de contraseña, notificador, imagen, despegable.

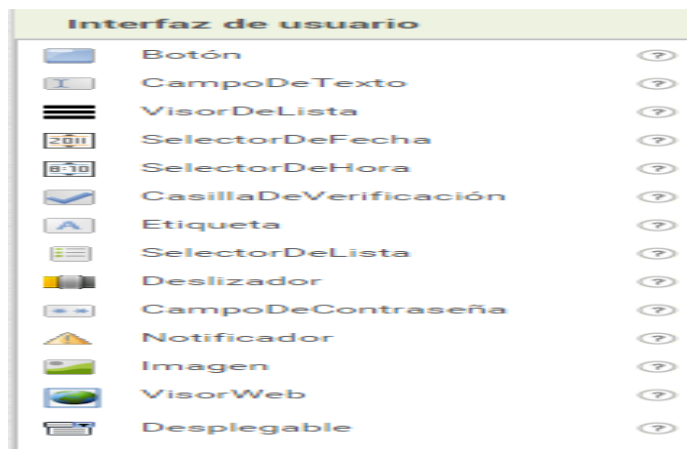


Figura 146. Segmento Interfaz De Usuario.(appinventor, 2022)

Disposición. - Permite organizar los elementos o gráficos en forma:

- Tabla tabular o matriz.
- Tabla vertical.
- Tabla horizontal

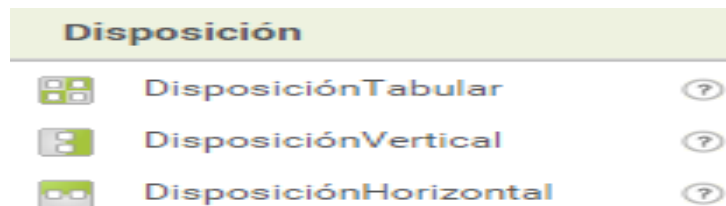


Figura 147. Paleta Disposición. (appinventor, 2022)

Medios. -Se podrá elegir elementos como: reproductor, sonido, texto a voz, traductor Yandex, reproductor de video, grabador, cámara, reconocimiento de voz, selector de imagen y una grabadora de sonido.



Figura 148. Paleta Medios.(appinventor, 2022)

Conectividad. - Estos elementos tienen funciones como conectividad mediante Bluetooth.

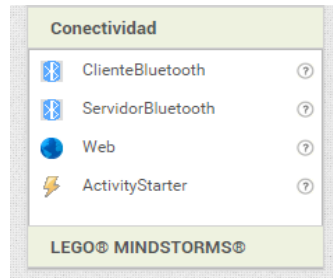


Figura 149. Paleta Conectividad.(appinventor, 2022)

Editor De Bloques

Bloques. - El Complemento de bloques es para establecer el comportamiento de la aplicación.

1.-Bloques integrados construidos en los cajones sirve para encontrar bloques de comportamientos generales es posible que desee agregar a su aplicación y arrastrarlos al espectador o visor bloques.

2.-Cajones específicos de los componentes se utiliza para encontrar bloques componentes específicos y arrastrarlos al espectador o visor de bloques.

- **Eventos son de color café.** - Es lo que se realizará cuando algo sucede.
- **Propiedades es de color verde.** - Entable una propiedad después del evento.
- **Llamadas son de color morado.** - Llama a un componente que realice una opción.

3.-Espectador O Visor: Bloques de arrastre se llevarán al espectador para construir relaciones o los comportamientos de la App.

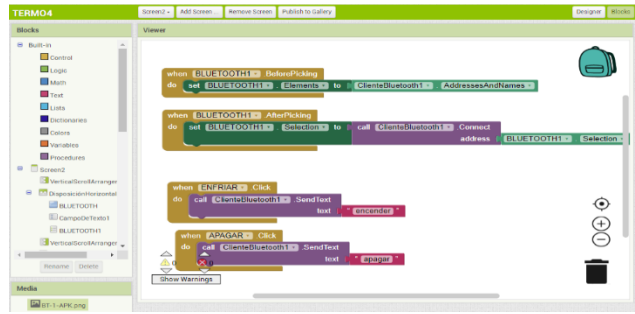


Figura 150. Editor De Bloques.(appinventor, 2022)

Bloque de Control

- **Si Entonces.** - Prueba una condición obtenida, si es verdadera se realizará.



Figura 151. Si Entonces.(appinventor, 2022)

- **Abrir Otra Pantalla.** -Abre la pantalla o Screen con el nombre proporcionado.



Figura 152. Abrir Otra Pantalla.(appinventor, 2022)

Bloque De Texto

- **Cadena De Texto.** - En esta se puede contener números, letra u otros caracteres especiales.



Figura 153. Cadena De Texto.(appinventor, 2022)

ANEXO F. CÓDIGO ARDUINO



```

PID
#include <OneWire.h>           //libreriasensor ds18b20
#include <DallasTemperature.h> //libreria sensor ds18b20
#include <Wire.h>              //libreria lcd
#include <LiquidCrystal_I2C.h> //libreria lcd
#include <SoftwareSerial.h>    //establecer conexion serial para el modulo bluetooth

SoftwareSerial Serial_2 (2, 3); // Crea nueva conexion- Pin2(RX) a TX y Pin3(TX) a RX del BT
String Mensaje;                // Variable de cadena de caracteres para almacenar el mensaje
//Crear el objeto lcd dirección 0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2); //
OneWire ourWire(12);           //Se establece el pin 12 como bus OneWire
DallasTemperature sensors(&ourWire); //Se declara una variable u objeto para nuestro sensor
bool manual=true;             //variable para elegir modo manual y no por bluetooth
#define cold 13// led encendido si esta frio
#define BLUE 11
#define APAGADO 0
#define ENCENDIDO 1
#define CS_THRESHOLD 15 // Definition of safety current (Check: "1.3 Monster Shield Example").
//Peltier 1
#define P1_A_PIN 7
#define P1_B_PIN 8
//peltier 2
#define P2_A_PIN 9
#define P2_B_PIN 10
#define PWM_P1 5
#define PWM_P2 6
#define EN_PIN_1 A0 //salida para habilitar salida 1 de la shield
#define EN_PIN_2 A1 //salida para habilitar salida 2 de la shield

short velocidad = 0; //default motor speed
unsigned short Estado_peltier = APAGADO;
int tiempo = 0;
float sensor=29.0; // tseratura del sensor default

```

Figura 154. Código Arduino Parte 1.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

PID §
float r1=100.0; //Referencia temperatura usuario default
volatile float u=0.0,u_1=0.0; //Acción de Control
byte Ts = 84; //Periodo de muestreo
//Parámetros del PID
float kp,ti,td;
float q0,q1,q2;
volatile float e=0.0,e_1=0.0,e_2=0.0;
float k=-0.1271,tau=1667,L=20; //Parámetros del Modelo del sistema
float theta= L +Ts/2;
// tau es la constante de tiempo que se obtiene con el tiempo de estabilizacion dividido para 4

/* fin de parametros para el PID*/

void setup()
{
  pinMode(P1_A_PIN, OUTPUT);
  pinMode(P1_B_PIN, OUTPUT);

  pinMode(P2_A_PIN, OUTPUT);
  pinMode(P2_B_PIN, OUTPUT);

  pinMode(PWM_P1, OUTPUT);
  pinMode(PWM_P2, OUTPUT);

  pinMode(EN_PIN_1, OUTPUT);
  pinMode(EN_PIN_2, OUTPUT);

  pinMode(BLUE, INPUT); // PIN DE ENTRADA PARA CONTROL MANUAL O Bluetooth controla led bluetooth D1

  pinMode(cold,OUTPUT); //Led "frio" como salida
  digitalWrite(cold,LOW);
  Serial_2.begin(9600); // Iniciamos el puerto nuevo Serial_2 (BT) a 9600 Baudios
  Serial.begin(115200); // Initiates the serial to do the monitoring
}
```

Figura 155. Código Arduino Parte 2.


```

PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

PID $

pinMode(13,1);      // Configura Pin 13 como Salida para led manual D2
pinMode(4,1);      // Configura Pin 4 como Salida para led frio D3

sensors.begin();  //Se inicia el sensor
lcd.init();
//Encender la luz de fondo.
lcd.backlight();

//*****//
//*****      SINTONIA POR ZIEGLER y NICHOLS      *****//
//*****//
//kp=((1.2*tau)/(k*theta))/2;
kp=(1.2*tau)/(k*theta);
ti=2.0*theta;
td=0.5*theta;

//*****//
//*****      PID DIGITAL      *****//
//*****//
// Calculo do controle PID digital
q0=kp*(1+Ts/(2.0*ti)+td/Ts);
q1=-kp*(1-Ts/(2.0*ti)+(2.0*td)/Ts);

q2=(kp*td)/Ts;

}

|

```

Figura 156. Código Arduino Parte3.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

PID $

void loop()
{

  manual= digitalRead(11);

  if(manual == false)
  {
    digitalWrite(13,true); // Apaga LED en Pin 13 led manual
    r1= analogRead(A6)*(6.00 / 1023.00)+23; //lee potenciómetro para saber temperatura deseada 16 temperatura minima
    //analogRead(A6)*(6.00 / 1023.00) me bota un valor entre 0 a 6, luego sumo el +23 para que el potenciómetro me indique valores entre 23 y 23+6=29
  }

  else{
    digitalWrite(13,false); // Apaga LED en Pin 13

    // definir temperatura de usuario cuando este en modo BT

    while (Serial_2.available()) { // Mientras que lleguen caracteres por el puerto serial_2
      delay(5);
      char c = Serial_2.read(); // Lee los caracteres uno a uno en la variable c
      Mensaje += c; // Almacena la suma de caracteres en el mensaje
    }

    if (Mensaje.length()>0){
      //Serial.println( Mensaje); // envia mensaje al PC

      if (Mensaje == "encender"){
        r1=10; //10 para ver cual es la temperatura minima en lazo abierto
        Serial.println( "Monster prendido ");
      }

      if (Mensaje == "apagar"){
        r1=29;
        Serial.println( "Monster apagado ");
      }
    }
  }
}
```

Figura 157. Código Arduino Parte 4.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
PID $
    Serial.println( "Monster apagado " );
}
}

Mensaje=""; // Borramos el mensaje ya enviado al PC
}


tiempo=millis()/1000;

sensors.requestTemperatures(); //Se envia el comando para leer la temperatura
sensor= sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor

PID(); //en lazo abierto velocidad=255;

if(e>=0){
    digitalWrite(4,0);
    Apagar();
    digitalWrite(EN_PIN_1, LOW); // habilita peltier 1
    digitalWrite(EN_PIN_2,LOW); // habilita peltier 2
}
else {
    digitalWrite(4,1);
    Encender();
    digitalWrite(EN_PIN_1, HIGH); // habilita peltier 1
    digitalWrite(EN_PIN_2, HIGH); // habilita peltier 2
}
```

Figura 158. Código Arduino Parte 5.



```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
PID $

Serial.print(tiempo);
Serial.print(",");
Serial.print(sensor);
Serial.print(",");
Serial.print(e);
Serial.print(",");
Serial.println(u);
mostrar_lcd( sensor, r1, tiempo);
//delay(1000); // delay para observar los cambios en el puerto serial de forma mas pausada
//este delay debe ser comentado (eliminado) en el programa final.

}

void Apagar()
{
  Estado_peltier = APAGADO;
  motorGo(Estado_peltier);
}

void Encender()
{
  Estado_peltier = ENCENDIDO;
  motorGo(Estado_peltier);
}

void motorGo( uint8_t direct) //Function that controls the variables: motor(0 ou 1), direction (cw ou ccw) e pwm (entra 0 e 255);
{
```

Figura 159. Código Arduino Parte 6.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

PID $

if(direct == ENCENDIDO)
{
  digitalWrite(P1_A_PIN, HIGH);
  digitalWrite(P1_B_PIN, LOW);
  delay(10);
  digitalWrite(P2_A_PIN, HIGH);
  digitalWrite(P2_B_PIN, LOW);
}
else
{
  digitalWrite(P1_A_PIN, LOW);
  digitalWrite(P1_B_PIN, LOW);
  delay(10);
  digitalWrite(P2_A_PIN, LOW);
  digitalWrite(P2_B_PIN, LOW);
}

analogWrite(PWM_P1, velocidad);
delay(10);
analogWrite(PWM_P2, velocidad);
}

void PID(void)
{
  e=(r1-sensor);

  //Serial.print("error ");
  //Serial.println(e);
  // delay (10);
  // Controle PID
  u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto
```

Figura 160. Código Arduino Parte 7.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

PID $

//u=100; //prueba en lazo abierto

if (u >= 100.0)          //Saturo la accion de control 'uT' en un tope maximo y minimo
  u = 100.0;

if (u <= 0.0 || r1==0)
  u = 0.0;

//Serial.print("U ");
//Serial.println(u);
//delay(10);

//Retorno a los valores reales
e_2=e_1;
e_1=e;
u_1=u;

//La accion calculada la transformo en PWM

velocidad = 2.55*u;
//analogWrite(velocidad,map(u, 0,100, 0,255));
//Serial.print("velocidad ");
//Serial.println(velocidad);
//delay(10);

}

void mostrar_lcd(float x1, float x2, int x3){
  // Cursor en la primera posición de la primera fila
  lcd.setCursor(0,0);
  lcd.print("ts:");
  lcd.print(x1,1); //temperatura sensor
  lcd.setCursor(9,0);
  lcd.print("C"); // "337" -> ""
  lcd.setCursor(0,1);
}
```

Figura 161. Código Arduino Parte 8.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
PID $
  //Serial.print("U ");
  //Serial.println(u);
  //delay(10);

  //Retorno a los valores reales
  e_2=e_1;
  e_1=e;
  u_1=u;

  //La accion calculada la transformo en PWM

  velocidad = 2.55*u;
  //analogWrite(velocidad,map(u, 0,100, 0,255));
  //Serial.print("velocidad ");
  //Serial.println(velocidad);
  //delay(10);

}

void mostrar_lcd(float x1, float x2, int x3){
  // Cursor en la primera posición de la primera fila
  lcd.setCursor(0,0);
  lcd.print("ts:");
  lcd.print(x1,1); //temperatura sensor
  lcd.setCursor(9,0);
  lcd.print("C"); // "337" -> ""
  lcd.setCursor(0,1);

  lcd.print("Tp:");
  lcd.print(x2,1);//1 decimal temperatura potenciómetro
  lcd.print("C"); // "337" -> ""
  // Cursor en la 11ª posición de la 2ª fila
  lcd.setCursor(12,1);
  lcd.print(x3);
  lcd.print("s");
}

```

Figura 162. Código Arduino Parte 9.

ANEXO G. PRUEBAS.

PRUEBA 1

La primera prueba realizada es en lazo abierto estos datos corresponden a la planta donde la primera fila que muestra el tiempo en segundo y la segunda fila la temperatura.

E	C	D	E	F	G
Monster prendido					
1	29,69	-6,69	100		
2	29,69	-6,69	100		
3	29,69	-6,69	100		
4	29,69	-6,69	100		
5	29,69	-6,69	100		
6	29,75	-6,75	100		
7	29,75	-6,75	100		
8	29,75	-6,75	100		
9	29,75	-6,75	100		
10	29,75	-6,75	100		
11	29,75	-6,75	100		
12	29,75	-6,75	100		
13	29,75	-6,75	100		
14	29,75	-6,75	100		
15	29,75	-6,75	100		
16	29,75	-6,75	100		
17	29,75	-6,75	100		
18	29,75	-6,75	100		
19	29,81	-6,81	100		
20	29,81	-6,81	100		
21	29,81	-6,81	100		
22	29,81	-6,81	100		
23	29,81	-6,81	100		
24	29,81	-6,81	100		
25	29,81	-6,81	100		
26	29,81	-6,81	100		

Aquí se analiza el comportamiento de la planta en lazo abierto.

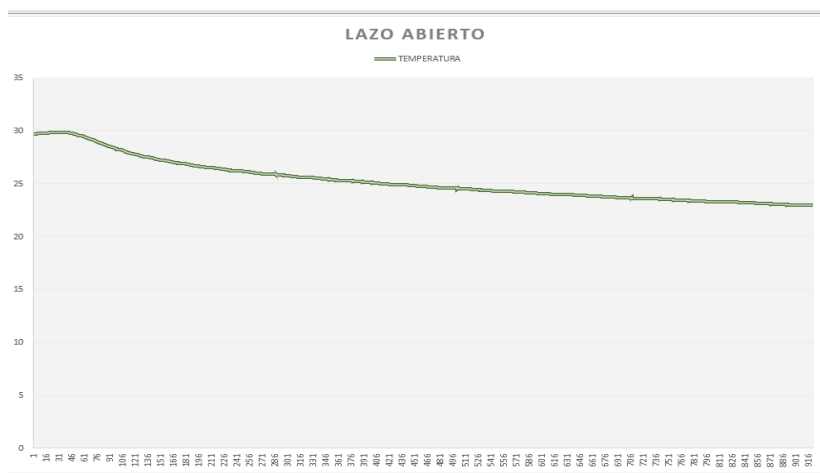


Figura 164. Prueba 1 Gráfica Lazo Abierto.

En el LCD se observa que en 23.9°C la planta tiene 719 segundos que equivale a 11 minutos aun terminando establecer el set point máximo, al llegar a la 23.0°C analizando el tiempo de encendido se establece su temperatura máxima en el tiempo de 15.35 minutos.



Figura 165. Prueba 1 Planta Lazo Abierto.


Código prueba 1

```
Lazoabierto
#include <OneWire.h> //libreriasensor ds18b20
#include <DallasTemperature.h> //libreria sensor ds18b20
#include <Wire.h> //libreria lcd
#include <LiquidCrystal_I2C.h> //libreria lcd
#include <SoftwareSerial.h> //establecer conexion serial para el modulo bluetooth

SoftwareSerial Serial_2 (2, 3); // Crea nueva conexion- Pin2(RX) a TX y Pin3(TX) a RX del BT
String Mensaje; // Variable de cadena de caracteres para almacenar el mensaje
//Crear el objeto lcd dirección 0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2); //
OneWire ourWire(12); //Se establece el pin 12 como bus OneWire
DallasTemperature sensors(&ourWire); //Se declara una variable u objeto para nuestro sensor
bool manual=true; //variable para elegir modo manual y no por bluetooth
#define cold 13// led encendido si esta frio
#define BLUE 11
#define APAGADO 0
#define ENCENDIDO 1
#define CS_THRESHOLD 15 // Definition of safety current (Check: "1.3 Monster Shield Example").
//Peltier 1
#define P1_A_PIN 7
#define P1_B_PIN 8
//peltier 2
#define P2_A_PIN 9
#define P2_B_PIN 10
#define PWM_F1 5
#define PWM_F2 6
#define EN_PIN_1 A0 //salida para habilitar salida 1 de la shield
#define EN_PIN_2 A1 //salida para habilitar salida 2 de la shield

short velocidad = 0; //default motor speed
unsigned short Estado_peltier = APAGADO;
int tiempo = 0;
float sensor=29.0; // tseratura del sensor default
```

Figura 166. Prueba 1 Código Arduino Parte1.



```
float r1=100.0; //Referencia temperatura usuario default
volatile float u=0.0,u_1=0.0; //Acción de Control
byte Ts = 12.5; //Periodo de muestreo
//Parámetros del PID
float kp,ti,td;
float q0,q1,q2;
volatile float e=0.0,e_1=0.0,e_2=0.0;
float k=-0.6,tau=175,L=48; //Parámetros del Modelo del sistema
float theta= L +Ts/2;
// tau es la constante de tiempo que se obtiene con el tiempo de estabilizacion dividido para 4

/* fin de parametros para el PID*/

void setup()
{
  pinMode(P1_A_PIN, OUTPUT);
  pinMode(P1_B_PIN, OUTPUT);

  pinMode(P2_A_PIN, OUTPUT);
  pinMode(P2_B_PIN, OUTPUT);

  pinMode(PWM_P1, OUTPUT);
  pinMode(PWM_P2, OUTPUT);

  pinMode(EN_PIN_1, OUTPUT);
  pinMode(EN_PIN_2, OUTPUT);

  pinMode(BLUE, INPUT); // PIN DE ENTRADA PARA CONTROL MANUAL O Bluetooth controla led bluetooth D1

  pinMode(cold,OUTPUT); //Led "frio" como salida
  digitalWrite(cold,LOW);
}
```

Figura 167. Prueba 1 Código Arduino Parte 2.

```
Lazoabierto Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

Lazoabierto §
|
Serial_2.begin(9600);    // Iniciamos el puerto nuevo Serial_2 (BT) a 9600 Baudios
Serial.begin(115200);   // Initiates the serial to do the monitoring
pinMode(13,1);         // Configura Pin 13 como Salida para led manual D2
pinMode(4,1);          // Configura Pin 4 como Salida para led frio D3

sensors.begin();      //Se inicia el sensor
lcd.init();
//Encender la luz de fondo.
lcd.backlight();

//*****
//*****      SINTONIA POR ZIEGLER y NICHOLS      *****
//*****
//kp=((1.2*tau)/(k*theta))/2;
kp=(1.2*tau)/(k*theta);
ti=2.0*theta;
td=0.5*theta;

//*****
//*****      PID DIGITAL      *****
//*****
// Calculo do controle PID digital
q0=kp*(1+Ts/(2.0*ti)+td/Ts);
q1=-kp*(1-Ts/(2.0*ti)+(2.0*td)/Ts);

q2=(kp*td)/Ts;

}
```

Figura 168. Prueba 1 Código Arduino Parte3.

```
Lazoabierto Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Lazoabierto $

void loop()
{
  manual= digitalRead(11);

  if(manual == false)
  {
    digitalWrite(13,true); // Apaga LED en Pin 13 led manual
    r1= analogRead(A6)*(6.00 / 1023.00)+23; //lee potenciómetro para saber temperatura deseada 16 temperatura minima
    //analogRead(A6)*(6.00 / 1023.00) me bota un valor entre 0 a 6, luego sumo el +23 para que el potenciómetro me indique valores entre 23 y 23+6=29
  }

  else{
    digitalWrite(13,false); // Apaga LED en Pin 13

    // definir temperatura de usuario cuando este en modo BT

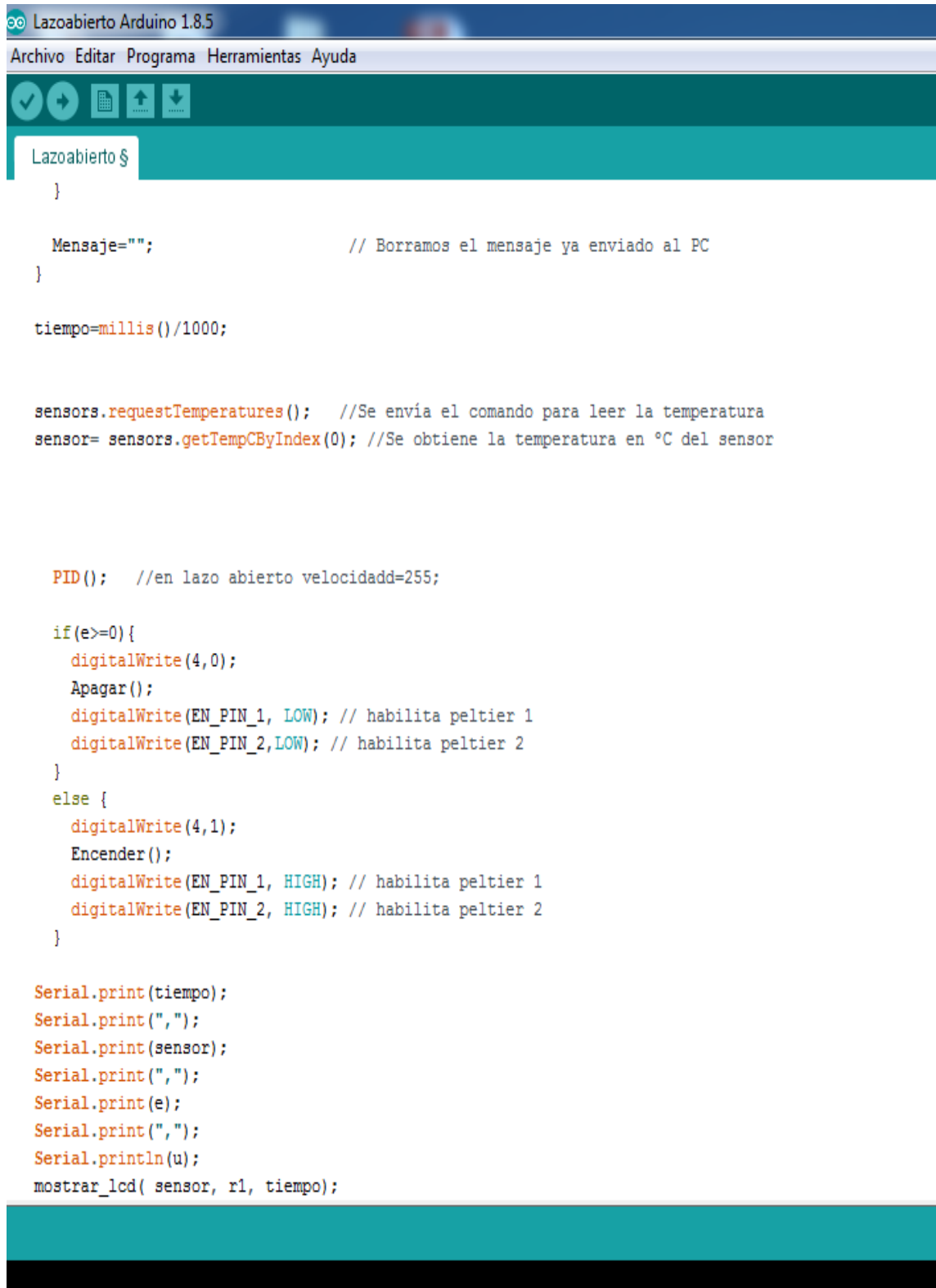
    while (Serial_2.available()) { // Mientras que lleguen caracteres por el puerto serial_2
      delay(5);
      char c = Serial_2.read(); // Lee los caracteres uno a uno en la variable c
      Mensaje += c; // Almacena la suma de caracteres en el mensaje
    }

    if (Mensaje.length()>0){
      //Serial.println( Mensaje); // envia mensaje al PC

      if (Mensaje == "encender"){
        r1=23; //10 para ver cual es la temperatura minima en lazo abierto
        Serial.println( "Monster prendido ");
      }

      if (Mensaje == "apagar"){
        r1=29;
        Serial.println( "Monster apagado ");
      }
    }
  }
}
```

Figura 169. Prueba 1 Código Arduino Parte4.



```
Lazoabierto Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

Lazoabierto $
}

Mensaje=""; // Borramos el mensaje ya enviado al PC
}

tiempo=millis()/1000;

sensors.requestTemperatures(); //Se envía el comando para leer la temperatura
sensor= sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor

PID(); //en lazo abierto velocidadd=255;

if(e>=0){
  digitalWrite(4,0);
  Apagar();
  digitalWrite(EN_PIN_1, LOW); // habilita peltier 1
  digitalWrite(EN_PIN_2,LOW); // habilita peltier 2
}
else {
  digitalWrite(4,1);
  Encender();
  digitalWrite(EN_PIN_1, HIGH); // habilita peltier 1
  digitalWrite(EN_PIN_2, HIGH); // habilita peltier 2
}

Serial.print(tiempo);
Serial.print(",");
Serial.print(sensor);
Serial.print(",");
Serial.print(e);
Serial.print(",");
Serial.println(u);
mostrar_lcd( sensor, r1, tiempo);
```

Figura 170. Prueba 1 Código Arduino Parte5.

```
Lazoabierto Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

Lazoabierto $

}

void Apagar()
{
  Estado_peltier = APAGADO;
  motorGo(Estado_peltier);
}

void Encender()
{
  Estado_peltier = ENCENDIDO;
  motorGo(Estado_peltier);
}

void motorGo( uint8_t direct) //Function that controls the variables: motor(0 ou 1), direction (cw ou ccw) e pwm (entra 0 e 255);
{
  if(direct == ENCENDIDO)
  {
    digitalWrite(P1_A_PIN, HIGH);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, HIGH);
    digitalWrite(P2_B_PIN, LOW);
  }
  else
  {
    digitalWrite(P1_A_PIN, LOW);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, LOW);
    digitalWrite(P2_B_PIN, LOW);
  }
}
```

Figura 171. Prueba 1 Código Arduino Parte6.



```
Lazoabierto Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Lazoabierto $
analogWrite(PWM_P1, velocidad);
delay(10);
analogWrite(PWM_P2, velocidad);
}

void PID(void)
{
    e=(r1-sensor);

    //Serial.print("error ");
    //Serial.println(e);
    // delay (10);
    // Controle PID
    //u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto

    u=100; //prueba en lazo abierto

    if (u >= 100.0) //Saturo la accion de control 'uT' en un tope maximo y minimo
        u = 100.0;

    if (u <= 0.0 || r1==0)
        u = 0.0;

    //Serial.print("U ");
    //Serial.println(u);
    //delay(10);

    //Retorno a los valores reales
    e_2=e_1;
    e_1=e;
    u_1=u;
}
```

Figura 172. Prueba 1 Código Arduino Parte7.

```
//Serial.print("U ");
//Serial.println(u);
//delay(10);

//Retorno a los valores reales
e_2=e_1;
e_1=e;
u_1=u;

//La accion calculada la transformo en PWM

velocidad = 2.55*u;
//analogWrite(velocidad,map(u, 0,100, 0,255));
//Serial.print("velocidad ");
//Serial.println(velocidad);
//delay(10);

}

void mostrar_lcd(float x1, float x2, int x3){
    // Cursor en la primera posición de la primera fila
    lcd.setCursor(0,0);
    lcd.print("ts:");
    lcd.print(x1,1); //temperatura sensor
    lcd.setCursor(9,0);
    lcd.print("C"); // "337" -> ""
    lcd.setCursor(0,1);
    lcd.print("Tp:");
    lcd.print(x2,1);//1 decimal temperatura potenciómetro
    lcd.print("C"); // "337" -> ""
    // Cursor en la 11ª posición de la 2ª fila
    lcd.setCursor(12,1);
    lcd.print(x3);
    lcd.print("s");
}
}
```

Figura 173. Prueba 1 Código Arduino Parte8.

PRUEBA 2

La prueba es de KP/2 estos datos corresponden a la planta donde la primera fila muestra el tiempo en segundo y la segunda fila la temperatura.

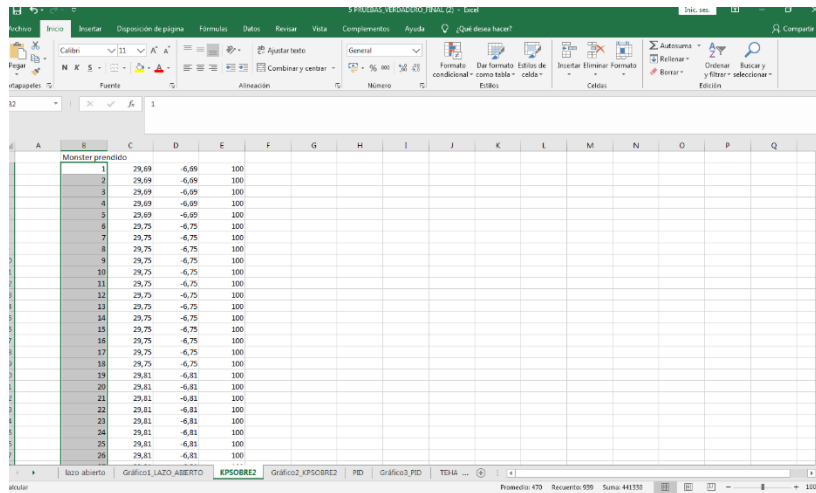


Figura 174. Prueba 2 Excel KP/2.

Aquí se analiza el comportamiento, de la planta en KP/2.

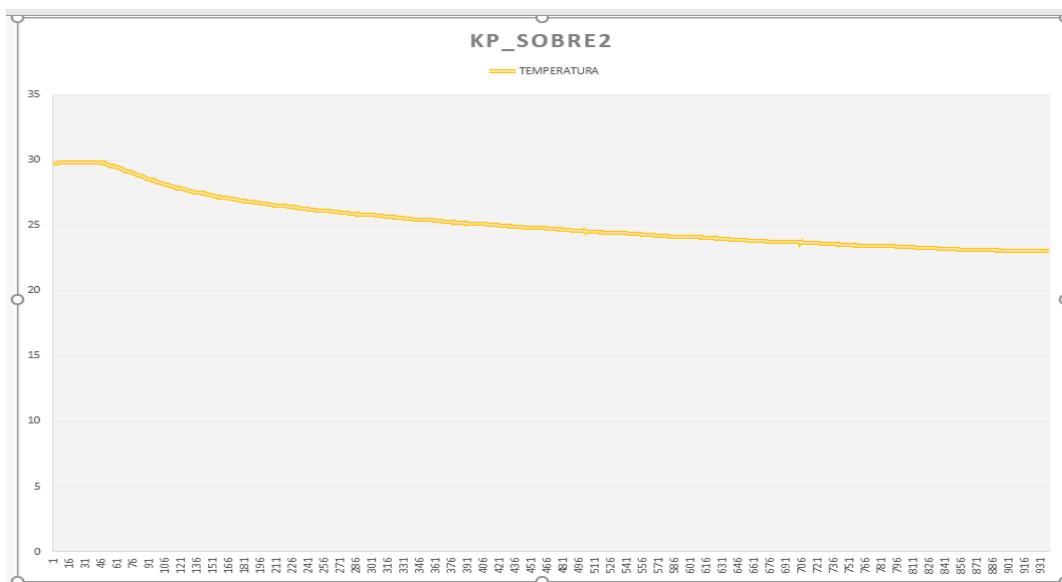


Figura 175. Prueba 2 Gráfica en KP/2.

En el LCD se observa que en 24.2°C la planta tiene 444 segundos que equivale a 7.4 minutos aun terminando establecer el set point máximo, al llegar a la 23.0°C analizando el tiempo de encendido se establece su temperatura máxima en el tiempo de 15.65 minutos.

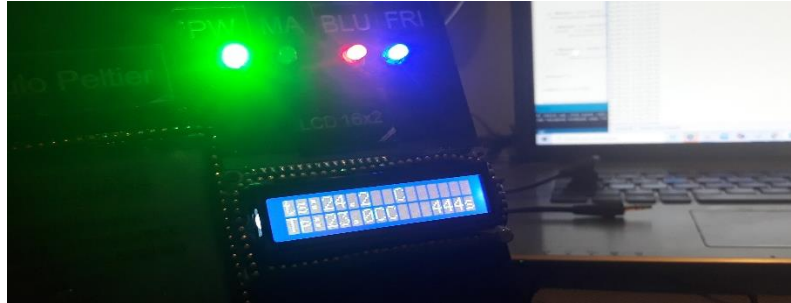


Figura 176. Prueba 2 Planta KP/2.

Código prueba 2

```

Kpsobre2
#include <OneWire.h> //libreriasensor ds18b20
#include <DallasTemperature.h> //libreria sensor ds18b20
#include <Wire.h> //libreria lcd
#include <LiquidCrystal_I2C.h> //libreria lcd
#include <SoftwareSerial.h> //establecer conexion serial para el modulo bluetooth

SoftwareSerial Serial_2 (2, 3); // Crea nueva conexion- Pin2(RX) a TX y Pin3(TX) a RX del BT
String Mensaje; // Variable de cadena de caracteres para almacenar el mensaje
//Crear el objeto lcd dirección 0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2); //
OneWire ourWire(12); //Se establece el pin 12 como bus OneWire
DallasTemperature sensors(ourWire); //Se declara una variable u objeto para nuestro sensor
bool manual=true; //variable para elegir modo manual y no por bluetooth
#define cold 13// led encendido si esta frio
#define BLUE 11
#define APAGADO 0
#define ENCENDIDO 1
#define CS_THRESHOLD 15 // Definition of safety current (Check: "1.3 Monster Shield Example").
//Peltier 1
#define P1_A_PIN 7
#define P1_B_PIN 8
//peltier 2
#define P2_A_PIN 9
#define P2_B_PIN 10
#define PWM_P1 5
#define PWM_P2 6
#define EN_PIN_1 A0 //salida para habilitar salida 1 de la shield
#define EN_PIN_2 A1 //salida para habilitar salida 2 de la shield

short velocidad = 0; //default motor speed
unsigned short Estado_peltier = APAGADO;
int tiempo = 0;
float sensor=29.0; // tseratura del sensor default

```

Figura 177. Prueba 2 Código Arduino Parte1.

The image shows a screenshot of the Arduino IDE interface. At the top, the window title is 'Kpsobre2 Arduino 1.8.5'. Below the title bar is a menu bar with 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. Underneath the menu bar is a toolbar with icons for a checkmark, a right arrow, a document, an up arrow, and a down arrow. The main text area contains the following code:

```
Kpsobre2 $

float r1=100.0; //Referencia temperatura usuario default
volatile float u=0.0,u_1=0.0; //Acción de Control
byte Ts = 84; //Periodo de muestreo
//Parámetros del PID
float kp,ti,td;
float q0,q1,q2;
volatile float e=0.0,e_1=0.0,e_2=0.0;
float k=-0.1271,tau=1667,L=20; //Parámetros del Modelo del sistema
float theta= L +Ts/2;
// tau es la constante de tiempo que se obtiene con el tiempo de estabilizacion dividido para 4

/* fin de parametros para el PID*/

void setup()
{
  pinMode(P1_A_PIN, OUTPUT);
  pinMode(P1_B_PIN, OUTPUT);

  pinMode(P2_A_PIN, OUTPUT);
  pinMode(P2_B_PIN, OUTPUT);

  pinMode(PWM_P1, OUTPUT);
  pinMode(PWM_P2, OUTPUT);

  pinMode(EN_PIN_1, OUTPUT);
  pinMode(EN_PIN_2, OUTPUT);

  pinMode(BLUE, INPUT); // PIN DE ENTRADA PARA CONTROL MANUAL O Bluetooth controla led bluetooch D1

  pinMode(cold,OUTPUT); //Led "frio" como salida
```

Figura 178. Prueba 2 Código Arduino Parte2.

```
Kpsobre2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Kpsobre2 $
|
digitalWrite(cold,LOW);
Serial_2.begin(9600); // Iniciamos el puerto nuevo Serial_2 (BT) a 9600 Baudios
Serial.begin(115200); // Initiates the serial to do the monitoring
pinMode(13,1); // Configura Pin 13 como Salida para led manual D2
pinMode(4,1); // Configura Pin 4 como Salida para led frio D3

sensors.begin(); //Se inicia el sensor
lcd.init();
//Encender la luz de fondo.
lcd.backlight();

//*****
//***** SINTONIA POR ZIEGLER y NICHOLS *****
//*****
kp=((1.2*tau)/(k*theta))/2;
//kp=(1.2*tau)/(k*theta);
ti=2.0*theta;
td=0.5*theta;

//*****
//***** PID DIGITAL *****
//*****
// Calculo do controle PID digital
q0=kp*(1+Ts/(2.0*ti)+td/Ts);
q1=-kp*(1-Ts/(2.0*ti)+(2.0*td)/Ts);

q2=(kp*td)/Ts;

}
```

Figura 179. Prueba 2 Código Arduino Parte3.



```

}

void loop()
{

  manual= digitalRead(11);

  if(manual == false)
  {
    digitalWrite(13,true); // Apaga LED en Pin 13 led manual
    r1= analogRead(A6)*(6.00 / 1023.00)+23; //lee potenciómetro para saber temperatura deseada 16 temperatura minima
    //analogRead(A6)*(6.00 / 1023.00) me bota un valor entre 0 a 6, luego sumo el +23 para que el potenciómetro me indique
  }

  else{
    digitalWrite(13,false); // Apaga LED en Pin 13

    // definir temperatura de usuario cuando este en modo BT

    while (Serial_2.available()) { // Mientras que lleguen caracteres por el puerto serial_2
      delay(5);
      char c = Serial_2.read(); // Lee los caracteres uno a uno en la variable c
      Mensaje += c; // Almacena la suma de caracteres en el mensaje
    }

    if (Mensaje.length(>0){
      //Serial.println( Mensaje); // envia mensaje al PC

      if (Mensaje == "encender"){
        r1=23; //10 para ver cual es la temperatura minima en lazo abierto
        Serial.println( "Monster prendido ");
      }
    }
  }
}

```

Figura 180. Prueba 2 Código Arduino Parte4.

```
Kpsobre2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Kpsobre2$

    if (Mensaje == "apagar"){
        r1=29;
        Serial.println( "Monster apagado ");
    }
}

Mensaje=""; // Borramos el mensaje ya enviado al PC
}

tiempo=millis()/1000;

sensors.requestTemperatures(); //Se envía el comando para leer la temperatura
sensor= sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor

PID(); //en lazo abierto velocidad=255;

if(e>=0){
    digitalWrite(4,0);
    Apagar();
    digitalWrite(EN_PIN_1, LOW); // habilita peltier 1
    digitalWrite(EN_PIN_2,LOW); // habilita peltier 2
}
else {
    digitalWrite(4,1);
    Encender();
    digitalWrite(EN_PIN_1, HIGH); // habilita peltier 1
    digitalWrite(EN_PIN_2, HIGH); // habilita peltier 2
}

Serial.print(tiempo);
Serial.print(",");
Serial.println(sensor);
```

Figura 181. Prueba 2 Código Arduino Parte5.

```
Kpsobre2$
Serial.print(",");
Serial.print(e);
Serial.print(",");
Serial.println(u);
mostrar_lcd( sensor, r1, tiempo);
//delay(1000); // delay para observar los cambios en el puerto serial de forma mas pausada
//este delay debe ser comentado (eliminado) en el programa final.

|
}

void Apagar()
{
  Estado_peltier = APAGADO;
  motorGo(Estado_peltier);
}

void Encender()
{
  Estado_peltier = ENCENDIDO;
  motorGo(Estado_peltier);
}

void motorGo( uint8_t direct) //Function that controls the variables: motor(0 ou 1), direction (cw ou ccw) e pwm (entra 0 e 255);
{
  if(direct == ENCENDIDO)
  {
    digitalWrite(P1_A_PIN, HIGH);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, HIGH);
    digitalWrite(P2_B_PIN, LOW);
  }
  else
```

Figura 182. Prueba 2 Código Arduino Parte6.

```
Kpsobre2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Kpsobre2 $
{
  digitalWrite(P1_A_PIN, LOW);
  digitalWrite(P1_B_PIN, LOW);
  delay(10);
  digitalWrite(P2_A_PIN, LOW);
  digitalWrite(P2_B_PIN, LOW);
}

analogWrite(PWM_P1, velocidad);
delay(10);
analogWrite(PWM_P2, velocidad);

}

void PID(void)
{
  e=(r1-sensor);

  //Serial.print("error ");
  //Serial.println(e);
  // delay (10);
  // Controle PID
  u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto

  //u=100; //prueba en lazo abierto

  if (u >= 100.0) //Saturo la accion de control 'uT' en un tope maximo y minimo
    u = 100.0;

  if (u <= 0.0 || r1==0)
    u = 0.0;

  //Serial.print("U ");
  //Serial.println(u);
}
```

Figura 183. Prueba 2 Código Arduino Parte7.


```
Kpsobre2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Kpsobre2 $
//Serial.print("0 ");
//Serial.println(u);

//delay(10);

//Retorno a los valores reales
e_2=e_1;
e_1=e;
u_1=u;

//La accion calculada la transformo en PWM

velocidad = 2.55*u;
//analogWrite(velocidad,map(u, 0,100, 0,255));
//Serial.print("velocidad ");
//Serial.println(velocidad);
//delay(10);

}

void mostrar_lcd(float x1, float x2, int x3){
    // Cursor en la primera posición de la primera fila
    lcd.setCursor(0,0);
    lcd.print("ts:");
    lcd.print(x1,1); //temperatura sensor
    lcd.setCursor(9,0);
    lcd.print("C"); // "337" -> ""
    lcd.setCursor(0,1);
    lcd.print("Tp:");
    lcd.print(x2,1);//1 decimal temperatura potenciómetro
    lcd.print("C"); // "337" -> ""
    // Cursor en la 11ª posición de la 2ª fila
    lcd.setCursor(12,1);
    lcd.print(x3);
    lcd.print("s");
}

```

Figura 184. Prueba 2 Código Arduino Parte8.

PRUEBA 3

La prueba es de theta igual A retardo estos datos corresponden a la planta donde la primera fila muestra el tiempo en segundo y la segunda fila la temperatura.

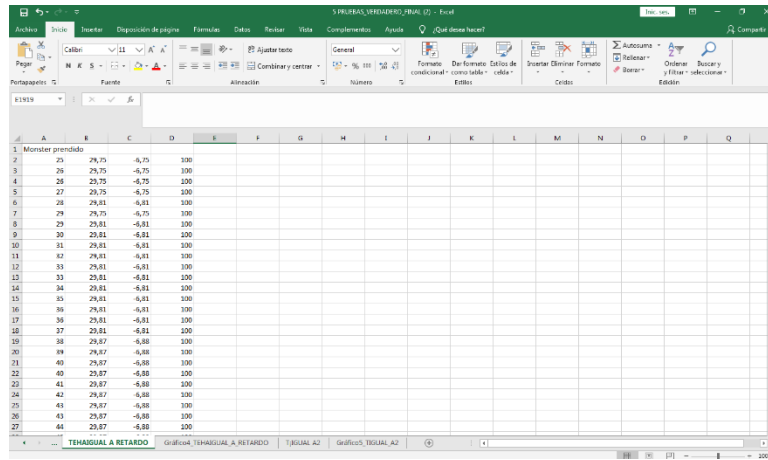


Figura 185. Prueba 3 Excel theta igual A retardo.

Aquí se analiza el comportamiento, de la planta cuando theta igual A retardo.

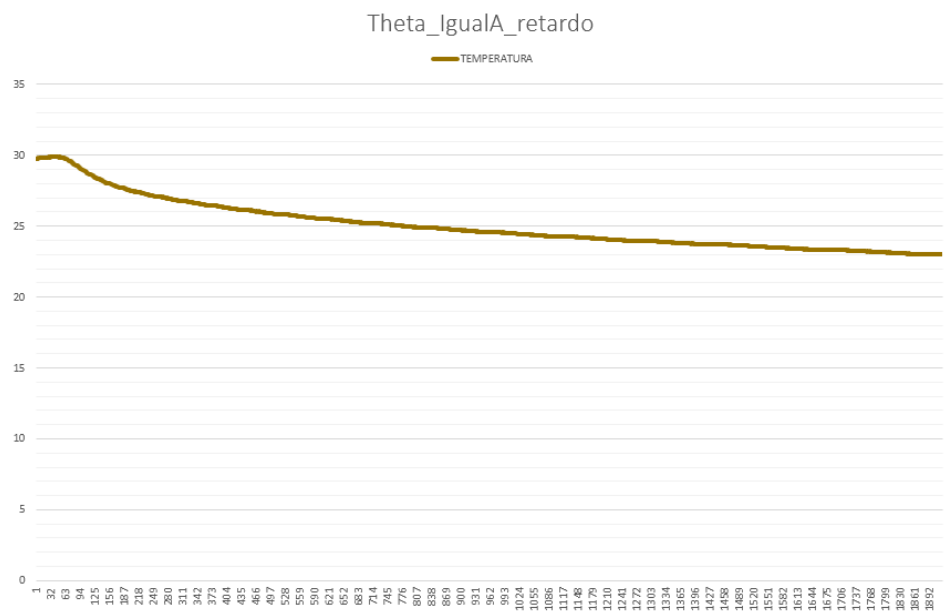


Figura 186. Prueba 3 Gráfica theta igual A retardo.

En el LCD se observa que en 23.1°C la planta tiene 1442 segundos que equivale a 24.03 minutos aun terminando establecer el set point máximo, al llegar a la 23.0 °C analizando el tiempo de encendido se establece su temperatura máxima en el tiempo de 25.2 minutos.

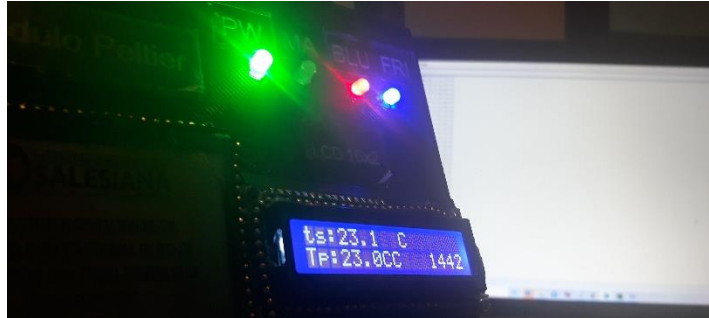


Figura 187. Prueba 3 Planta theta igual A retardo.

Código prueba 3

```

thetaigualretardo

#include <OneWire.h> //libreriasensor ds18b20
#include <DallasTemperature.h> //libreria sensor ds18b20
#include <Wire.h> //libreria lcd
#include <LiquidCrystal_I2C.h> //libreria lcd
#include <SoftwareSerial.h> //establecer conexion serial para el modulo bluetooth

SoftwareSerial Serial_2 (2, 3); // Crea nueva conexion- Pin2(RX) a TX y Pin3(TX) a RX del BT
String Mensaje; // Variable de cadena de caracteres para almacenar el mensaje
//Crear el objeto lcd dirección 0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2); //
OneWire ourWire(12); //Se establece el pin 12 como bus OneWire
DallasTemperature sensors(&ourWire); //Se declara una variable u objeto para nuestro sensor
bool manual=true; //variable para elegir modo manual y no por bluetooth
#define cold 13// led encendido si esta frio
#define BLUE 11
#define APAGADO 0
#define ENCENDIDO 1
#define CS_THRESHOLD 15 // Definition of safety current (Check: "1.3 Monster Shield Example").
//Peltier 1
#define P1_A_PIN 7
#define P1_B_PIN 8
//peltier 2
#define P2_A_PIN 9
#define P2_B_PIN 10
#define PWM_P1 5
#define PWM_P2 6
#define EN_PIN_1 A0 //salida para habilitar salida 1 de la shield
#define EN_PIN_2 A1 //salida para habilitar salida 2 de la shield

short velocidad = 0; //default motor speed
unsigned short Estado_peltier = APAGADO;
int tiempo = 0;
float sensor=29.0; // tseratura del sensor default

```

Figura 188. Prueba 3 Código Arduino Parte1.



```
thetaigualretardo Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

thetaigualretardo $

/* parametros para el PID*/

float r1=100.0; //Referencia temperatura usuario default
volatile float u=0.0,u_1=0.0; //Acción de Control
byte Ts = 84; //Periodo de muestreo
//Parámetros del PID
float kp,ti,td;
float q0,q1,q2;
volatile float e=0.0,e_1=0.0,e_2=0.0;
float k=-0.1271,tau=1667,L=20; //Parámetros del Modelo del sistema
//float theta= L +Ts/2;
float theta= L ;
// tau es la constante de tiempo que se obtiene con el tiempo de estabilizacion dividido para 4

/* fin de parametros para el PID*/

void setup()
{
  pinMode(P1_A_PIN, OUTPUT);
  pinMode(P1_B_PIN, OUTPUT);

  pinMode(P2_A_PIN, OUTPUT);
  pinMode(P2_B_PIN, OUTPUT);

  pinMode(PWM_P1, OUTPUT);
  pinMode(PWM_P2, OUTPUT);

  pinMode(EN_PIN_1, OUTPUT);
  pinMode(EN_PIN_2, OUTPUT);
}
```

Figura 189. Prueba 3 Código Arduino Parte2.

```

thetaigualretardo Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
thetaigualretardo $
}
pinMode(BLUE, INPUT); // PIN DE ENTRADA PARA CONTROL MANUAL O Bluetooth controla led blueooh D1

pinMode(cold,OUTPUT); //Led "frio" como salida
digitalWrite(cold,LOW);
Serial_2.begin(9600); // Iniciamos el puerto nuevo Serial_2 (BT) a 9600 Baudios
Serial.begin(115200); // Initiates the serial to do the monitoring
pinMode(13,1); // Configura Pin 13 como Salida para led manual D2
pinMode(4,1); // Configura Pin 4 como Salida para led frio D3

sensors.begin(); //Se inicia el sensor
lcd.init();
//Encender la luz de fondo.
lcd.backlight();

//*****
//***** SINTONIA POR ZIEGLER y NICHOLS *****
//*****
//kp=((1.2*tau)/(k*theta))/2;
kp=(1.2*tau)/(k*theta);
ti=2.0*theta;
td=0.5*theta;

//*****
//***** PID DIGITAL *****
//*****
// Calculo do controle PID digital
q0=kp*(1+Ts/(2.0*ti)+td/Ts);
q1=-kp*(1-Ts/(2.0*ti)+(2.0*td)/Ts);

q2=(kp*td)/Ts;

}

```

Figura 190. Prueba 3 Código Arduino Parte3.

```
thetaigualretardo Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
thetaigualretardo $

}

void loop()
{

  manual= digitalRead(11);

  if(manual == false)
  {
    digitalWrite(13,true); // Apaga LED en Pin 13 led manual
    r1= analogRead(A6)*(6.00 / 1023.00)+23; //lee potencio metro para saber tseratura deseada 16 temperatura minima
    //analogRead(A6)*(6.00 / 1023.00) me bota un valor entre 0 a 6, luego sumo el +23 para que el potencio metro me indique valores entre 23 ;
  }

  else{
    digitalWrite(13,false); // Apaga LED en Pin 13

    // definir temperatura de usuario cuando este en modo BT

    while (Serial_2.available()) { // Mientras que lleguen caracteres por el puerto serial_2
      delay(5);
      char c = Serial_2.read(); // Lee los caracteres uno a uno en la variable c
      Mensaje += c; // Almacena la suma de caracteres en el mensaje
    }

    if (Mensaje.length()>0){
      //Serial.println( Mensaje); // envia mensaje al PC

      if (Mensaje == "encender"){
        r1=23; //10 para ver cual es la temperatura minima en lazo abierto
        Serial.println( "Monster prendido ");
      }
    }
  }
}
```

Figura 191. Prueba 3 Código Arduino Parte4.

```
thetaigualretardo Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
thetaigualretardo $

    if (Mensaje == "apagar"){
        r1=29;
        Serial.println( "Monster apagado ");
    }
}

Mensaje=""; // Borramos el mensaje ya enviado al PC
}

tiempo=millis()/1000;

sensors.requestTemperatures(); //Se envía el comando para leer la temperatura
sensor= sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor

PID(); //en lazo abierto velocidad=255;

if(e>=0){
    digitalWrite(4,0);
    Apagar();
    digitalWrite(EN_PIN_1, LOW); // habilita peltier 1
    digitalWrite(EN_PIN_2,LOW); // habilita peltier 2
}
else {
    digitalWrite(4,1);
    Encender();
    digitalWrite(EN_PIN_1, HIGH); // habilita peltier 1
    digitalWrite(EN_PIN_2, HIGH); // habilita peltier 2
}

Serial.print(tiempo);
Serial.print(",");
Serial.print(sensor);
```

Figura 192. Prueba 3 Código Arduino Parte5.

```
thetaigualretardo Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
thetaigualretardo $
Serial.print(",");
Serial.print(e);
Serial.print(",");
Serial.println(u);
mostrar_lcd( sensor, r1, tiempo);
//delay(1000); // delay para observar los cambios en el puerto serial de forma mas pausada
//este delay debe ser comentado (eliminado) en el programa final.

}

void Apagar()
{
  Estado_peltier = APAGADO;
  motorGo(Estado_peltier);
}

void Encender()
{
  Estado_peltier = ENCENDIDO;
  motorGo(Estado_peltier);
}

void motorGo( uint8_t direct) //Function that controls the variables: motor(0 ou 1), direction (cw ou ccw) e pwm (e
{
  if(direct == ENCENDIDO)
  {
    digitalWrite(P1_A_PIN, HIGH);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, HIGH);
    digitalWrite(P2_B_PIN, LOW);
  }
}
```

Figura 193. Prueba 3 Código Arduino Parte6.


```
thetaigualretardo Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

thetaigualretardo $
    digitalWrite(P2_B_PIN, LOW);
  }
  else
  {
    digitalWrite(P1_A_PIN, LOW);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, LOW);
    digitalWrite(P2_B_PIN, LOW);
  }

  analogWrite(PWM_P1, velocidad);
  delay(10);
  analogWrite(PWM_P2, velocidad);
}

void PID(void)
{
  e=(r1-sensor);

  //Serial.print("error ");
  //Serial.println(e);
  // delay (10);
  // Controle PID
  u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto

  //u=100; //prueba en lazo abierto

  if (u >= 100.0) //Saturó la acción de control 'uT' en un tope máximo y mínimo
    u = 100.0;

  if (u <= 0.0 || r1==0)
    u = 0.0;
}
```

Figura 194. Prueba 3 Código Arduino Parte7.

```
thetaigualretardo Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
thetaigualretardo §

//Serial.print("U ");
//Serial.println(u);
//delay(10);

//Retorno a los valores reales
e_2=e_1;
e_1=e;
u_1=u;

//La accion calculada la transformo en PWM

velocidad = 2.55*u;
//analogWrite(velocidad,map(u, 0,100, 0,255));
//Serial.print("velocidad ");
//Serial.println(velocidad);
//delay(10);

}

void mostrar_lcd(float x1, float x2, int x3){
    // Cursor en la primera posición de la primera fila
    lcd.setCursor(0,0);
    lcd.print("ts:");
    lcd.print(x1,1); //temperatura sensor
    lcd.setCursor(9,0);
    lcd.print("C"); // "337" -> ""
    lcd.setCursor(0,1);
    lcd.print("Tp:");
    lcd.print(x2,1); //1 decimal temperatura potenciómetro
    lcd.print("C"); // "337" -> ""
    // Cursor en la 11ª posición de la 2ª fila
    lcd.setCursor(12,1);
    lcd.print(x3);
    lcd.print("s");
}
}
```

Figura 195. Prueba 3 Código Arduino Parte8.

PRUEBA 4

La prueba es ts igual a 2 estos datos corresponden a la planta donde la primera fila muestra el tiempo en segundo y la segunda fila la temperatura.

Time (s)	Temperature	Value 1	Value 2
14	32.31	-9.31	100
15	32.31	-9.31	0
16	32.31	-9.31	100
17	32.31	-9.31	100
18	32.31	-9.31	100
19	32.31	-9.31	100
20	32.25	-9.25	100
21	32.25	-9.25	100
22	32.25	-9.25	100
23	32.25	-9.25	100
24	32.25	-9.25	100
25	32.25	-9.25	100
26	32.25	-9.25	100
27	32.25	-9.25	100
28	32.25	-9.25	100
29	32.25	-9.25	100
30	32.25	-9.25	100
31	32.25	-9.25	100
32	32.25	-9.25	100
33	32.25	-9.25	100

Figura 196. Prueba 4 Excel ts igual a 2.

Aquí se analiza el comportamiento, de la planta cuando ts igual A 2.

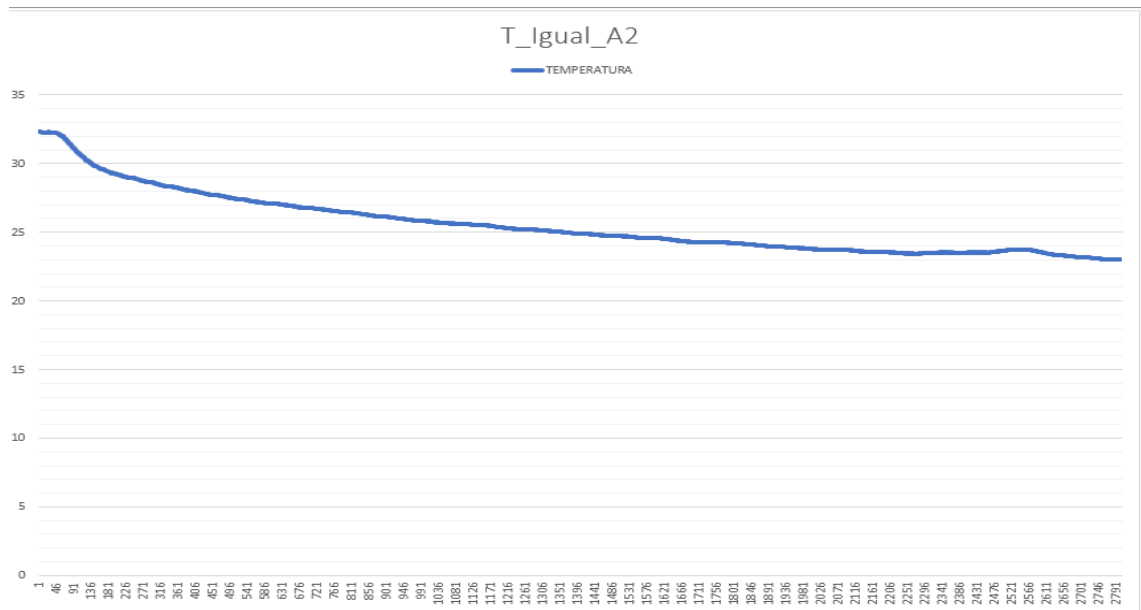


Figura 197. Prueba 4 Gráfica ts igual a 2.

En el LCD se observa que en 23.1°C la planta tiene 2151 segundos que equivale a 35.85 minutos aun terminando establecer el set point máximo, al llegar a la 23.0°C analizando el tiempo de encendido se establece su temperatura máxima en el tiempo de 36.7 minutos.



Figura 198. Prueba 4 Planta ts igual a 2.

Código prueba 4

```

Tsigual2
#include <OneWire.h> //libreriasensor ds18b20
#include <DallasTemperature.h> //libreria sensor ds18b20
#include <Wire.h> //libreria lcd
#include <LiquidCrystal_I2C.h> //libreria lcd
#include <SoftwareSerial.h> //establecer conexion serial para el modulo bluetooth

SoftwareSerial Serial_2 (2, 3); // Crea nueva conexion- Pin2(RX) a TX y Pin3(TX) a RX del BT
String Mensaje; // Variable de cadena de caracteres para almacenar el mensaje
//Crear el objeto lcd dirección 0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2); //
OneWire ourWire(12); //Se establece el pin 12 como bus OneWire
DallasTemperature sensors(&ourWire); //Se declara una variable u objeto para nuestro sensor
bool manual=true; //variable para elegir modo manual y no por bluetooth
#define cold 13// led encendido si esta frio
#define BLUE 11
#define APAGADO 0
#define ENCENDIDO 1
#define CS_THRESHOLD 15 // Definition of safety current (Check: "1.3 Monster Shield Example").
//Peltier 1
#define P1_A_PIN 7
#define P1_B_PIN 8
//peltier 2
#define P2_A_PIN 9
#define P2_B_PIN 10
#define PWM_P1 5
#define PWM_P2 6
#define EN_PIN_1 A0 //salida para habilitar salida 1 de la shield
#define EN_PIN_2 A1 //salida para habilitar salida 2 de la shield

short velocidad = 0; //default motor speed
unsigned short Estado_peltier = APAGADO;
int tiempo = 0;
float sensor=29.0; // tseratura del sensor default

```

Figura 199. Prueba 4 Código Arduino Parte1.



```
Tsiiigual2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

Tsiiigual2 §
float r1=100.0; //Referencia temperatura usuario default
volatile float u=0.0,u_1=0.0; //Acción de Control
byte Ts = 2; //Periodo de muestreo
//Parámetros del PID
float kp,ti,td;
float q0,q1,q2;
volatile float e=0.0,e_1=0.0,e_2=0.0;
float k=-0.1271,tau=1667,L=20; //Parámetros del Modelo del sistema
float theta= L +Ts/2;
// tau es la constante de tiempo que se obtiene con el tiempo de estabilizacion dividido para 4

/* fin de parametros para el PID*/

void setup()
{
  pinMode(P1_A_PIN, OUTPUT);
  pinMode(P1_B_PIN, OUTPUT);

  pinMode(P2_A_PIN, OUTPUT);
  pinMode(P2_B_PIN, OUTPUT);

  pinMode(PWM_P1, OUTPUT);
  pinMode(PWM_P2, OUTPUT);

  pinMode(EN_PIN_1, OUTPUT);
  pinMode(EN_PIN_2, OUTPUT);

  pinMode(BLUE, INPUT); // PIN DE ENTRADA PARA CONTROL MANUAL O Bluetooth controla led bluetooth D1

  pinMode(cold,OUTPUT); //Led "frio" como salida
  digitalWrite(cold,LOW);
```

Figura 200. Prueba 4 Código Arduino Parte2.

```
Tsiigual2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Tsiigual2 $

Serial_2.begin(9600); // Iniciamos el puerto nuevo Serial_2 (BT) a 9600 Baudios
Serial.begin(115200); // Initiates the serial to do the monitoring
pinMode(13,1); // Configura Pin 13 como Salida para led manual D2
pinMode(4,1); // Configura Pin 4 como Salida para led frio D3

sensors.begin(); //Se inicia el sensor
lcd.init();
//Encender la luz de fondo.
lcd.backlight();

//*****
//***** SINTONIA POR ZIEGLER y NICHOLS *****
//*****
//kp=((1.2*tau)/(k*theta))/2;
kp=(1.2*tau)/(k*theta);
ti=2.0*theta;
td=0.5*theta;

//*****
//***** PID DIGITAL *****
//*****
// Calculo do controle PID digital
q0=kp*(1+Ts/(2.0*ti)+td/Ts);
q1=-kp*(1-Ts/(2.0*ti)+(2.0*td)/Ts);

q2=(kp*td)/Ts;

}
```

Figura 201. Prueba 4 Código Arduino Parte3.

```
void loop()
{

  manual= digitalRead(11);

  if(manual == false)
  {
    digitalWrite(13,true); // Apaga LED en Pin 13 led manual
    r1= analogRead(A6)*(6.00 / 1023.00)+23; //lee potenciómetro para saber temperatura deseada 16 temperatura minima
    //analogRead(A6)*(6.00 / 1023.00) me bota un valor entre 0 a 6, luego sumo el +23 para que el potenciómetro me in
  }

  else{
    digitalWrite(13,false); // Apaga LED en Pin 13

    // definir temperatura de usuario cuando este en modo BT

    while (Serial_2.available()) { // Mientras que lleguen caracteres por el puerto serial_2
      delay(5);
      char c = Serial_2.read(); // Lee los caracteres uno a uno en la variable c
      Mensaje += c; // Almacena la suma de caracteres en el mensaje
    }

    if (Mensaje.length()>0){
      //Serial.println( Mensaje); // envia mensaje al PC

      if (Mensaje == "encender"){
        r1=23; //10 para ver cual es la temperatura minima en lazo abierto
        Serial.println( "Monster prendido ");
      }

      if (Mensaje == "apagar"){
        r1=29;
        Serial.println( "Monster apagado ");
      }
    }
  }
}
```

Figura 202. Prueba 4 Código Arduino Parte4.

```
Tsiigual2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Tsiigual2 §

  Mensaje=""; // Borramos el mensaje ya enviado al PC
}

tiempo=millis()/1000;

sensors.requestTemperatures(); //Se envía el comando para leer la temperatura
sensor= sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor

PID(); //en lazo abierto velocidadd=255;

if(e>=0){
  digitalWrite(4,0);
  Apagar();
  digitalWrite(EN_PIN_1, LOW); // habilita peltier 1
  digitalWrite(EN_PIN_2,LOW); // habilita peltier 2
}
else {
  digitalWrite(4,1);
  Encender();
  digitalWrite(EN_PIN_1, HIGH); // habilita peltier 1
  digitalWrite(EN_PIN_2, HIGH); // habilita peltier 2
}

Serial.print(tiempo);
Serial.print(",");
Serial.print(sensor);
Serial.print(",");
Serial.print(e);
Serial.print(",");
Serial.println(u);

ostrar_lcd( sensor, r1, tiempo);
```

Figura 203. Prueba 4 Código Arduino Parte5.



```
Tsiigual2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Tsiigual2 $
}

void Apagar()
{
  Estado_peltier = APAGADO;
  motorGo(Estado_peltier);
}

void Encender()
{
  Estado_peltier = ENCENDIDO;
  motorGo(Estado_peltier);
}

void motorGo( uint8_t direct) //Function that controls the variables: motor(0 ou 1)
{
  if(direct == ENCENDIDO)
  {
    digitalWrite(P1_A_PIN, HIGH);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, HIGH);
    digitalWrite(P2_B_PIN, LOW);
  }
  else
  {
    digitalWrite(P1_A_PIN, LOW);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, LOW);
    digitalWrite(P2_B_PIN, LOW);
  }
}
```

Figura 204. Prueba 4 Código Arduino Parte6.



```
Tsiigual2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

Tsiigual2 $

analogWrite(PWM_P1, velocidad);
delay(10);
analogWrite(PWM_P2, velocidad);

}

void PID(void)
{
    e=(r1-sensor);

    //Serial.print("error ");
    //Serial.println(e);
    // delay (10);
    // Controle PID
    u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto

    //u=100; //prueba en lazo abierto

    if (u >= 100.0) //Saturo la accion de control 'uT' en un tope maximo y minimo
        u = 100.0;

    if (u <= 0.0 || r1==0)
        u = 0.0;

    //Serial.print("U ");
    //Serial.println(u);
    //delay(10);

    //Retorno a los valores reales
    e_2=e_1;
    e_1=e;
    u_1=u;
}
```

Figura 205. Prueba 4 Código Arduino Parte7.

```
Tsiigual2 Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

Tsiigual2 §

//Serial.print("U ");
//Serial.println(u);
//delay(10);

//Retorno a los valores reales
e_2=e_1;
e_1=e;
u_1=u;

//La accion calculada la transformo en PWM

velocidad = 2.55*u;
//analogWrite(velocidad,map(u, 0,100, 0,255));
//Serial.print("velocidad ");
//Serial.println(velocidad);
//delay(10);

}

void mostrar_lcd(float x1, float x2, int x3){
    // Cursor en la primera posición de la primera fila
    lcd.setCursor(0,0);
    lcd.print("ts:");
    lcd.print(x1,1); //temperatura sensor
    lcd.setCursor(9,0);
    lcd.print("C"); // "337" -> ""
    lcd.setCursor(0,1);
    lcd.print("Tp:");
    lcd.print(x2,1);//1 decimal temperatura potenciómetro
    lcd.print("C"); // "337" -> ""
    // Cursor en la 11ª posición de la 2ª fila
    lcd.setCursor(12,1);
    lcd.print(x3);
    lcd.print("s");
}
}
```

Figura 206. Prueba 4 Código Arduino Parte8.

PRUEBA 5

La prueba es del PID estos datos corresponden a la planta donde la primera fila muestra el tiempo en segundo y la segunda fila la temperatura.

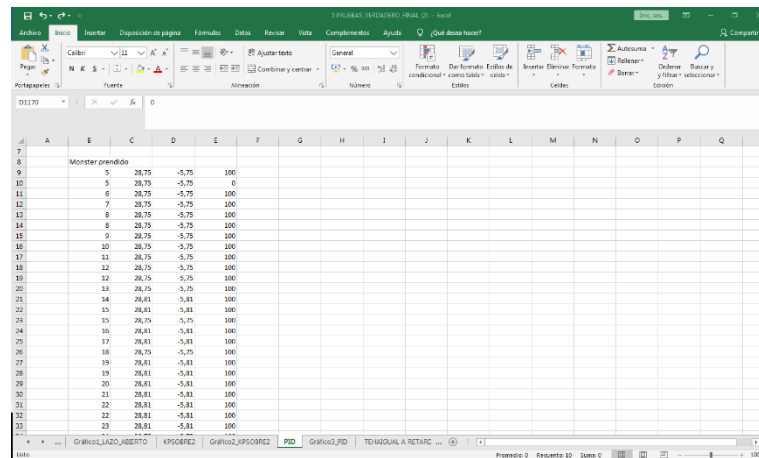


Figura 207. Prueba 5 Excel PID.

Aquí se analiza el comportamiento, de la planta PID.

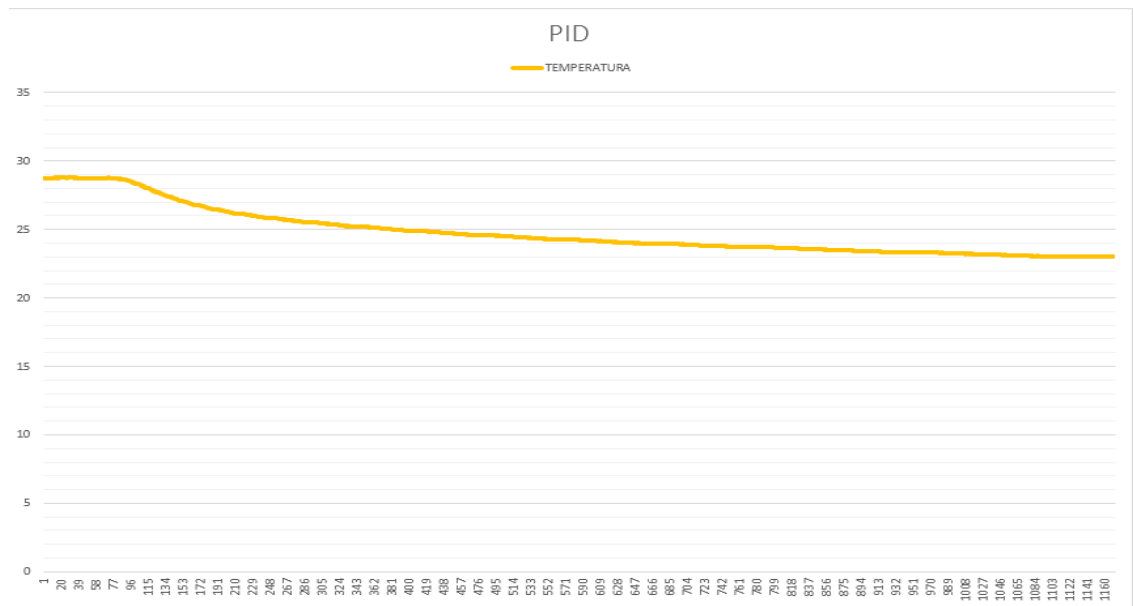


Figura 208. Prueba 5 Gráfica PID.

En el LCD se observa que en 24.2°C la planta tiene 455 segundos que equivale a 7.58 minutos aun terminando establecer el Set Point máximo, al llegar a la 23.0°C analizando el tiempo de encendido se establece su temperatura máxima en el tiempo de 14.4 minutos.



Figura 209. Prueba 5 Planta PID.

Código prueba 5

```

PID
#include <OneWire.h>           //libreriasensor ds18b20
#include <DallasTemperature.h> //libreria sensor ds18b20
#include <Wire.h>              //libreria lcd
#include <LiquidCrystal_I2C.h> //libreria lcd
#include <SoftwareSerial.h>    //establecer conexion serial para el modulo bluetooth

SoftwareSerial Serial_2 (2, 3); // Crea nueva conexion- Pin2(RX) a TX y Pin3(TX) a RX del BT
String Mensaje;               // Variable de cadena de caracteres para almacenar el mensaje
//Crear el objeto lcd dirección 0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2); //
OneWire ourWire(12);          //Se establece el pin 12 como bus OneWire
DallasTemperature sensors(ourWire); //Se declara una variable u objeto para nuestro sensor
bool manual=true;            //variable para elegir modo manual y no por bluetooth
#define cold 13// led encendido si esta frio
#define BLUE 11
#define APAGADO 0
#define ENCENDIDO 1
#define CS_THRESHOLD 15 // Definition of safety current (Check: "1.3 Monster Shield Example").
//Peltier 1
#define P1_A_PIN 7
#define P1_B_PIN 8
//peltier 2
#define P2_A_PIN 9
#define P2_B_PIN 10
#define PWM_P1 5
#define PWM_P2 6
#define EN_PIN_1 A0 //salida para habilitar salida 1 de la shield
#define EN_PIN_2 A1 //salida para habilitar salida 2 de la shield

short velocidad = 0; //default motor speed
unsigned short Estado_peltier = APAGADO;
int tiempo = 0;
float sensor=29.0; // tseratura del sensor default

```

Figura 210. Prueba 5 Código Arduino Parte1.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
PID §

float r1=100.0; //Referencia temperatura usuario default
volatile float u=0.0,u_1=0.0; //Acción de Control
byte Ts = 84; //Periodo de muestreo
//Parámetros del PID
float kp,ti,td;
float q0,q1,q2;
volatile float e=0.0,e_1=0.0,e_2=0.0;
float k=-0.1271,tau=1667,L=20; //Parámetros del Modelo del sistema
float theta= L +Ts/2;
// tau es la constante de tiempo que se obtiene con el tiempo de estabilizacion dividido para 4

/* fin de parametros para el PID*/

void setup()
{
  pinMode(P1_A_PIN, OUTPUT);
  pinMode(P1_B_PIN, OUTPUT);

  pinMode(P2_A_PIN, OUTPUT);
  pinMode(P2_B_PIN, OUTPUT);

  pinMode(PWM_P1, OUTPUT);
  pinMode(PWM_P2, OUTPUT);

  pinMode(EN_PIN_1, OUTPUT);
  pinMode(EN_PIN_2, OUTPUT);

  pinMode(BLUE, INPUT); // PIN DE ENTRADA PARA CONTROL MANUAL O Bluetooth controla led blueetooth D1

  pinMode(cold,OUTPUT); //Led "frio" como salida
  digitalWrite(cold,LOW);
}
```

Figura 211. Prueba 5 Código Arduino Parte2.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
✓ ↻ 📄 ⬆️ ⬇️
PID $
|
Serial_2.begin(9600); // Iniciamos el puerto nuevo Serial_2 (BT) a 9600 Baudios
Serial.begin(115200); // Initiates the serial to do the monitoring
pinMode(13,1); // Configura Pin 13 como Salida para led manual D2
pinMode(4,1); // Configura Pin 4 como Salida para led frio D3

sensors.begin(); //Se inicia el sensor
lcd.init();
//Encender la luz de fondo.
lcd.backlight();

//*****
//***** SINTONIA POR ZIEGLER y NICHOLS *****
//*****
//kp=((1.2*tau)/(k*theta))/2;
kp=(1.2*tau)/(k*theta);
ti=2.0*theta;
td=0.5*theta;

//*****
//***** PID DIGITAL *****
//*****
// Calculo do controle PID digital
q0=kp*(1+Ts/(2.0*ti)+td/Ts);
q1=-kp*(1-Ts/(2.0*ti)+(2.0*td)/Ts);

q2=(kp*td)/Ts;

}
```

Figura 212. Prueba 5 Código Arduino Parte3.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

PID $

void loop()
{

  manual= digitalRead(11);

  if(manual == false)
  {
    digitalWrite(13,true); // Apaga LED en Pin 13 led manual
    r1= analogRead(A6)*(6.00 / 1023.00)+23; //lee potenciometro para saber tseratura deseada 16 temperatu
    //analogRead(A6)*(6.00 / 1023.00) me bota un valor entre 0 a 6, luego sumo el +23 para que el potenc:
  }

  else{
    digitalWrite(13,false); // Apaga LED en Pin 13

    // definir temperatura de usuario cuando este en modo BT

    while (Serial_2.available()) { // Mientras que lleguen caracteres por el puerto serial_2
      delay(5);
      char c = Serial_2.read(); // Lee los caracteres uno a uno en la variable c
      Mensaje += c; // Almacena la suma de caracteres en el mensaje
    }

    if (Mensaje.length(>0){
      //Serial.println( Mensaje); // envia mensaje al PC

      if (Mensaje == "encender"){
        r1=23; //10 para ver cual es la temperatura minima en lazo abierto
        Serial.println( "Monster prendido ");
      }

      if (Mensaje == "apagar"){
        r1=29;
        Serial.println( "Monster apagado ");
      }
    }
  }
}
```

Figura 213. Prueba 5 Código Arduino Parte4.


```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
✓ ↻ 📄 ⬆️ ⬇️
PID $
}

Mensaje=""; // Borramos el mensaje ya enviado al PC
}

tiempo=millis()/1000;

sensors.requestTemperatures(); //Se envía el comando para leer la temperatura
sensor= sensors.getTempCByIndex(0); //Se obtiene la temperatura en °C del sensor

PID(); //en lazo abierto velocidad=255;

if(e>=0){
  digitalWrite(4,0);
  Apagar();
  digitalWrite(EN_PIN_1, LOW); // habilita peltier 1
  digitalWrite(EN_PIN_2,LOW); // habilita peltier 2
}
else {
  digitalWrite(4,1);
  Encender();
  digitalWrite(EN_PIN_1, HIGH); // habilita peltier 1
  digitalWrite(EN_PIN_2, HIGH); // habilita peltier 2
}

Serial.print(tiempo);
Serial.print(",");
Serial.print(sensor);
Serial.print(",");
Serial.print(e);
Serial.print(",");
Serial.println(u);
```

Figura 214. Prueba 5 Código Arduino Parte5.

```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

PID $

mostrar_lcd( sensor, rl, tiempo);
//delay(1000); // delay para observar los cambios en el puerto serial de forma mas pausada
//este delay debe ser comentado (eliminado) en el programa final.

}

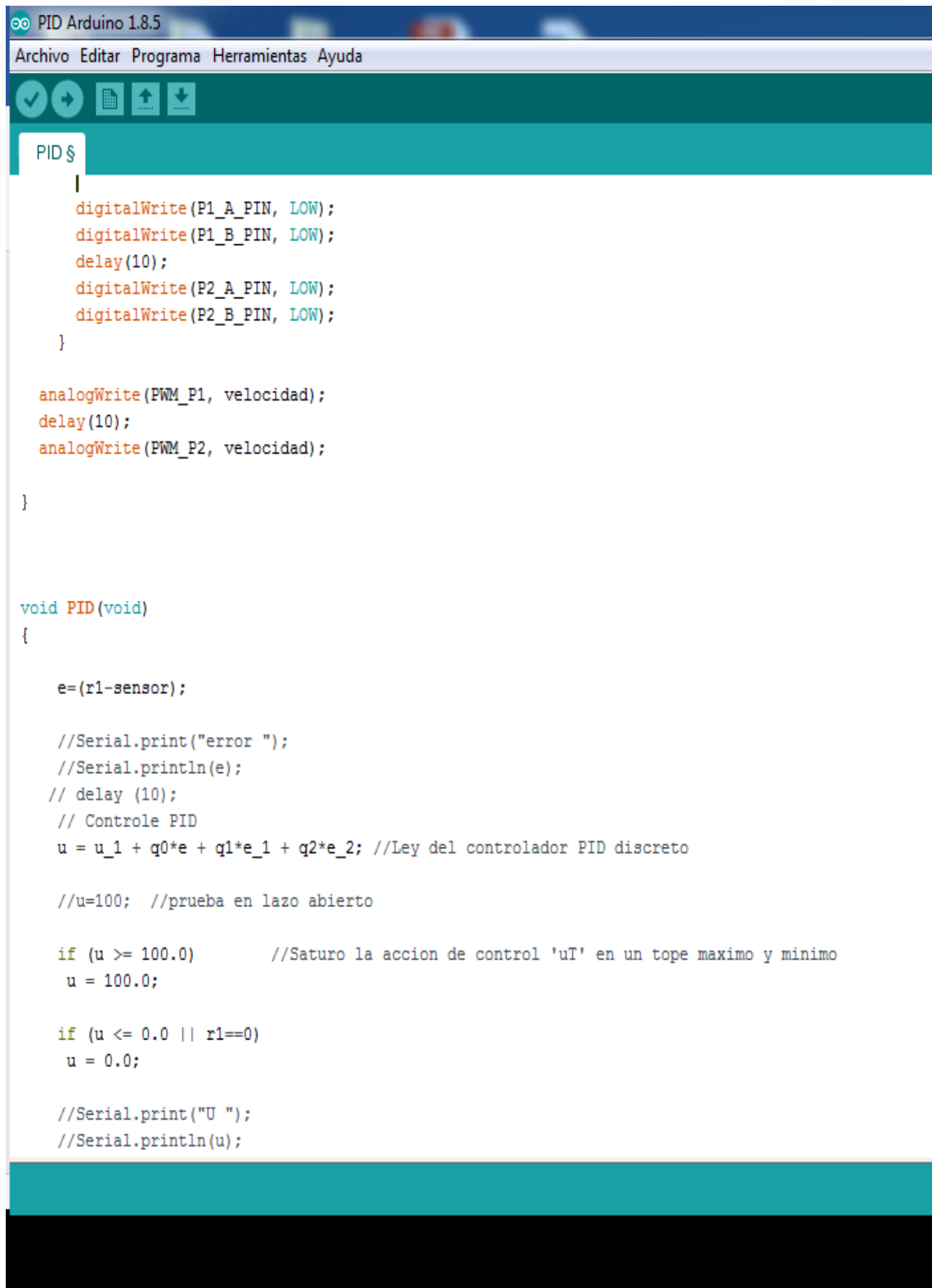
void Apagar()
{
  Estado_peltier = APAGADO;
  motorGo(Estado_peltier);
}

void Encender()
{
  Estado_peltier = ENCENDIDO;
  motorGo(Estado_peltier);
}

void motorGo( uint8_t direct) //Function that controls the variables: motor(0 ou 1), direction (cw ou ccw) e pwm (entra 0 e 255)
{

  if(direct == ENCENDIDO)
  {
    digitalWrite(P1_A_PIN, HIGH);
    digitalWrite(P1_B_PIN, LOW);
    delay(10);
    digitalWrite(P2_A_PIN, HIGH);
    digitalWrite(P2_B_PIN, LOW);
  }
  else
  {
```

Figura 215. Prueba 5 Código Arduino Parte6.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "PID Arduino 1.8.5". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a keyboard, an upload arrow, and a download arrow. The main text area contains the following code:

```
PID $
|
digitalWrite(P1_A_PIN, LOW);
digitalWrite(P1_B_PIN, LOW);
delay(10);
digitalWrite(P2_A_PIN, LOW);
digitalWrite(P2_B_PIN, LOW);
}

analogWrite(PWM_P1, velocidad);
delay(10);
analogWrite(PWM_P2, velocidad);

}

void PID(void)
{

    e=(r1-sensor);

    //Serial.print("error ");
    //Serial.println(e);
    // delay (10);
    // Controle PID
    u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto

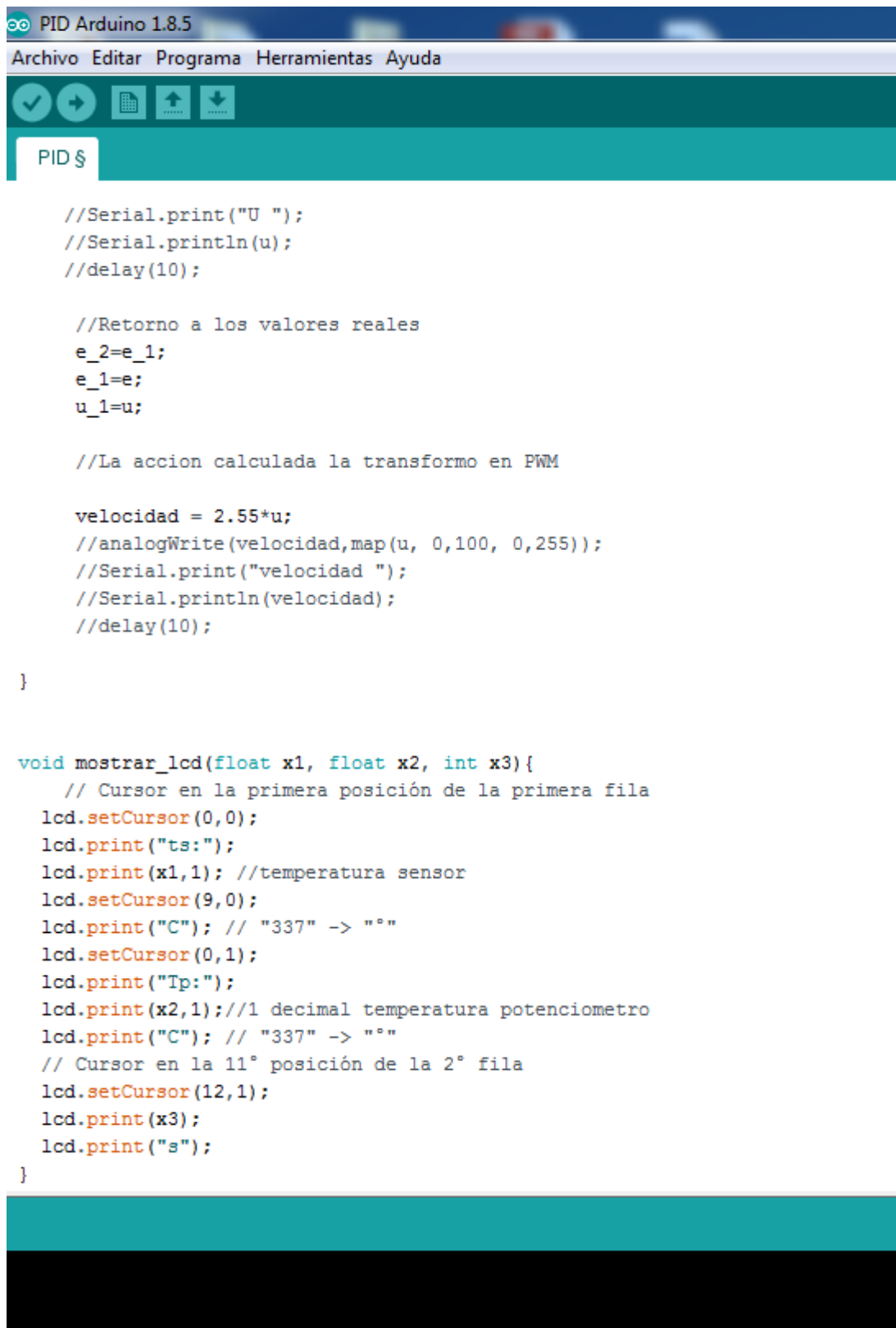
    //u=100; //prueba en lazo abierto

    if (u >= 100.0)          //Saturo la accion de control 'uT' en un tope maximo y minimo
        u = 100.0;

    if (u <= 0.0 || r1==0)
        u = 0.0;

    //Serial.print("U ");
    //Serial.println(u);
```

Figura 216. Prueba 5 Código Arduino Parte7.



```
PID Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
PID $

//Serial.print("U ");
//Serial.println(u);
//delay(10);

//Retorno a los valores reales
e_2=e_1;
e_1=e;
u_1=u;

//La accion calculada la transformo en PWM

velocidad = 2.55*u;
//analogWrite(velocidad,map(u, 0,100, 0,255));
//Serial.print("velocidad ");
//Serial.println(velocidad);
//delay(10);

}

void mostrar_lcd(float x1, float x2, int x3){
  // Cursor en la primera posición de la primera fila
  lcd.setCursor(0,0);
  lcd.print("ts:");
  lcd.print(x1,1); //temperatura sensor
  lcd.setCursor(9,0);
  lcd.print("C"); // "337" -> ""
  lcd.setCursor(0,1);
  lcd.print("Tp:");
  lcd.print(x2,1);//1 decimal temperatura potenciometro
  lcd.print("C"); // "337" -> ""
  // Cursor en la 11ª posición de la 2ª fila
  lcd.setCursor(12,1);
  lcd.print(x3);
  lcd.print("s");
}

```

Figura 217. Prueba 5 Código Arduino Parte8.

