



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

SISTEMA DE SINCRONIZACIÓN Y CONEXIÓN DE ALTERNADORES A UNA RED DE POTENCIA INFINITA

Alumno: Miguel Ángel Chaikhiev Chaikhieva

Tutor: Prof. D. Carlos Sánchez Martos
Dpto: Ingeniería Eléctrica

Julio, 2022



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Eléctrica

Don CARLOS SÁNCHEZ MARTOS, tutor del Trabajo Fin de Grado titulado:
SISTEMA DE SINCRONIZACIÓN Y CONEXIÓN DE ALTERNADORES A UNA
RED DE POTENCIA INFINITA, que presenta MIGUEL ANGEL CHAIKHIEV
CHAIKHIEVA, autoriza su presentación para defensa y evaluación en la
Escuela Politécnica Superior de Jaén.

Jaén, JULIO de 2022

El alumno:

Los tutores:

Índice

1. Introducción.....	10
1.1. Objetivos.	10
1.2. Fases de desarrollo.	11
1.3. Resumen.....	11
2. Motor primario: Motor de corriente continua, funcionamiento y control.....	13
2.1. Estudio del control de velocidad de motores de CC.	13
2.1.1. Principio de funcionamiento.	13
2.1.2. Excitatriz del motor de corriente continua.....	14
2.2. Selección de un sistema de control de velocidad para motores CC.	16
3. Excitatriz de corriente continua, funcionamiento y control.	17
3.1. Selección de un sistema de control de la excitatriz.	19
4. Sistema de control automático de sincronización del alternador a la red.	
Magnitudes eléctricas y mecánicas involucradas.....	21
4.1. Estudio de la maniobra de sincronización y acoplamiento de alternadores a una red de potencia infinita.	21
4.1.1. Condiciones de acoplamiento.	21
4.1.2. Acoplamiento sin cumplimiento de las etapas de sincronización.	24
4.2. Selección de un equipo que controle la maniobra de sincronización.....	30
4.2.1. Microcontrolador stm32 con Arduino.....	30
4.2.2. Interpretación de la ficha técnica stm32f091RC	31
4.2.3. Configuración inicial.	34
4.2.4. Introducción al software Arduino.....	37
4.3. Flujograma de sincronización.	42
4.4. Bloque de lectura y señalización.....	44
4.5. Bloque de control.....	56
4.5.1. Control electrónico: IGBT.	56
4.5.2. Sistema de excitación del IGBT.....	63
4.6. Programación de la etapa de sincronización.....	71
4.6.1. Tensión de referencia.	72

4.6.2.	Inicialización del programa.....	75
4.6.3.	Primera y segunda condición: Igualación en frecuencias y tensión.	79
4.6.4.	Tercera condición: Secuencia directa.....	87
4.6.5.	Cuarta condición: Desfase nulo.	88
4.6.6.	Función de Seguridad.	89
5.	<i>Implementación y ensayos.....</i>	90
5.1.	Conexionado del sistema.	92
6.	<i>Conclusiones.</i>	101
7.	<i>Anexos.....</i>	103
7.1.	Esquemas y Planos.	103
7.2.	Data-Sheet.	109
7.3.	Webgrafía y Bibliografía.....	117

Índice de Ilustraciones

<i>Ilustración 1: Sistema derivación compund larga.....</i>	<i>14</i>
<i>Ilustración 2: Control del motor cc, puente en H.</i>	<i>16</i>
<i>Ilustración 3: Cortocircuitado de una resistencia.</i>	<i>17</i>
<i>Ilustración 4: Excitación del alternador con excitatriz piloto.....</i>	<i>18</i>
<i>Ilustración 5: Rectificadores trifásicos para el control de la excitatriz de los alternadores.....</i>	<i>19</i>
<i>Ilustración 6: Señal PWM, pulse width modulation.....</i>	<i>20</i>
<i>Ilustración 7: Conexionado de bornes en un motor trifásico para obtención de secuencia de fases y de giro.</i>	<i>22</i>
<i>Ilustración 8: Sincronoscopio de lámparas.</i>	<i>22</i>
<i>Ilustración 9: Sincronoscopio de aguja.</i>	<i>23</i>
<i>Ilustración 10: Ángulo de desfase entre tensión de red y generada.....</i>	<i>24</i>
<i>Ilustración 11: Esquema fasorial de una máquina síncrona con carga.</i>	<i>25</i>
<i>Ilustración 12: Curva de capacidad y puntos operacionales para un alternador específico.....</i>	<i>28</i>
<i>Ilustración 13: Periodos de tiempo mínimo durante los que un módulo de generación de electricidad debe ser capaz de funcionar.....</i>	<i>28</i>
<i>Ilustración 14: Rango de potencia reactiva por tensión preacoplamiento de alternadores.....</i>	<i>29</i>
<i>Ilustración 15: USB-TTL para la carga del programa en la placa stm32.....</i>	<i>30</i>
<i>Ilustración 16: Ficha técnica de la placa stm32f03RC</i>	<i>32</i>
<i>Ilustración 17: Ejemplo de adaptación de frecuencia de trabajo.</i>	<i>33</i>
<i>Ilustración 18: Gestor de tarjetas de Arduino.....</i>	<i>34</i>
<i>Ilustración 19: Descargar el paquete stm32.....</i>	<i>35</i>
<i>Ilustración 20: Preferencias para modificar el gestor URLs adicional de tarjetas en Arduino.....</i>	<i>36</i>
<i>Ilustración 21: Selección de nuestro modelo específico stm32.....</i>	<i>36</i>
<i>Ilustración 22: Logo de Arduino.....</i>	<i>37</i>
<i>Ilustración 23: Sketch básico para programación Arduino.....</i>	<i>38</i>
<i>Ilustración 24: Flujograma: Control hasta la segunda condición.</i>	<i>40</i>
<i>Ilustración 25: Flujograma: Hasta el acoplamiento.....</i>	<i>41</i>
<i>Ilustración 26: Secuencia directa de fases.....</i>	<i>43</i>
<i>Ilustración 27: Desfase entre la señal de red y la generada.</i>	<i>43</i>
<i>Ilustración 28: Referencia de tensión.</i>	<i>44</i>
<i>Ilustración 29: Divisor de tensión para lectura de valores.....</i>	<i>45</i>
<i>Ilustración 30: Filtros paso bajo compuesto por elementos pasivos.</i>	<i>49</i>
<i>Ilustración 31: Descomposición de señal cuadrada en varias senoidales, mayor cantidad de señales mejor descripción de la onda.....</i>	<i>50</i>
<i>Ilustración 33: Señales de red y generada filtrada.</i>	<i>52</i>
<i>Ilustración 32: Señal de red y generada sin filtrar.....</i>	<i>52</i>
<i>Ilustración 34: Conexión Pull Up y Pull Down, respectivamente.....</i>	<i>53</i>

<i>Ilustración 35: Valores típicos de tensión en LEDs.....</i>	<i>54</i>
<i>Ilustración 36: Conexión de leds en placa.....</i>	<i>55</i>
<i>Ilustración 37: Patillaje de las IGBT.</i>	<i>56</i>
<i>Ilustración 38: IGBT k30N60 con la que se opera.</i>	<i>57</i>
<i>Ilustración 39: Tiempos de disparo de la IGBT.....</i>	<i>57</i>
<i>Ilustración 40: Cortocircuitado de una resistencia con el programa Arduino.</i>	<i>63</i>
<i>Ilustración 41. Señal PWM para 250 valores.....</i>	<i>64</i>
<i>Ilustración 42: Señal PWM para 20 valores.....</i>	<i>64</i>
<i>Ilustración 43: Driver elevador inversor para excitación del IGBT.....</i>	<i>65</i>
<i>Ilustración 44: Zonas de saturación MOSFET.</i>	<i>66</i>
<i>Ilustración 45: Driver de control de MOSFET.....</i>	<i>67</i>
<i>Ilustración 46; Implementación real del control de los IGBT.</i>	<i>68</i>
<i>Ilustración 47: Valores de salida para una tensión de 50,5V y salida de 45,2V para un pulso de 245 valores.</i>	<i>68</i>
<i>Ilustración 48: Esquema de comprobación de variación de tensión.....</i>	<i>69</i>
<i>Ilustración 49: Valores de salida para un pulso de 250 y 10, respectivos.....</i>	<i>69</i>
<i>Ilustración 50: Señal modificada por ángulo de disparo de 240 valores.</i>	<i>70</i>
<i>Ilustración 51: Definición de los Pins de trabajo.....</i>	<i>71</i>
<i>Ilustración 52. Función setup() de única ejecución.</i>	<i>72</i>
<i>Ilustración 53: Punto de referencia continua para la lectura de la señal de red y generada.</i>	<i>73</i>
<i>Ilustración 54: Lectura de valores para uso referencial.....</i>	<i>74</i>
<i>Ilustración 55: Llamada de la función "Leer Valor".</i>	<i>74</i>
<i>Ilustración 56: Señal PWM.</i>	<i>76</i>
<i>Ilustración 57: Función Startup en Arduino.</i>	<i>78</i>
<i>Ilustración 58: Segundo bloque de reiteración en Arduino.....</i>	<i>79</i>
<i>Ilustración 59: Lectura de frecuencia y correspondiente modificación de variables en Arduino.....</i>	<i>80</i>
<i>Ilustración 60: Función de lectura de tensión en Arduino.....</i>	<i>82</i>
<i>Ilustración 61: Comienzo de la primera condición en Arduino.</i>	<i>83</i>
<i>Ilustración 62: Modificación del pulso correspondiente al margen de error en el que operemos.....</i>	<i>84</i>
<i>Ilustración 63: Proceso de obtención de la F.E.M. en Arduino.....</i>	<i>85</i>
<i>Ilustración 64: Modificación de DUTY 2 tras obtener el parámetro Contr_puls_V en Arduino.</i>	<i>86</i>
<i>Ilustración 65: Función de comprobación de secuencia en Arduino.</i>	<i>87</i>
<i>Ilustración 66: Función de control de desfase nulo en Arduino.</i>	<i>88</i>
<i>Ilustración 67. Función de seguridad implementada en Arduino.</i>	<i>89</i>
<i>Ilustración 68: Alternador de baja potencia.</i>	<i>90</i>
<i>Ilustración 69: Motor cc de baja potencia.....</i>	<i>91</i>
<i>Ilustración 70: Sistema de baja potencia, alternador alineado con el motor cc.....</i>	<i>93</i>
<i>Ilustración 71: Reóstatos externos, izquierda para el alternador derecha para el motor cc.....</i>	<i>94</i>

<i>Ilustración 72: Conexionado completo del sistema de sincronización automático.</i>	95
<i>Ilustración 73: Seccionadores para la lectura de la tensión generada y red.</i>	96
<i>Ilustración 74: Sincronoscopio de lámparas.</i>	97
<i>Ilustración 75: Dispositivo de alimentación de la stm32.</i>	98
<i>Ilustración 76: Toma de tensión para motor cc y excitatriz del alternador.</i>	98
<i>Ilustración 77: Cortocircuitado de la resistencia.</i>	103
<i>Ilustración 78: Driver de control del transistor.</i>	103
<i>Ilustración 79: Data-Sheet IGBT 1.</i>	109
<i>Ilustración 80: Data-Sheet IGBT 2.</i>	110
<i>Ilustración 81: Data-Sheet IGBT 3.</i>	111
<i>Ilustración 82: Data-Sheet BJT PN2222A, 1.</i>	112
<i>Ilustración 83: Data-Sheet BJT PN2222A, 2.</i>	113
<i>Ilustración 84: Data-Sheet STM32F091RC, 1.</i>	114
<i>Ilustración 85: Data-Sheet STM32F091RC, 2.</i>	115
<i>Ilustración 86: Data-Sheet STM32F091RC,4.</i>	116
<i>Ilustración 87: Data-Sheet STM32F091RC, 3.</i>	116

Índice de planos

<i>Plano 1: Conexión de las máquinas eléctricas, junto a la toma de lectura.</i>	104
<i>Plano 2: Conexiones STM32.</i>	105
<i>Plano 3: Drivers y filtros de lectura y control.</i>	106
<i>Plano 4: Flujograma primera parte.</i>	107
<i>Plano 5: Flujograma segunda parte.</i>	108

1. Introducción.

1.1. Objetivos.

El propósito de este Trabajo de Fin de grado es el desarrollo e implementación de un sistema de acoplamiento y sincronización automático a red de potencia infinita, desde la lectura de los valores actuales de interés hasta la adaptación del control automático de las condiciones expuestas para tal fin:

- *Tensión generada eficaz con valor idéntico al de red.*
- *Misma secuencia de fases.*
- *Frecuencia generada igual o levemente mayor a la de red.*
- *Desfase entre la tensión de red y la generada nula.*

Trataremos el desarrollo del sistema con un enfoque didáctico para tratar de explicar todos los pasos y problemas que se han presentado, que permitan en un futuro poder realizar el mismo montaje sin necesidad del alumno en labor del desarrollo de este trabajo fin de grado.

El sistema de programación seleccionado es un microcontrolador stm32 en base Arduino por la familiaridad del alumno. Este sistema de programación se encargará de obtener todos los parámetros de lectura y respuesta necesarias.

La automatización del proceso de sincronización y acoplamiento se trabajará sobre dos bancos de motores (sistemas de alta y baja potencia) a los que se van a referir durante el desarrollo del trabajo.

Los diferentes parámetros y planos que se van a tratar de referencia parten del banco de información presente en el Trabajo Fin de Grado en Electrónica Industrial, ***“Implementación de un Sistema Automático de Sincronización y Conexión de Alternadores a una Red de Potencia Infinita”***, desarrollado por Miguel Ángel Chaikhiev Chaikhieva.

1.2. Fases de desarrollo.

Partiremos desde la explicación teórica del funcionamiento de los motores con los que operamos, planteado cuales son las diferentes variables eléctricas/mecánicas de interés en el control para realizar la maniobra de sincronización y acoplamiento. Obtendremos la adaptación individual de cada metodología de control y, finalmente, realizaremos el desarrollo en conjunto de ambos sistemas.

Seguidamente procederemos a la observación de los resultados obtenidos, precisión de las variables de control junto a la verificación de su funcionamiento.

1.3. Resumen.

Los elementos encargados de generar la mayor parte de la energía consumida actualmente son máquinas eléctricas rotativas denominadas generadores síncronos. El control de esta generación para adaptarse a las exigencias externas o internas del sistema se presentan como objetivo fundamental en la actualidad pues la electricidad se presenta en casi todos los aspectos cotidianos actuales, siendo un bien casi esencial en la mayor parte del mundo. La demanda basada en el uso de los diversos avances tecnológicos desde su descubrimiento nos muestra la gran importancia en el control de la generación, distribución y consumo que nos permita obtener unos sistemas económicos y rentables que otorguen seguridad con un bajo nivel de contaminación.

La generación eléctrica contiene un amplio margen de opciones que han de ser controlados y regulados, por ejemplo, en centrales de combustión, necesitamos controlar el flujo de combustible para obtener una variación de potencia neta, en centrales hidráulicas el aporte de caudal, etc. Es debido a esta razón que la correcta actuación sobre estos sistemas nos permitirá responder de forma rápida y segura a la variación de la demanda diaria con

cierto margen de seguridad, teniendo un impacto vital en el funcionamiento de la sociedad.

El sistema de generación eléctrica en el que se basa este TFG, los alternadores, van a ser impulsados por una máquina de corriente continua, simulando el comportamiento de cualquier fuente de energía mecánica externa generadora de un par/velocidad en el eje de acoplamiento, pudiendo ser una presa hidroeléctrica, una central de combustión, un molino eólico, etc.

Para otorgar esa energía generada por el alternador a la red, denominada “*Red de Potencia Infinita*”, pues contiene infinidad de generadores/consumidores eléctricos manteniendo el valor de la tensión y frecuencia constantes (teóricamente), se deben de cumplir unos requisitos automatizables, en lo que refiere al control de la fuente de energía mecánica, par/velocidad del motor de corriente continua, junto al alternador que deseamos acoplar a red para no dañar el material ni poner en peligro el material de uso y al técnico durante su operación.

2. Motor primario: Motor de corriente continua, funcionamiento y control.

2.1. Estudio del control de velocidad de motores de CC.

Para la obtención de energía externa, como hemos comentado, usaremos un motor de corriente continua con excitación compound, por lo tanto, necesitamos controlar los diferentes parámetros que nos permitan regular la potencia mecánica ofrecida junto a las variables que esto conlleva, el par y la velocidad de rotación.

2.1.1. Principio de funcionamiento.

Toda máquina puede funcionar como motor o como generador dependiendo de los diferentes parámetros de interés. El funcionamiento como motor de cc se basa en la alimentación del rotor y estator mediante corriente continua. El estator genera un campo magnético fijo entre las zapatas polares, por lo tanto, cuando se alimenta el rotor con corriente continua, creando otro campo magnético fijo que gira con el eje y auxiliándose de los devanados auxiliares para favorecer la estabilidad del flujo magnético, produce una fuerza electromotriz inducida en el estator y, en consecuencia, el par junto a la velocidad de giro del rotor.

Como cualquier material ferromagnético, se produce histéresis magnética tras retirar un material de una zona de inducción, esta conserva parte de la señal magnética produciendo un magnetismo remanente. Este efecto produce en los motores una pérdida en forma de calor del núcleo magnético durante su funcionamiento como motor o generador.

2.1.2. Excitatriz del motor de corriente continua.

Los circuitos inductor-inducido se pueden conectar de diferentes formas con repercusiones a su funcionamiento. En nuestro caso, el motor de c.c. trabaja con excitación compuesta o compound larga esto es, cada polo inductor tiene dos bobinas de excitación, en serie y paralelo al circuito inducido para el compound, el conjunto se conecta a una única red de corriente continua de alimentación proporcionando propiedades combinadas de excitación en serie y paralelo.

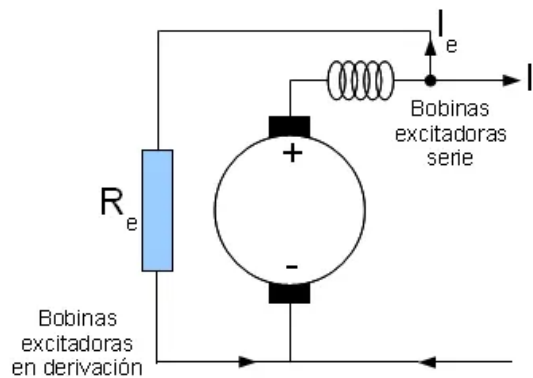


Ilustración 1: Sistema derivación compund larga.

Sabemos que las escobillas recogen las f.e.m.s inducidas durante la vuelta entera de rotor, funcionando como un rectificador mecánico de la tensión del inducido, es decir, permite obtener una f.e.m. siempre con el mismo sentido. Obtenemos por lo tanto la relación entre las características mecánicas internas del motor (ondulado o imbricado, junto a la cantidad de números de polos, etc.), la velocidad de giro y el valor del flujo magnético.

$$E = \frac{n}{60} * Z * \frac{P}{c} * \varphi = K_e * n * \varphi \quad (1)$$

Dónde:

E es la fuerza electromotriz inducida, se mide en V.

n es la velocidad de giro del rotor, se mide en rad/s.

Z es el número de conductores.

P es el número de polos.

C es el número de circuitos derivados.

K_e es la constante mecánica interna del motor de cc.

φ es el flujo magnético, se mide en Wb.

La ecuación 1 se puede aplicar siempre que se trabaje dentro de la zona no saturada del motor, en la que se mantiene la proporcionalidad lineal entre el flujo magnético y la corriente de excitación. Fuera de esta zona la curva varía y no sigue una proporcionalidad constante para valores elevados.

Sabemos que el flujo electromagnético depende del valor de la corriente de excitación que recorre el devanado inductor, por lo tanto, tenemos que considerar la inductancia de ambas bobinas y el acoplamiento entre estas, el estudio del sistema de la ilustración 1 se basa en la aplicación de Kirchhoff y funcionamiento como motor en cc.

Mediante el valor de la corriente de excitación del inductor serie/paralelo (" I_e ") podemos ir variando la velocidad de giro del motor de corriente continua hasta el valor que especifiquemos y de una manera proporcional hasta cierto valor de corriente de excitación por en la cual la máquina se satura.

En resumen, la potencia mecánica generada por el motor cc que vamos a aprovechar se controla mediante el giro del rotor del motor de cc, directamente sobre el parámetro que induce una mayor o menor velocidad sobre el eje, en este caso, la corriente de excitación del inductor en serie y paralelo.

2.2. Selección de un sistema de control de velocidad para motores CC.

Los modelos actuales de control de velocidad de motores cc se centran en el control de la tensión de alimentación cc que recae en el devanado inductor mediante tiristores de potencia como indica la imagen 2.

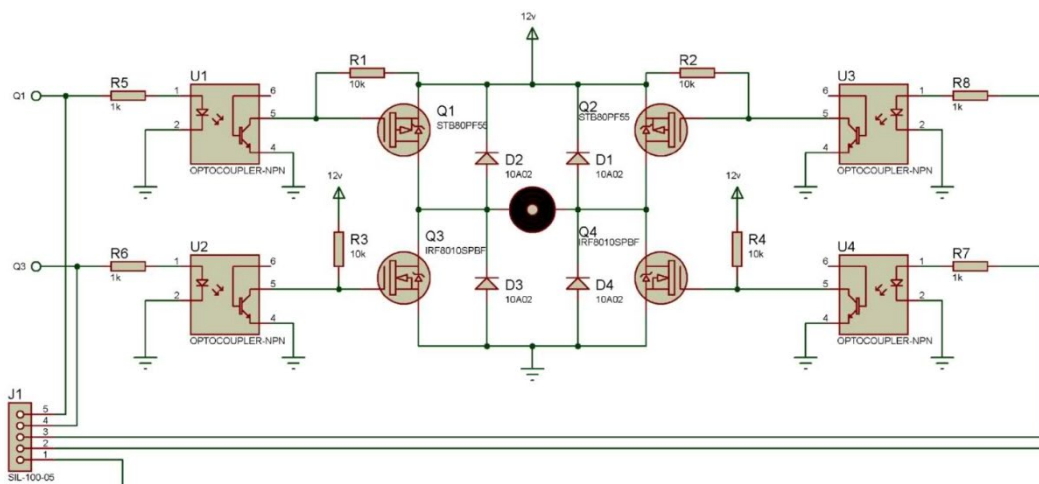


Ilustración 2: Control del motor cc, puente en H.

Observamos un control en puente H de un motor cc, con objetivo de invertir el sentido de giro para acelerar o desacelerar según se requiera. La idea que debemos de destacar es la alimentación del motor, dependiendo de la cantidad de tensión que fijemos en el motor cc, variaremos la velocidad independientemente de su sentido de giro, es esta idea de variación de tensión para el control de la velocidad el punto de partida.

Variando la tensión de alimentación variamos la corriente de excitación del inductor y en consecuencia la velocidad de giro del motor, pero si trabajamos a tensión constante y queremos obtener una variación de la corriente de excitación podemos tratar de disponer una resistencia cortocircuitada, es decir, variando/controlando el valor de la resistencia en serie al devanado inductor podremos controlar la corriente de excitación del motor y obtener nuestro objetivo, variar la velocidad, ilustración 3.

El cortocircuitado se realiza con transistor electrónico, la opción principal fue el estudio de un MOSFET “FDB2552”, para operar con 150V de tensión y hasta 37 A de corriente, entraremos en más detalle posteriormente.

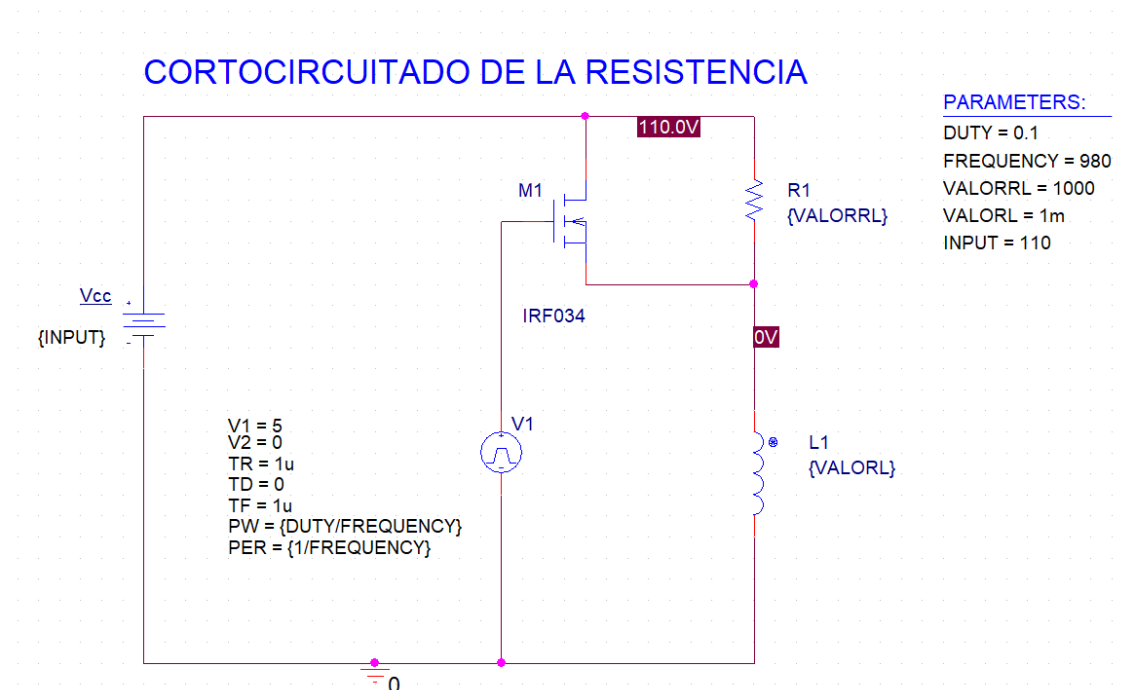


Ilustración 3: Cortocircuitado de una resistencia.

3. Excitatriz de corriente continua, funcionamiento y control.

El sistema de excitación del alternador se corresponde a un motor cc en esquema independiente, esto es, los circuitos del inductor y del inducido se alimentan de fuentes ajenas. Interesa un flujo en el entrehierro con valor máximo durante el arranque para desarrollar el par específico con la mínima cantidad de corriente, adicionalmente necesitaremos presentar un valor reducido de resistencia en paralelo para disminuir las pérdidas.

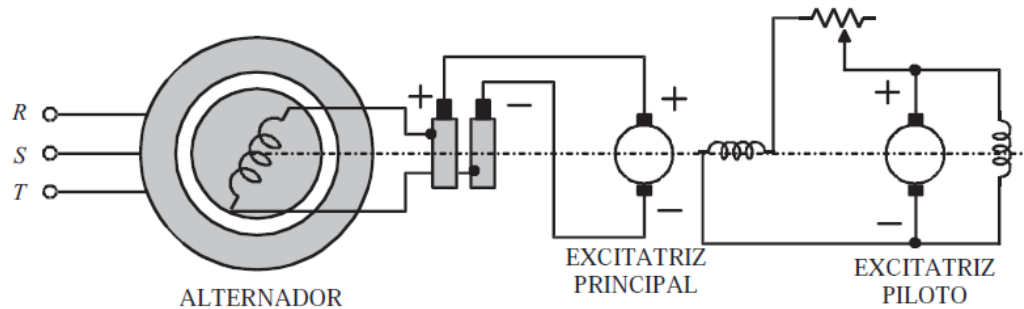


Ilustración 4: Excitación del alternador con excitatriz piloto.

Antes del acoplamiento, el alternador funciona en vacío, esto es, sin carga, sin corriente en el inducido ni potencia desarrollada, por lo tanto, el único devanado por el que circula alguna corriente es el devanado inductor o de excitación. El campo magnético de la máquina es debido únicamente a la corriente de excitación. Cuando gira el rotor a velocidad de sincronismo, el campo magnético creado por la corriente continua que circula por el inductor gira con él. Por lo tanto, las fases del inducido, dispuestas en el estator del alternador, ven un campo magnético variable en el tiempo cuya frecuencia viene dada por la relación de la velocidad de giro-frecuencia y cuya tensión inducida sobre estas vienen relacionadas por esta misma frecuencia, estructura mecánica interna y la inducción del sistema de excitación.

En la zona de trabajo lineal la relación entre la corriente de excitación y el flujo electromagnético es constante, la variación de esta corriente de excitación repercutirá sobre la F.E.M. generada y su respectivo control nos permitirá cumplir la condición de sincronización y acoplamiento en igualdad de tensión de red.

En resumen, la fuerza electromotriz generada por la excitatriz del alternador se controla mediante la corriente de excitación del devanado inductor, su valor depende directamente de la tensión de excitación que dispongamos pues la resistencia en el inductor es constante, en este caso, la tensión de excitación del alternador es nuestro parámetro de control.

3.1. Selección de un sistema de control de la excitatriz.

Los modelos de control actuales de la excitatriz del alternador se basan en el control de la tensión que recae en los bornes de la excitatriz y del proceso para su obtención. La excitación mediante excitatrices de corriente alterna con rectificadores son un ejemplo del control de motores síncronos, evita el uso de colector de delgas para la excitatriz principal como se puede ver en la ilustración 5.

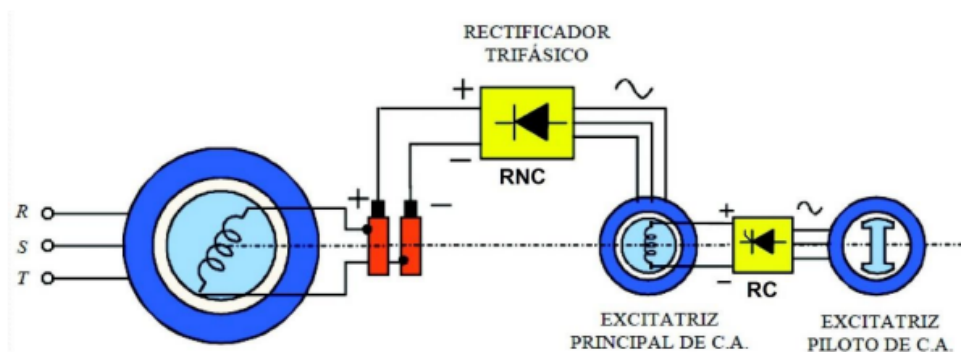


Ilustración 5: Rectificadores trifásicos para el control de la excitatriz de los alternadores.

La idea es variar el ángulo de disparo de los tiristores para variar el módulo de la tensión de excitación en el alternador, de tal manera, aumentando/disminuyendo el ángulo de disparo ofreceremos una mayor/menor tensión y, en consecuencia, una velocidad respectiva.

Este proyecto tratará el control de la tensión de alimentación de la excitatriz como enfoque principal mediante el cortocircuito de una resistencia dispuesta en serie del sistema de excitación del alternador, el cortocircuito de dicha resistencia, al igual que en el motor de cc, nos permitirá variar la tensión que recae en el sistema de excitación, y producir una f.e.m. específica

Controlando la apertura y cierre, es decir, el pulso, variaremos la resistencia adicional en la excitación.

$$\delta = \frac{T_{on}}{T_{off} + T_{on}} = \frac{T_{on}}{T} \quad (2)$$

Donde

δ es el pulso de la apertura y cierre.

Ton es el tiempo de apertura, se mide en s.

Toff es el tiempo de cierre, se mide en s.

T es el periodo de trabajo, se mide en s.

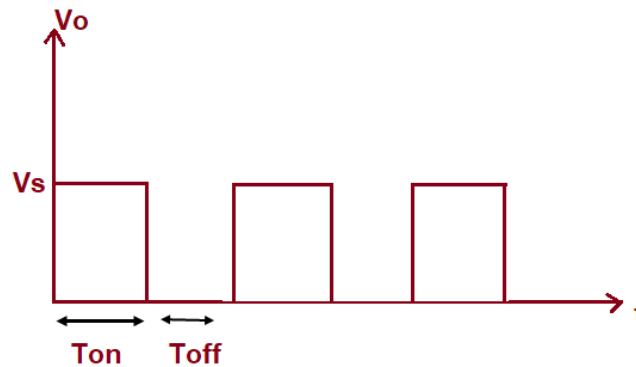


Ilustración 6: Señal PWM, pulse width modulation.

Observando el esquema de la ilustración 5 y 6, la señal solo transcurre cuando el transistor este polarizado y encendido para los intervalos positivos, lo contrario para los negativos que ofrecen una señal del transistor cerrada. Podemos definir el valor promedio de salida en función del pulso de onda, mediante.

$$V_{o, avg} = \frac{1}{T} * \int_0^{Ton} V_s dt + \frac{1}{T} * \int_0^{Toff} 0 * dt = \frac{Ton}{T} * V_s = \delta * V_s$$

Debemos considerar que siempre y cuando se realice una apertura y cierre en un periodo constante trabajaremos con armónicos que pueden perturbar el funcionamiento del sistema, en apartados posteriores se analizan y se pondrán condiciones para su correspondiente disipación.

4. Sistema de control automático de sincronización del alternador a la red. Magnitudes eléctricas y mecánicas involucradas.

4.1. Estudio de la maniobra de sincronización y acoplamiento de alternadores a una red de potencia infinita.

4.1.1. Condiciones de acoplamiento.

Las cuatro condiciones que debemos de comprobar para poder realizar la maniobra de acoplamiento a red y que nos aseguren que estemos sincronizados con la red son las siguientes:

- Tensión generada eficaz con valor idéntico al de red.
- Misma secuencia de fases.
- Frecuencia generada igual o levemente mayor a la de red.
- Desfase entre la tensión de red y la generada nula.

La **secuencia directa** se corresponde a un mismo sentido de giro y el cambio de fases en los bornes produce la inversión de este, a su vez, la inversión de la secuencia generada.

La **tensión generada eficaz** con valor símil al de red junto a la **igualdad de frecuencias** se debe de comprobar en cada maniobra de sincronización, un elemento que se utiliza para su comprobación es el sincronoscopio.

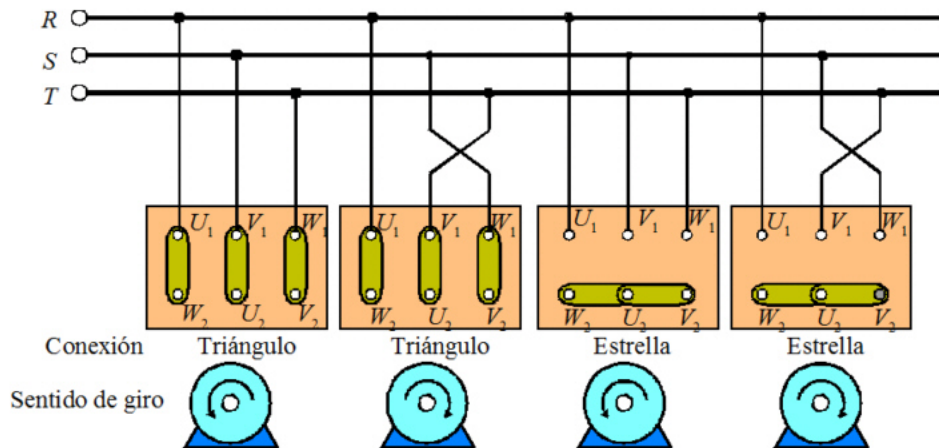


Ilustración 7: Conexión de bornes en un motor trifásico para obtención de secuencia de fases y de giro.

El sincronoscopio se compone por tres lámparas, colocadas en los vértices de un triángulo equilátero dispuestos para comprobar las condiciones de la secuencia de giro, tensión eficaz y frecuencia. La frecuencia de parpadeo aumenta según se aproximen las frecuencias generadas y de red. Si la conexión del sincronoscopio es de led-encendido, se apagarán los leds según trabajemos con una tensión similar entre la red y la f.e.m. generada y coincidan las frecuencias.

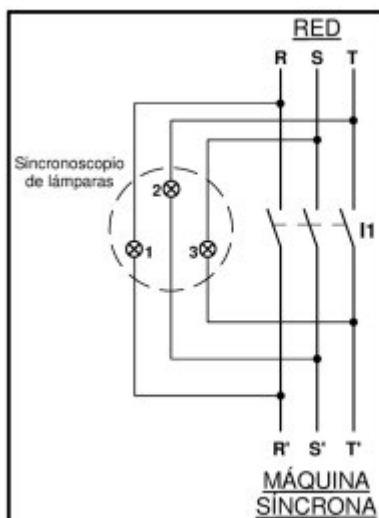


Ilustración 8: Sincronoscopio de lámparas.

Una vez no brillen los leds, significa que hemos cumplido condiciones de generación de tensión y frecuencia y procedemos a la comprobación del desfase entre la tensión de red y la generada.

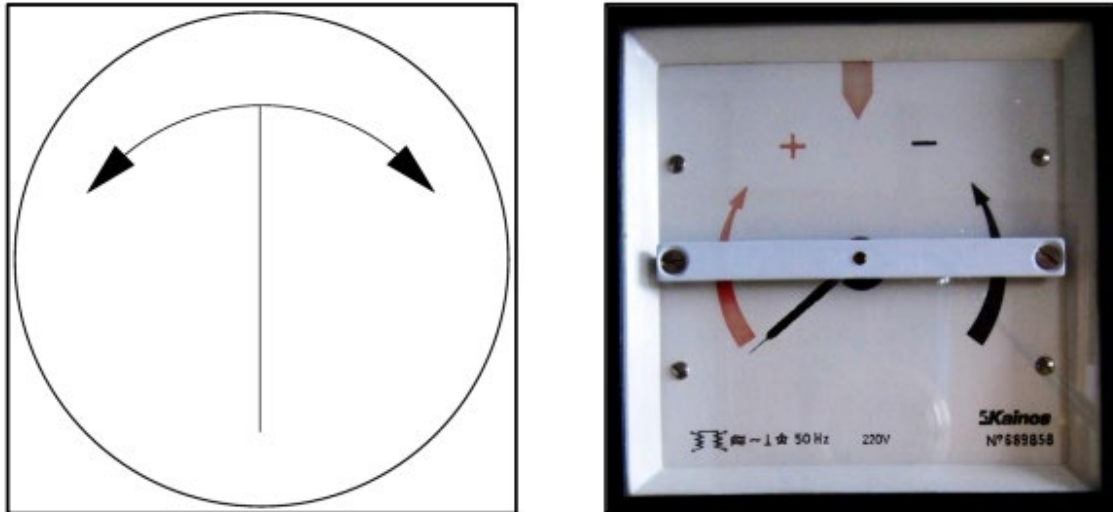


Ilustración 9: Sincronoscopio de aguja.

El **desfase entre tensión de red y generado** se suele medir mediante un sincronoscopio de aguja. Es un motor que gira una aguja de referencia que tiene devanados de estator y rotor para conectarlos a la tensión de red y de fase. La flecha indicará igualdad de señales o desfase nulo cuando ambas señales estén sincronizadas, permitiendo realizar la maniobra de acoplamiento.

En resumen, debemos de medir dos fases generadas del alternador y una de red como mínimo para realizar el proceso de sincronización.

4.1.2. Acoplamiento sin cumplimiento de las etapas de sincronización.

Debemos de comentar las diferentes situaciones que pueden darse en caso de realizar la conexión a red sin cumplir las condiciones exigidas comentadas.

Generalmente, antes del acoplamiento el campo magnético del generador eléctrico es producido por la combinación de las fuerzas magnetomotrices producidas por los devanados del inductor de nuestro sistema. Pero tras la conexión del circuito, en el estátor aparece una corriente que crea una fuerza electromotriz de reacción que se combina con la del campo original produciendo una densidad de flujo resultante en el entrehierro.

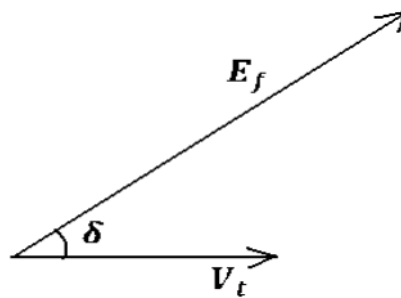


Ilustración 10: Ángulo de desfase entre tensión de red y generada.

Por lo tanto, el estudio de este flujo resultante se realiza mediante la descomposición de ambos efectos en dos impedancias: cuadratura y directo.

El eje directo es perpendicular al voltaje interno mientras que el de cuadratura se fija en la misma dirección que el del eje directo pero adelantado 90° , ilustración 11. El estudio de estos ejes permite realizar el control post-acoplamiento del sistema, pues el voltaje en bornes de salida medido es el resultado de la tensión generada y la tensión que recae en las resistencias de los devanados del estator, de la reactancia síncrona directa y la reactancia en cuadratura. Representando esta operación obtenemos el diagrama fasorial de un generador síncrono.

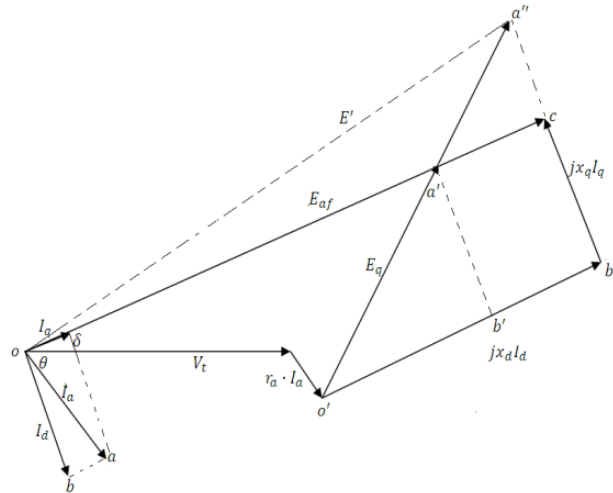


Ilustración 11: Esquema fasorial de una máquina síncrona con carga.

* En caso de trabajar con una máquina de polos lisos, la impedancia de cuadratura y de eje directo son iguales, se trabaja con una impedancia síncrona general.

Una vez comprendidos el funcionamiento post-acoplamiento, podemos definir las ecuaciones que definen la potencia eléctrica activa y reactiva. Vamos a desarrollar las ecuaciones para una máquina de polos simples para facilitar la comprensión de las situaciones específicas.

La **potencia activa** viene definida por:

$$P = \frac{3 * V * E_o}{X_s} * \sin \delta \quad (3)$$

Donde

P es la potencia activa, medida en W.

V es la tensión de red, en V

Eo es la tensión generada por el generador, en V.

Xs es la reactancia síncrona del alternador, en caso de polos salientes se divide en Xd (impedancia de eje directo) y Xq (impedancia de cuadratura), en Ohm.

δ es el ángulo de potencia o desfase entre tensión de red y generada, en grados o radianes eléctricos.

La **potencia reactiva** viene definida por:

$$Q = \frac{3 * V * E_o}{X_s} * \cos \delta - \frac{3 * V^2}{X_s} \quad (4)$$

Donde

Q es la potencia reactiva, medida en VAr.

V es la tensión de red, en V

Eo es la tensión generada por el generador, en V.

Xs es la reactancia síncrona del alternador, en caso de polos salientes se divide en Xd (impedancia de eje directo) y Xq (impedancia de cuadratura), en Ohm.

δ es el ángulo de potencia o desfase entre tensión de red y generada, en grados o radianes eléctricos.

Observando la ecuación 4, tenemos dos términos, el primero se corresponde a la potencia reactiva generada por el alternador, mientras que el segundo término es la potencia reactiva disipada por la reactancia síncrona. En el momento en que el primer término sea menor que el segundo, el generador consumirá potencia reactiva en vez de producirla. En consecuencia, al aumentar la corriente de excitación del alternador, aumenta la F.E.M. y seguidamente el ángulo de desfase produciéndose un aumento de energía reactiva.

- **Controlando la corriente de excitación del generador, controlamos la potencia reactiva.**

Al aumentar la potencia mecánica del eje, produciremos un aumento de la potencia activa generada.

- ***Controlando la corriente inductora del motor de cc controlamos la potencia activa generada.***

De forma análoga, si la tensión generada por el alternador es inferior a la tensión de red, el ángulo de desfase entre ambas se eleva y, auxiliándonos de la ecuación 4, observamos como se consume potencia reactiva de la red. Si la tensión eficaz es superior a la de red, el alternador genera la potencia reactiva.

Para entender los efectos de una conexión a tensiones desiguales, debemos de comprender cuales son los criterios para obtener la máxima cantidad de potencia activa del motor.

En caso de realizar una conexión a red con una frecuencia alejada a la referencia explícita, la red produce una aceleración/desaceleración brusca en el motor primario aumentando el par transitorio y todas las pérdidas relacionadas con esta variación, por ejemplo, una aceleración brusca produce un aumento de entrega de potencia activa, pudiéndose dar el caso de sobrepaso de los límites de la máquina y su correspondiente destrucción.

En la ilustración 12 se observan los límites típicos de operación de una máquina específica para cierta cantidad de potencia activa y potencia reactiva, el trabajo nominal (funcionamiento operacional máximo) permite trabajar con un valor de corriente reactiva máximo específico, si se sobrepasa este límite podemos llegar a sobrecalentar la máquina y producir su respectiva destrucción. El mismo caso para el funcionamiento de una potencia activa superior, podemos operar con una tolerancia superior a la nominal, pero reduciendo los valores de potencia reactiva y, en consecuencia, una reducción de la corriente de excitación del alternador. En caso de un acoplamiento sobreexcitando en secuencia de fases y potencia, muy probablemente saldríamos de estos límites de operación produciendo la destrucción del sistema con el que estemos trabajando.

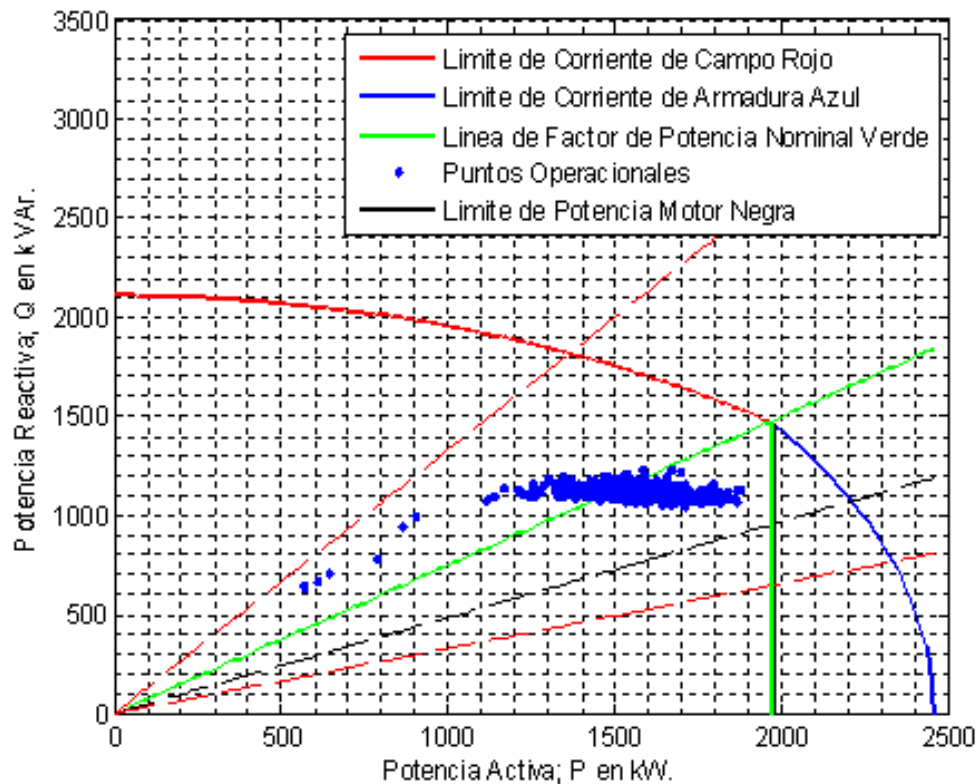


Ilustración 12: Curva de capacidad y puntos operacionales para un alternador específico.

En referencia a la normativa BOE-A-2020-8965, se pueden realizar conexiones a red con diferentes frecuencias durante tiempo específicos, ilustración 13.

Zona	Rango de frecuencias	Periodo de tiempo de funcionamiento
España peninsular	47,5 Hz-48,5 Hz	30 minutos
	48,5 Hz-49,0 Hz	Ilimitado
	49,0 Hz-51,0 Hz	Ilimitado
	51,0 Hz-51,5Hz	30 minutos

Ilustración 13: Periodos de tiempo mínimo durante los que un módulo de generación de electricidad debe ser capaz de funcionar.

Esto nos permite tener una idea del margen de programación de nuestro sistema automatizado, para valores de 48.5Hz a 51Hz, siempre teniendo en cuenta los límites de operación que puede generar nuestro alternador de potencia activa y reactiva.

La ilustración 14 presenta los valores por normativa (BOE-A-2020-8965) que especifican la potencia reactiva límite para el acoplamiento y ajustes para la desconexión automática de modelos de generación de electricidad conectados en red de distribución radial a potencia máxima nominal. El rango de tensión se debe limitar a 0.95 pu como mínimo a 1.05 pu como máximo.

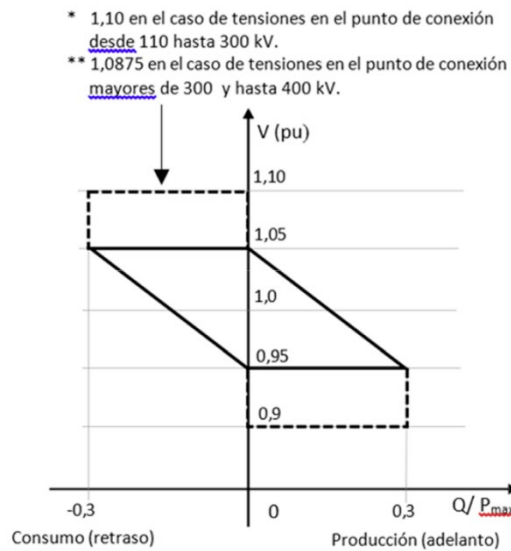


Ilustración 14: Rango de potencia reactiva por tensión preacoplamiento de alternadores.

4.2. Selección de un equipo que controle la maniobra de sincronización.

Vamos a proceder al análisis de las selecciones realizadas por el alumno comentando las ventajas y desventajas que se han ido identificando en el proceso.

4.2.1. Microcontrolador stm32 con Arduino.

Recurrimos a los microcontroladores stm32 con viabilidad de uso en Arduino. La comodidad y utilidad permite usar este entorno sencillo y mundialmente conocido, permitiéndonos hacer uso de las diferentes herramientas creadas por la comunidad de Arduino que podemos incluir mediante la localización de sus respectivas librerías y sencillas modificaciones que iremos comentando.

Los tres métodos de programar el STM32 con entornos basados en Arduino son mediante un *conversor USB-TTL*, un programador *ST-Link* o directamente a través del *puerto micro-USB*.

- El ST-Link v2 es un programador/debugger construido por ST Microelectronics que permite programar y depurar directamente los procesadores STM32 y STM8, implementando “Single Wire Interface Module” y “Serial Wire Debugging” para comunicar con los procesadores de la placa de desarrollo.

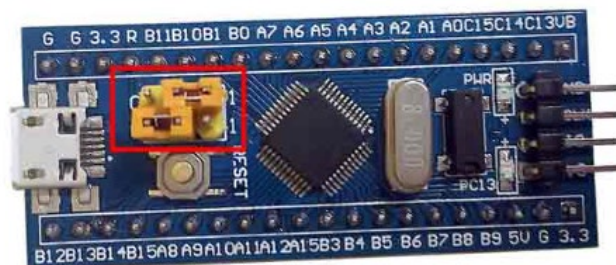


Ilustración 15: USB-TTL para la carga del programa en la placa stm32.

- Los conversores USB-TTL son dispositivos de intercomunicación que permite conectar directamente la placa stm32 con el programa Arduino, ilustración 15.
- Conexión directa mediante puerto micro-USB es la intercomunicación mediante cable micro-USB conectado a la placa stm32 y el ordenador para intercomunicar con el programa de desarrollo Arduino.

Nosotros realizaremos la programación directa mediante conexión al puerto micro-USB.

4.2.2. Interpretación de la ficha técnica stm32f091RC

Antes del comienzo de la programación debemos de tener en cuenta los diferentes conceptos con los que vamos a trabajar, tales como la frecuencia de trabajo y la resolución de nuestra placa. Esta información la obtenemos de la ficha técnica de nuestra placa stm32, obtenida de la propia empresa, como se observa en el Data-Sheet del stm32.

Los parámetros de interés son la resolución del ADC y de los Timers de trabajo, para poder operar con cierta cantidad de bits en la captación de señales analógicas, lo que nos permite un mayor o menor margen de precisión.

Fijándonos en la ficha técnica observamos en la ilustración 16 que los Timers tienen una resolución general de 16 bits, es decir, para una frecuencia de oscilación interna de 32MHz obtenemos a la salida una señal de $32\text{MHz}/2^{16} = 488.28 \text{ Hz} \sim 480\text{Hz}$, con una resolución de 2^{16} (65536) valores numéricos, la pre-configuración de la stm32 es a 1kHz.

La resolución del ADC (convertidor analógico digital) es de 12 bits, es decir, que para un margen de 0 a 3.3V de operación de entrada obtenemos un margen numérico de 0 a 2^{12} (0 a 4096) frente a los 1024 de la placa Arduino.

Seguidamente necesitamos la resolución para la medida temporal entre señales mediante los timers (Contadores Internos de la placa) que ofrece, en la ilustración 16 observamos que disponemos de 6 timers con diferentes características. En nuestro caso vamos a necesitar 3 timers para modificación del pulso la tensión generada fase R y S junto a la lectura de la frecuencia de red y generada. El valor de referencia son los 50Hz y hacemos uso de 2 timers para la salida PWM, encargados de general las señales de control de los transistores.

3.14 Timers and watchdogs

The STM32F091xB/xC devices include up to six general-purpose timers, two basic timers and an advanced control timer.

Table 7 compares the features of the different timers.

Table 7. Timer feature comparison

Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary outputs
Advanced control	TIM1	16-bit	Up, down, up/down	integer from 1 to 65536	Yes	4	3
General purpose	TIM2	32-bit	Up, down, up/down	integer from 1 to 65536	Yes	4	-
	TIM3	16-bit	Up, down, up/down	integer from 1 to 65536	Yes	4	-
	TIM14	16-bit	Up	integer from 1 to 65536	No	1	-
	TIM15	16-bit	Up	integer from 1 to 65536	Yes	2	1
	TIM16	16-bit	Up	integer from 1 to 65536	Yes	1	1
	TIM17	16-bit	Up	integer from 1 to 65536	Yes	1	1
Basic	TIM6 TIM7	16-bit	Up	integer from 1 to 65536	Yes	-	-

Ilustración 16: Ficha técnica de la placa stm32f03RC

Debemos de considerar que algunos timers no pueden funcionar a la par, es decir, no se puede utilizar el funcionamiento de un Timer para medir el tiempo mediante interrupciones y simultáneamente la generación de la señal PWM. Por lo que necesitamos 3 TIMERS en vez de 2.

La resolución para los timers de uso general es de 32 bits para el TIM2, el TIM3 es de 16 bits, el TIM4 es de 16 bit, TIM 15/16/17 trabajan con 16 bits también.

Los TIMERS de función general trabajan con 16 bits y uno específico con 32 bit (TIM2), es decir, podemos operar con valores de hasta 16 bits de resolución de lectura.

La predefinición de la frecuencia de trabajo depende de tres parámetros, prescaler, counter y clock. Estos tres parámetros se informan en el Data-Sheet correspondiente a cada microprocesador y a excepción de “clock” los otros parámetros se modifican para operar con una frecuencia diferente mediante la siguiente ecuación.

$$f = \frac{\text{Clock}}{(\text{prescaler} + 1) * (\text{counter} + 1)} \quad (5)$$

Donde.

- Clock es la frecuencia del reloj del microcontrolador, en Hz.
- Prescaler es un parámetro para realizar una subdivisión a la entrada del contador/temporizador, este parámetro divide la frecuencia de oscilación en determinadas etapas.
- Counter es un parámetro de referencia para cada bucle en el comparador, es decir, es un temporizador que trabaja hasta la cantidad de “counter”, para un tiempo determinado.

Para su mejor comprensión realizamos un ejemplo en la ilustración 17.

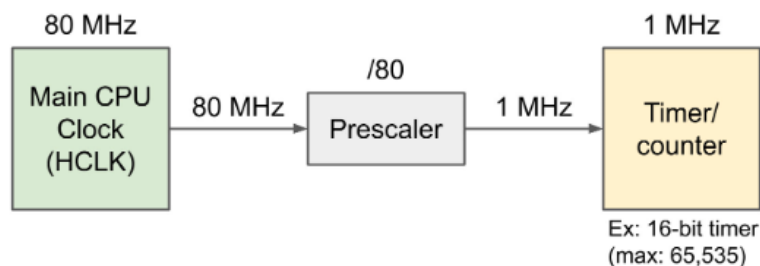


Ilustración 17: Ejemplo de adaptación de frecuencia de trabajo.

Nuestro microprocesador trabaja a una velocidad de 80MHz como máximo, es nuestro punto de partida. Seguidamente queremos reducir la frecuencia de trabajo con un “prescaler” hasta 1 MHz, es decir, “prescaler” 80, seguidamente disponemos el “counter” con cierta cantidad de bits para su resolución, en este ejemplo 16 bit ($2^{16} = 65535$ valores). En resumen, tenemos un reloj interno que envía un “tick” cada microsegundo (1 MHz) con 16

bits de acciones, es decir, $2^{16}/1e6 = 65.5$ milisegundos como medida máxima entre acciones.

Seguidamente debemos de seleccionar algún TIMER que pueda desarrollar adaptarse a los requisitos que hemos impuesto.

4.2.3. Configuración inicial.

El sistema de control se regulará con la ayuda de la placa STM32-F091 RC, Núcleo 64, cortex M0. Su correspondiente configuración y los de cualquier microcontrolador de la misma familia se reduce a tres pasos.

A. Drivers del entorno de trabajo.

Para poder intercomunicar Arduino a la placa de trabajo es necesario descargar de la página oficial del producto los drivers específicos y cargarlos en el sistema de Arduino.

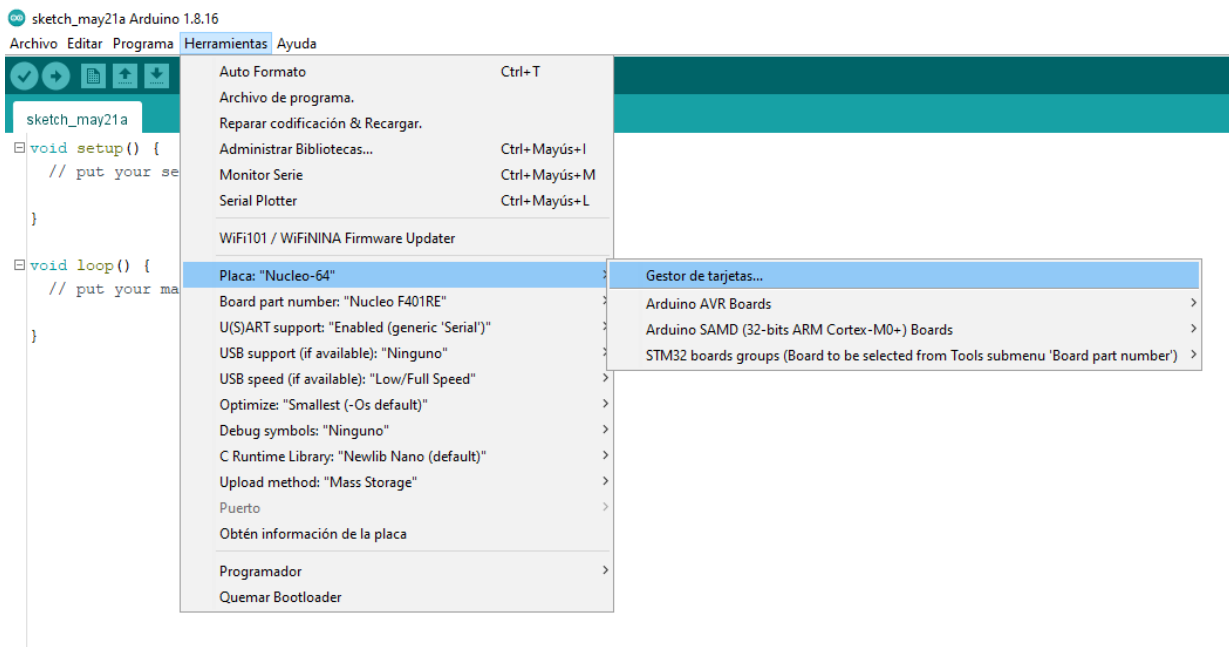


Ilustración 18: Gestor de tarjetas de Arduino.

Abrimos el IDE de Arduino y nos dirigimos al gestor de placas mediante Herramientas, Placa, gestor de Placas. Seguidamente añadimos el soporte para las placas STM32.

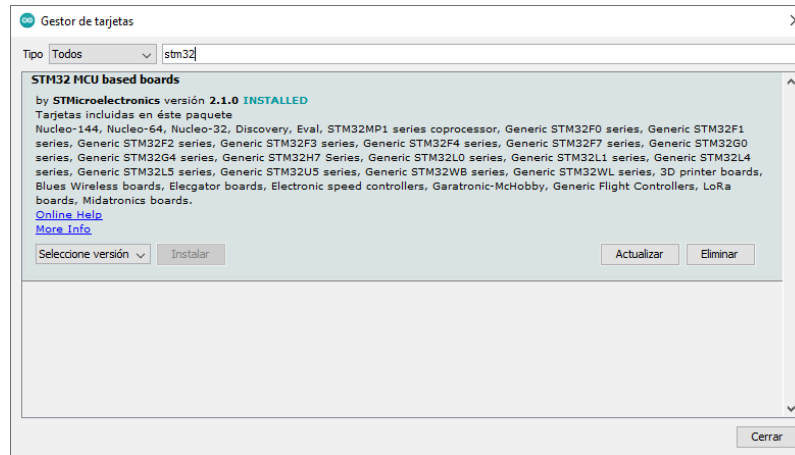


Ilustración 19: Descargar el paquete stm32.

Descargamos el soporte para las placas stm32 para poder localizar nuestra placa específica posteriormente.

B. Configuración de URLs adicionales de tarjetas.

Para poder descargar los archivos debemos de vincular nuestro programa al banco de información que las contiene, en este caso, seguimos los siguientes pasos, Archivos, Preferencias, gestor de URLs adicionales de tarjetas y modificamos el gestor de URLs adicionales de tarjetas.

El enlace que debemos de introducir es el siguiente:

https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json

En ocasiones, si da error el paso 1, debemos de realizar el paso 2 y seguidamente proseguir al 1.

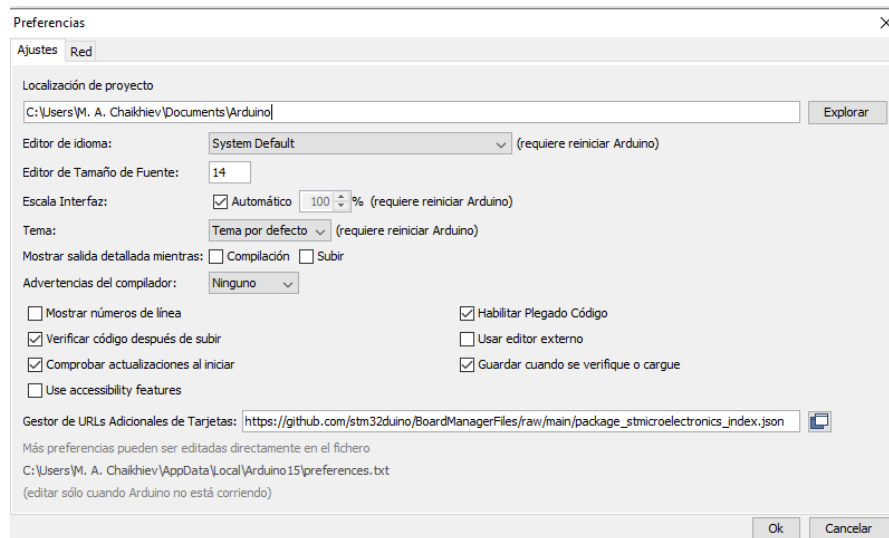


Ilustración 20: Preferencias para modificar el gestor URLs adicional de tarjetas en Arduino.

C. Configuración de la tarjeta Núcleo 64 en el IDE Arduino.

Una vez implementados ambos pasos, volvemos a Herramientas, selección de placa para la elección del núcleo 64 en las placas stm32f.

Tras la **selección del núcleo 64**, seleccionamos nuestra placa específica en herramientas, board part number, STM32FX. Seguidamente Archivos de Arduino para localizar nuestra placa específica, Núcleo F091C. Con estos pasos ya hemos realizado la configuración de la placa y procedemos a la programación en Arduino.

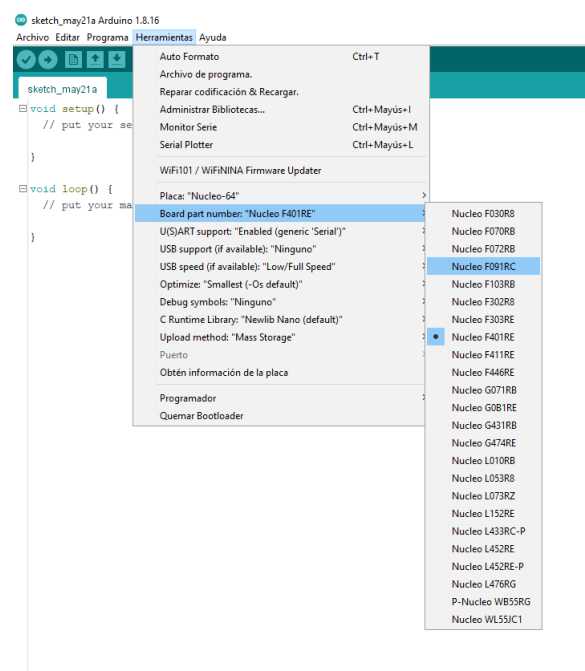


Ilustración 21: Selección de nuestro modelo específico stm32.

4.2.4. Introducción al software Arduino.

La programación de la stm32, placa programable con entradas y salidas digitales/analógicas, mediante el software Arduino es una forma ideal para la inicialización en la automatización o realización de sencillos proyectos en electrónica y robótica. Al tratar con un sistema capaz de obtener información externa, mediante sensores, y realizar acciones correspondientes a las necesidades del programador por medio de actuadores, motores, visualizadores, etc. de manera sencilla y de forma automática, hace de este software la principal elección para la práctica y aprendizaje de operarios y estudiantes, permitiendo la realización de proyectos más complejos como los que estamos dispuestos a realizar la sincronización y el acoplamiento al sistema a la red de potencia infinita.

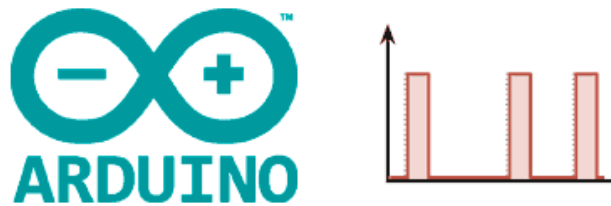


Ilustración 22: Logo de Arduino.

La programación en Arduino se realiza por sus propias placas y microprocesadores, pero nos permite la adaptación para la programación del microcontrolador stm32, mediante un entorno sencillo y potente que incluye las herramientas necesarias para compilar el programa, sobrescribir programas sin mostrar grandes dificultades.

En la ilustración 23, observamos la estructura de un “sketch” básico. Con dos bloques denominados “setup()” y “loop()”. El primero es el bloque de llamada en la inicialización de los proyectos, se inicializa en el comienzo del programa solo una vez y se suele utilizar para la inicialización de parámetros, configuración de pin, librerías, etc.

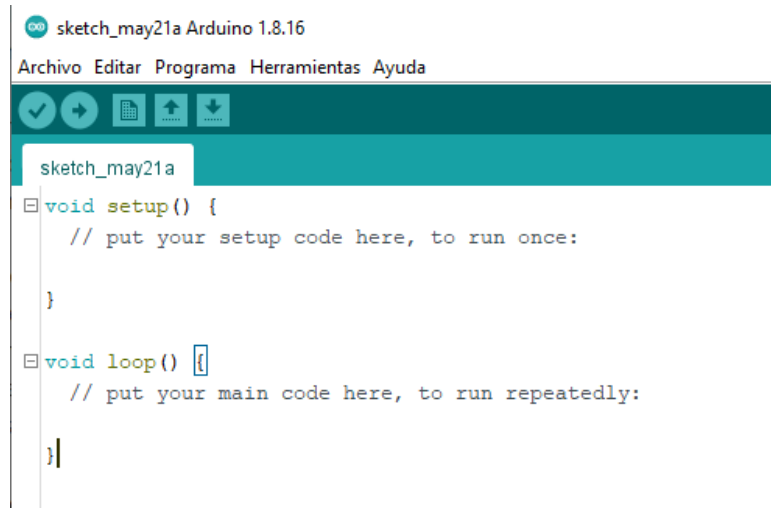


Ilustración 23: Sketch básico para programación Arduino.

El segundo bloque realiza la función consecutiva de ejecución, es decir, constantemente realiza las funciones programadas en el intervalo de sus límites. Es el principal medio de programación para comprobar constantemente los parámetros externos y realizar operaciones de respuesta adecuadas a la finalidad del proyecto.

El lenguaje de programación en el que se basa Arduino es C++ o C, no es literalmente a lo que debería de ser el lenguaje, si no una mezcla o adaptación que proviene de “avr-libc” que permite la programación de alta calidad en los microcontroladores AVR de Atmel, con diferencias muy explícitas.

Los pasos para la realización del programa son el diseño estructuras de la función que deseamos realizar, la programación en C que permite tal objetivo, la compilación y comprobación de este programa mediante diferentes

situaciones que nos permitan tener total seguridad de funcionamiento en los diferentes márgenes de operación.

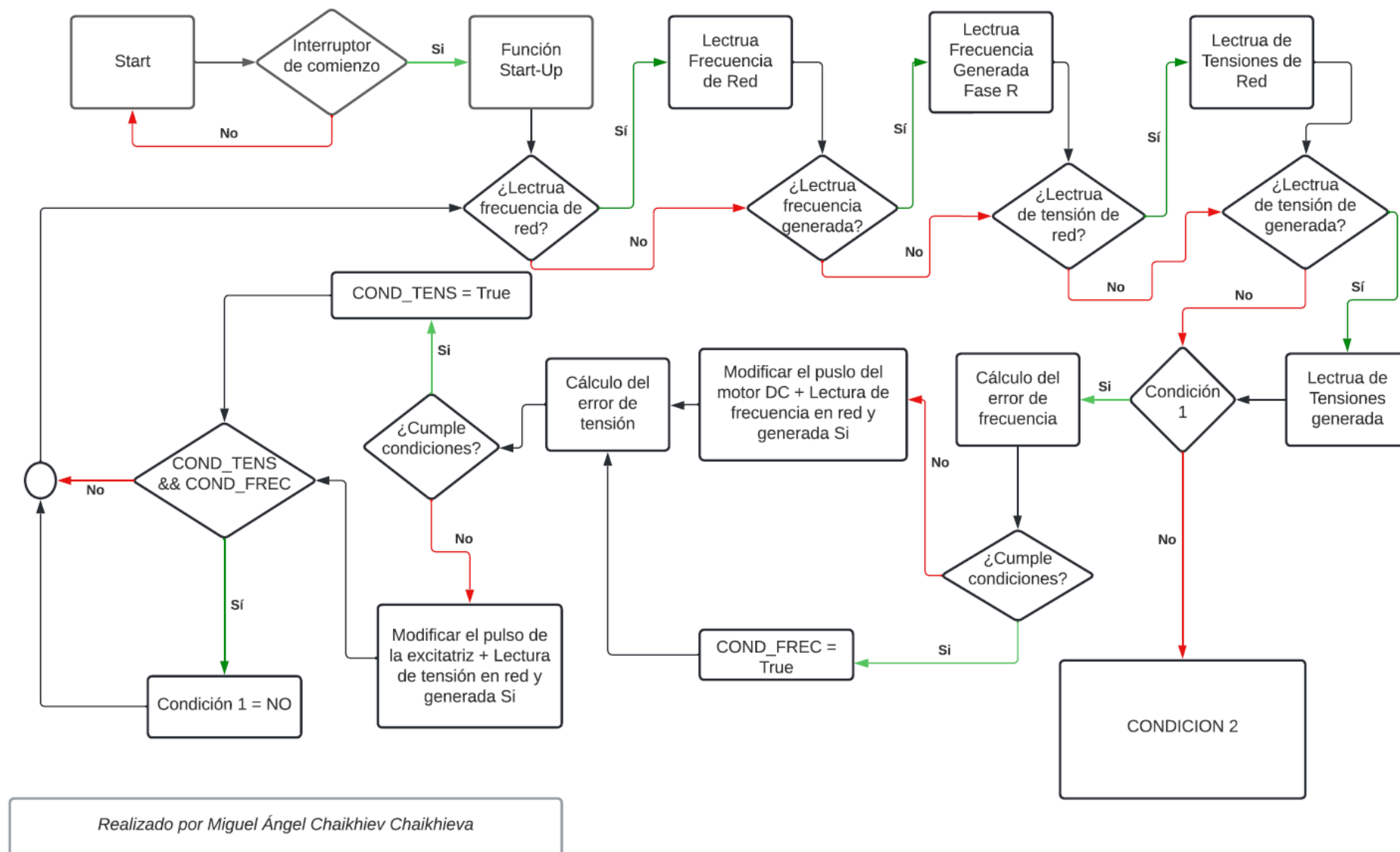
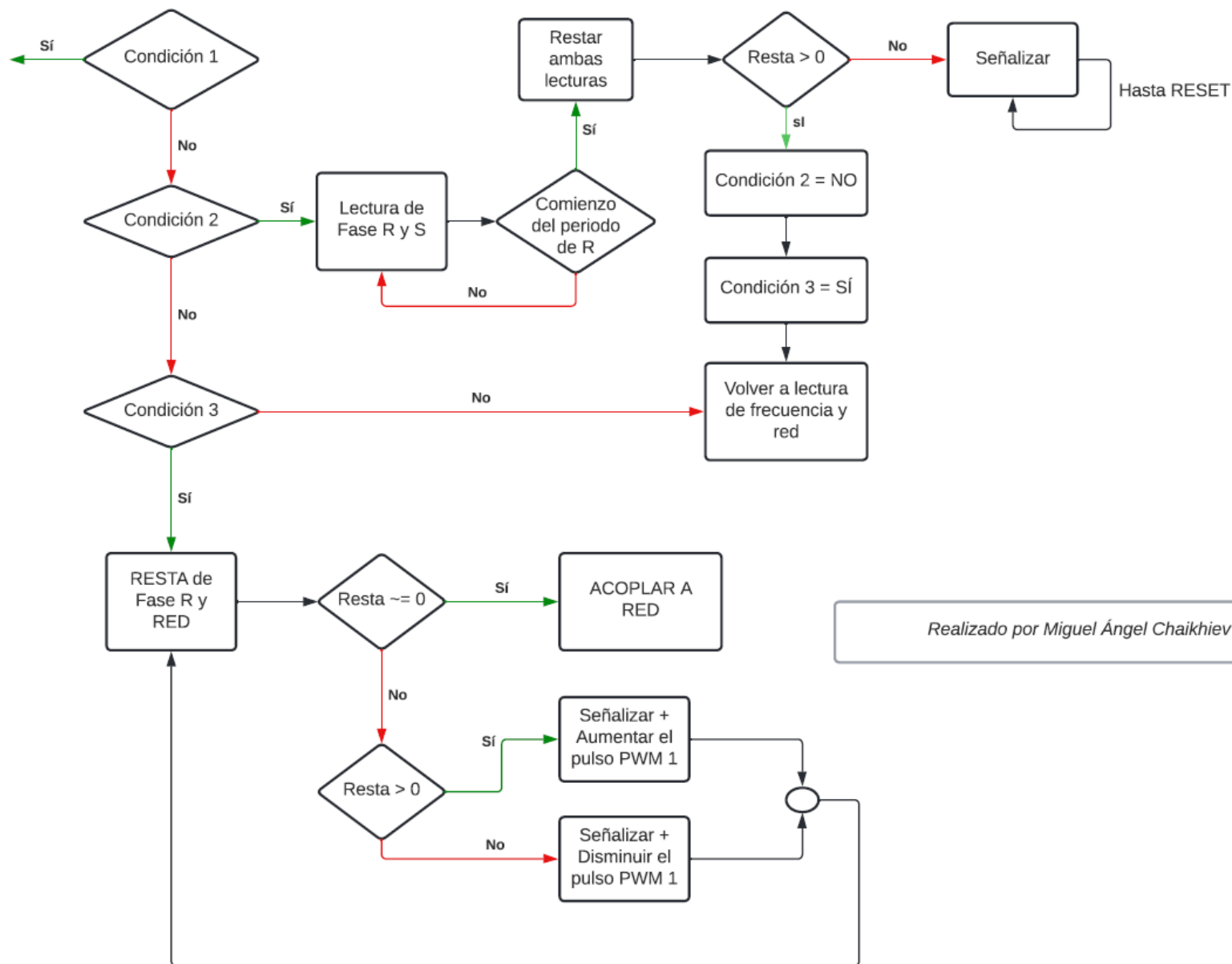


Ilustración 24: Flujograma: Control hasta la segunda condición.



Realizado por Miguel Ángel Chaikhiev Chaikhieva

4.3. Flujograma de sincronización.

En este apartado vamos a explicar el punto de vista para el desarrollo del programa en Arduino para desempeñar todas las funciones especificadas en el TFG mediante un flujo-grama para su implementación.

Tomando de referencia la ilustración 24, partimos desde una conexión de tensión activa, es decir, alimentando con una tensión de 100V el motor de cc para el sistema de baja potencia y su respectiva para el sistema de alta, junto a la excitatriz del alternador con 90V o su respectiva para el sistema de alta potencia (entorno a los 25V).

Utilizando el interruptor como comienzo del sistema automatizado realizamos las lecturas de las frecuencias y tensiones tanto de red como las generadas, una vez realizada la lectura de cada parámetro respectivo se deben volver a especificar su lectura, pues en caso de que no sea necesario partiremos con la comprobación de las siguientes condiciones.

La primera condición es la encargada de obtener una tensión eficaz similar entre la red y la generada junto a una frecuencia de giro idéntica, próxima a los 50Hz (50,1Hz ~ 49,49Hz). El error se calcula como la diferencia entre los parámetros de referencia y generados

Al incumplirse las condiciones de frecuencia, se procederá a la modificación del pulso de disparo para el respectivo dispositivo electrónico que controla el cortocircuito de la resistencia y la exigencia de la lectura de la frecuencia una vez modificado el pulso.

Si la condición del error de frecuencia se cumple, es decir, la frecuencia generada son 50Hz, procedemos al cálculo de error de la tensión de red de forma igual. La diferencia entre el parámetro de referencia y la generada debe de establecerse a valores reducidos dentro de un margen, por ejemplo, 10 valores numéricos como comentaremos en el siguiente apartado, si cumplimos la condición, es decir, la tensión generada se torna cerca de los 230V/240V generados (dependiendo del valor de red) procedemos al cumplimiento de la primera condición (CONDICIÓN 1 = NO) y volvemos

al comienzo del programa, el cual se saltará la necesidad de las lecturas de tensión y frecuencia de red y generada hasta la comprobación de la condición 2, la secuencia de fase mediante la resta entre los valores generados R y S, siendo la secuencia directa el valor positivo para el comienzo del pulso.

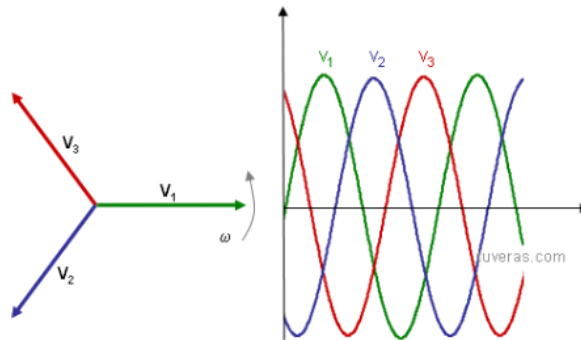


Ilustración 26: Secuencia directa de fases.

En la ilustración 26, V2 se corresponde a la fase S y V3 a la fase T desfasada 120° y 240° respectivamente, la toma de datos al comienzo del periodo permite ver que V2 tiene valor negativo y V1 nulo, por lo tanto, si restamos la fase R de la S debemos de obtener un valor positivo, el cual demuestra que estamos en secuencia directa.

Si no cumple la condición de secuencia de fase, es decir operamos en secuencias inversas, debemos de observar las conexiones de salida del generador, sus devanados. En secuencia directa corresponde con la conexión RST, la secuencia inversa (RTS) se corresponde a la conexión interna de los bornes del generador como se observa en la ilustración 7.

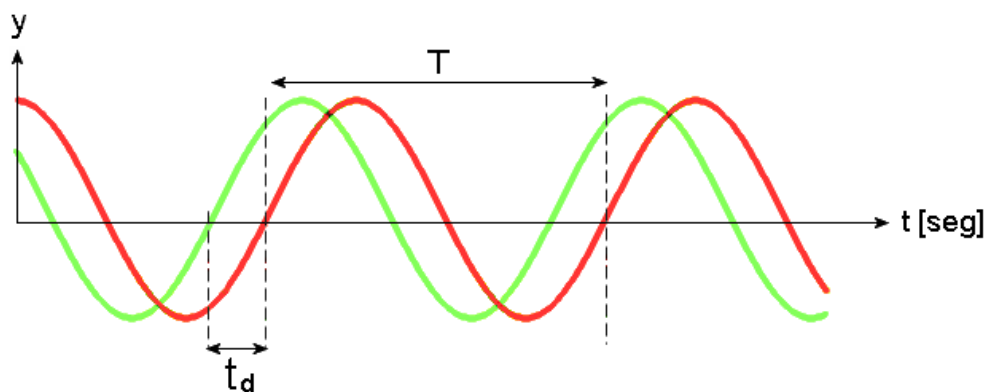


Ilustración 27: Desfase entre la señal de red y la generada.

Tras la lectura de la secuencia procedemos a la comprobación de la última etapa, desfase nulo, teóricamente tenemos misma tensión y frecuencia de red, adicionalmente la onda puede estar desplazada respecto a la tensión de red como se observa en la ilustración 27. Para reducir este valor td se debe de ir incrementando o reduciendo el valor hasta que el desfase sea nulo, como se observa en el flujograma de la condición 2 y 3, cuando cumplamos la última condición estamos sincronizados con la señal de red y procedemos al acoplamiento.

4.4. Bloque de lectura y señalización.

Debemos de realizar un control de la señal de red y las fases R-S de la f.e.m. generada por el alternador, para eso debemos de modificar los altos valores de tensión para que la placa stm32 pueda realizar la correspondiente lectura.

Tensión de referencia.

El microprocesador no puede trabajar con valores superiores a 3.3 V de entrada, ni 50mA, este último valor de la corriente máxima es aproximado, pero nos permite tener una idea del valor reducido de intensidad con el que operamos. Otra condición es que solo se trabaja con valores positivos, los negativos pueden destruir el microprocesador tras un periodo de trabajo prolongado.

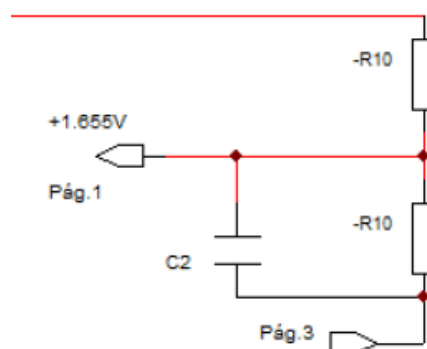


Ilustración 28: Referencia de tensión.

La opción seleccionada para adaptar la señal de salida y no trabajar con valores negativos es una elevación de referencia del neutro para trabajar solamente en el margen positivo de onda desplazado la parte proporcional. Como se ve en la ilustración 28, con tensión de alimentación Vcc de +3.3V.

Se añade un offset en DC mediante el uso de dos resistencias y un condensador que proporcionen un punto medio entre GND y +Vcc. La tensión del punto medio se obtiene como un divisor de tensión, donde ambas resistencias son iguales, mientras que el condensador puede operar con un margen entre los microfaradios.

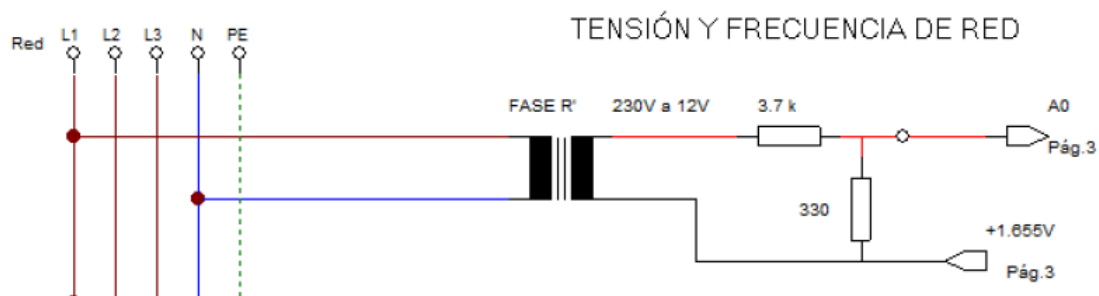


Ilustración 29: Divisor de tensión para lectura de valores.

El transformador seleccionado ha de soportar la tensión de red y la corriente nominal de trabajo, se ha seleccionado un transformador de 230V a 12V con una potencia aparente de 50VA. Esto da lugar a que la máxima tensión de trabajo sea la tensión de pico con una tolerancia del 10% y a 50HZ. Debemos de seleccionar un valor superior a la tensión generada por razón de no pasar el límite de operando durante la sincronización y acoplamiento, cogemos como referencia 245V.

$$V_{red, \text{máx}} = 245V + 10\% V_{tr} = 269.5 V$$

$$V_{red, \text{mín}} = 245V - 10\% V_{tr} = 220.5 V$$

Ambos valores nos permiten tener una idea de la tolerancia de red, entorno al 10%, el cuál debemos de tener en cuenta en la programación de la sincronización y acoplamiento.

Los valores más desfavorables se encuentran en los extremos máximo para el diseño de los componentes.

$$V_{tr,pico} = \sqrt{2} * V_{rms} = \sqrt{2} * 269.5 = 381.13 \text{ V}$$

Con valor máximo en el primario de pico de 381.13V, obtenemos el valor en el secundario.

$$V_{red,m\acute{a}x} = 12V + 10\% V_{tr} = 13.2 \text{ V}$$

$$V_{red,m\acute{i}n} = 12V - 10\% V_{tr} = 10.8 \text{ V}$$

Los valores más desfavorables se encuentran en los extremos máximo para el diseño de los componentes. La relación de transformación en este caso es de 230V a 12V.

$$m_v = \frac{230}{12} = 19.16667$$

Por lo tanto, la tensión en el secundario:

$$V_{tr,pico \text{ m}\acute{a}x} = \frac{381.13}{19.16667} \text{ V} = 19.885 \text{ V}$$

Obtenemos, por lo tanto, para el transformador **230V / 12V** una referencia máxima de **381.13 / 19.885V**, para la tensión de pico con la que tenemos que operar.

La tensión de pico de entrada es de 381.13V, en el secundario un valor de 19.885 V el cuál debemos de reducir a un margen de 1.65V para trabajarlo en la stm32, con este fin realizamos un divisor de tensión como en la Ilustración 29.

$$V_{salida} = V_{entrada} * \frac{R3}{R2 + R3} \quad (3)$$

Dónde.

V_{salida} es la tensión de 1.65V que queremos obtener.

$V_{entrada}$ es la tensión de pico de 19.885V como valor máximo.

$R3$ y $R2$ son resistencias que debemos de diseñar.

Por comodidad seleccionamos que las resistencias sean múltiplos, interesándonos un valor elevado para trabajar con corrientes reducidas. No podemos trabajar con resistencias demasiado altas pues al trabajar con una impedancia elevada se puede llegar a falsificar la lectura de los valores en las señales analógicas (el microprocesador trabaja con una impedancia elevada de base que al obtener valores externos similares produce errores de lectura), por eso, seleccionamos la resistencia $R2$ con valor de 3700 Ohm y aplicamos la ecuación 5.

$$1.65 = 19.885 * \frac{R3}{3700 + R3} \rightarrow 1.65 * (3700 + R3) = 19.885 * R3$$

$$6105 + 1.65 * R3 = 19.885 * R3 \rightarrow 6105 = (19.885 - 1.65) * R3$$

Obtenemos el valor de $R3$.

$$R3 = \frac{6105}{18.235} = 334.79 \text{ Ohm}$$

Seleccionamos un valor normalizado de la resistencia y obtenemos los márgenes de salida con lo que vamos a trabajar, siendo $R2$ y $R3$, respectivamente, $3.7K\Omega$ y 330Ω .

$$V_{salida, \text{máxima}} = 19.885 * \frac{330}{3.7k + 330} = 1.6283V$$

$$V_{salida, típica} = 12V * \sqrt{2} * \frac{330}{3.7k + 330} = 1.338965V$$

El valor de trabajo real que obtenemos en el primario para un margen de 3.3V.

$$1.65 = V_{sal} * \frac{330}{3.7k + 330} \rightarrow V_{sal} = \frac{20.15 V \text{ de pico}}{\sqrt{2}} = 14.25V \text{ secundario}$$

Por lo tanto, en el primario

$$V_{pr} = 14.5V * m_v = 14.5V * 19.16667 = 273.09V$$

Es la tensión registrada en el primario del transformador, la tensión de red, cuando medimos de 0 a 3.3V para valores máximos.

La corriente de operación máxima se obtiene en los valores extremos de operación para una tensión en el secundario correspondiente a los 386,221 valores de pico, es decir:

$$I_{máx} = \frac{\frac{386.221V}{m_v * \sqrt{2}}}{R1 + R2} = \frac{14.2487}{3700 + 330} = 0.0035356 A = 3.536 mA$$

Tres señales del mismo valor producen una corriente entorno a los 11 mA, lejos de los 50 mA máximos de operación que no queremos sobrepasar.

Filtros de tensión.

Al trabajar con el punto de referencia cc de 1,655V de tensión continua para obtener un valor off-set en la lectura de las señales analógicas de red y generada en las fases r y s apreciamos ciertos errores tras varios minutos de trabajo debido a la variación de la amplitud a frecuencia entorno a los 50Hz, este error produce una variación de tensión de referencia falsificando la lectura de las señales de red y generada, ya que si el valor de referencia varía se ven afectadas todas las señales referidas a esa tensión.

Para poder resolver este defecto que afecta totalmente a la programación, se ha decidido el desarrollo de un filtro paso bajo constituido por un condensador y una resistencia para atenuar toda señal cuya frecuencia supere los ~15Hz para poder trabajar con una señal cc constante, ilustración 30.

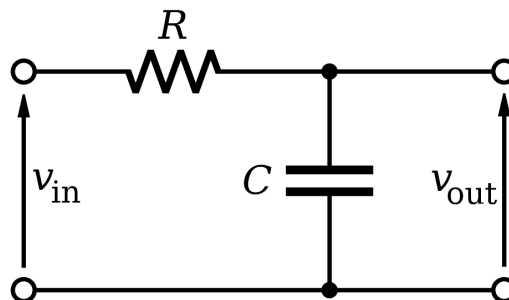


Ilustración 30: Filtros paso bajo compuesto por elementos pasivos.

El filtro electrónico paso bajo corresponde por permitir el paso de las frecuencias más bajas y atenuar las frecuencias más altas. Estos filtros pueden ser constituidos por elementos pasivos o activos, el alumno, a razón del material en pertenencia propia, desarrollo el filtro con elementos pasivos, condensador y resistencia. Los cálculos para tratar la onda son los siguientes.

$$f = \frac{1}{2\pi * C * R}$$

Donde

f es la frecuencia de trabajo, en Hz.

C es la capacitancia del condensador, en F.

R es la resistencia, en Ohm.

Con valor de condensador de 100uF, de resistencia 100Ohm, obtenemos la frecuencia de trabajo.

$$f = \frac{1}{2\pi * 100 * 10^{-6} * 100} = 15.91 \text{ Hz}$$

Es decir, toda señal con frecuencia superior a 15,91 Hz se atenúa permitiendo solo trabajar con frecuencias inferiores, en este caso, la nula. Observar la referencia en el “*Plano Eléctrico 3*”, tras la salida del punto de referencia cc.

La señal generada presenta el ruido referido a los acoplamientos inductivos del motor, pérdidas de histéresis, etc. Produciendo lo que teóricamente sería una señal senoidal ideal, un ruido periódico en la generación de ésta. Este ruido impide la correcta lectura de la frecuencia desde el microprocesador, impidiendo desarrollar una precisión de sincronización exigida por normativa, por lo tanto, para poder operar se ha diseñado un filtrado para operar con frecuencias de corto rango.

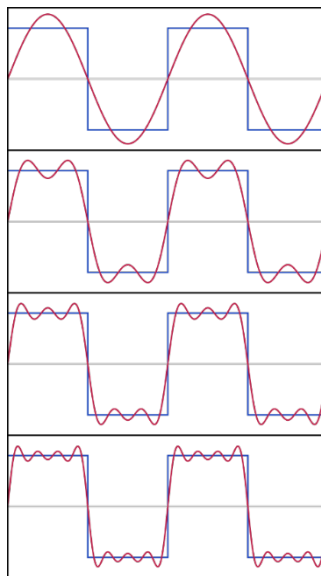


Ilustración 31: Descomposición de señal cuadrada en varias senoidales, mayor cantidad de señales mejor descripción de la onda.

Si tratamos una señal analógica periódica, independientemente de su idealidad teórica, es decir, con margen de ruido, se puede desarrollar por series de Fourier. Una descomposición de funciones periódicas sinusoidales para descomposición en funciones más simples para tratar la lectura de señales no lineal que presentan componentes armónicas de lectura.

Los armónicos presentes como múltiplos de la frecuencia fundamental a la que nos referenciamos (por ejemplo, la señal de red es de 50Hz, mientras que la señal generada puede ser 30Hz dependiendo de la velocidad de giro del rotor) por números enteros impares. Por lo tanto, si deseamos tratar solo con la señal principal y obviar la lectura de ruido externo debemos de diseñar un filtro paso bajo para señales inferiores a cierto valor de armónico.

Tomando de referencia la ilustración 30, en la que se describe el proceso del desarrollo de un filtro paso bajo para la atenuación de frecuencias superiores a cierto valor, operamos para tratar de eliminar armónicos superiores al 10º valor, como referencia a los 500Hz (décimo armónico, $10 \cdot 50\text{Hz}$, siendo 50Hz la frecuencia referencia).

Sustituyendo los valores de condensador 100nF, resistencia 2700 Ohm en la ecuación recién usada, obtenemos la frecuencia de corte.

$$f = \frac{1}{2\pi \cdot 100 \cdot 10^{-9} \cdot 2700} = 589.46 \text{ Hz}$$

Es decir, la señal periódica con frecuencia superior a los 590 Hz se atenúa, observamos el resultado en la ilustración 32. El diseño del filtro se dispone tanto en la lectura de la señal generada como en la lectura de la señal de red.

Cabe comentar que el filtrado de la señal armónica de la función seno ideal produce una reducción de lectura en la tensión eficaz generada por el microprocesador, este valor reducido se supone constante y antes de realizar el proceso de sincronización se debe de observar la diferencia de valores para una tensión, por ejemplo, 230V de red e incluir este valor en el desarrollo del programa de la stm32 en el ámbito de Arduino comentado en posteriores apartados.

Para su respectiva conexión observar el esquema del “*Plano Eléctrico 3*”, antes de la lectura correspondiente por el microprocesador stm32.

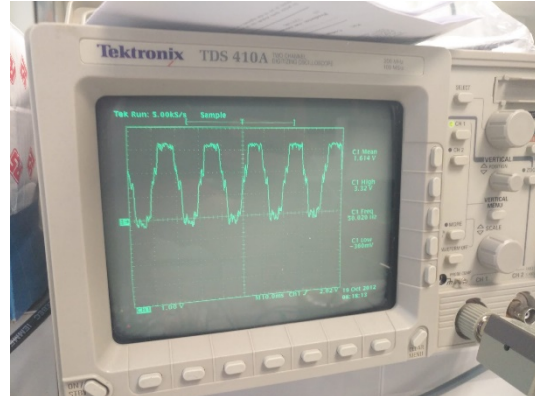
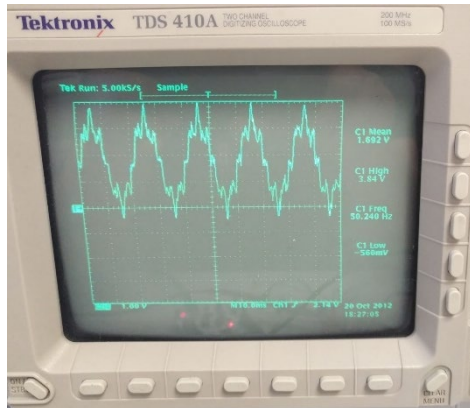


Ilustración 33: Señal de red y generada sin filtrar.

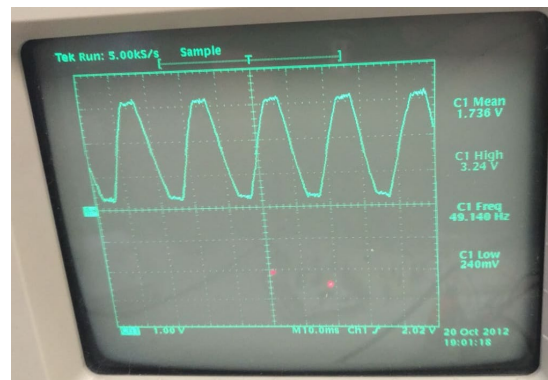
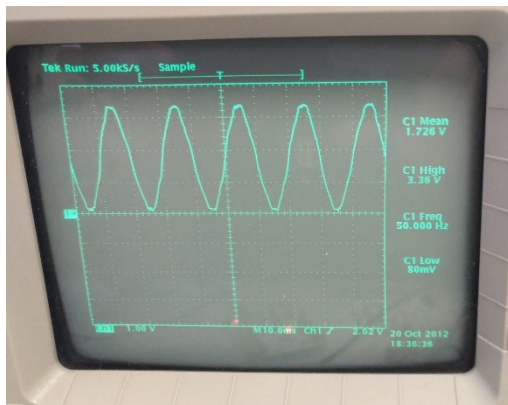


Ilustración 32: Señales de red y generada filtrada.

Interruptor.

El inicio del sistema se realizará mediante un **interruptor** que vamos a disponer en el esquema. En caso de que éste, durante el funcionamiento, se apagará, debemos de cargar la señal PWM progresivamente a valores nominales para reducir las corrientes de excitación correspondientes al alternador y al motor de corriente continua. En el apartado de programación se profundizará en este tema.

La conexión del interruptor se puede realizar de dos formas, resistencias de Pull Down y Pull Up, ilustración 34. Son mecanismos que permiten trabajar con alta tensión (5V) o baja (0V) antes de accionar el interruptor.

La resistencia de Pull-Up mantiene el valor HIGH cuando el pulsador está abierto, cuando se cierra permuta a LOW, la corriente se ve limitada por la resistencia.

La resistencia de Pull-Down mantiene el valor LOW cuando el pulsador está abierto, cuando se cierra permuta a HIGH, la corriente se ve limitada por la resistencia.

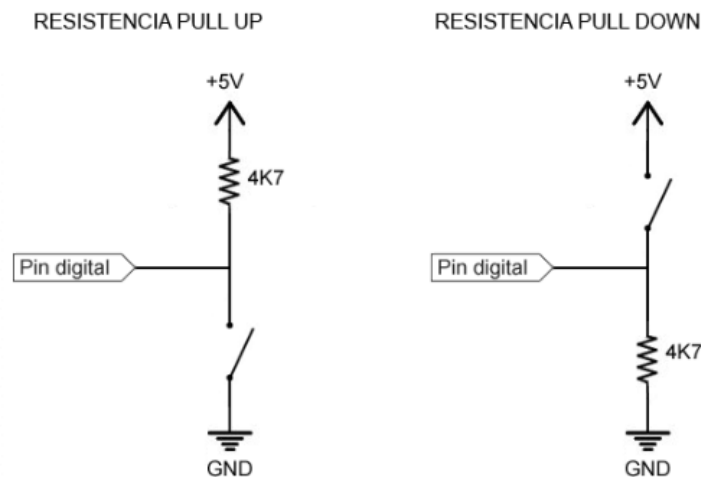


Ilustración 34: Conexión Pull Up y Pull Down, respectivamente.

La resistencia se usa para no producir un cortocircuito al conectar los 5V directamente a la lectura del smt32, pues provoca una corriente elevada que puede producir el sobrecalentamiento de los componentes y conductor, llevándolos a su destrucción. Además, la entrada quedaría en estado indeterminado sin posibilidad de definir cuál es el valor de la entrada al tratar con la construcción interna del autómata o el último estado conectado (estado de alta impedancia). El valor de la resistencia se condiciona por la “autoridad” que deseemos, una mayor “autoridad” corresponde a una mayor corriente y viceversa. Al aumentar la corriente, aumentan las pérdidas. Se ha seleccionado una conexión PULL-DOWN.

Led de señalización.

Para la representación visual de funcionamiento trataremos con varios leds de diferentes colores.

- Rojo, representando algún error o disparo de protección.
- Verde, representando el correcto acoplamiento o finalización de etapa.
- Amarillo, representando el proceso de sincronización en funcionamiento.

Estos Leds deben ir respectivamente protegidos mediante resistencias para que no se quemen por incapacidad disipadora calorífica en referencia a la corriente máxima de trabajo.

Color	Tensión umbral
Rojo	1.9v
Amarillo	1.7v a 2v
Verde	2.4v
Naranja	2.4v
Blanco	3.4v
Azul	3.4v

Ilustración 35: Valores típicos de tensión en LEDs.

Tomando de referencia una corriente máxima de circulación de 20mA y la tensión de alimentación los 5V que ofrece la stm32, obtenemos las resistencias respectivas.

$$V_t - V_{led} = I_{m\acute{a}x} * R$$

Dónde.

V_t es la tensión de alimentación, se mide en V.

V_{led} es la tensión umbral del led, se mide en V.

$I_{m\acute{a}x}$ es la corriente máxima de disipación, se mide en A.

R es la resistencia para instalar, se mide en Ohm.

Operamos.

$$R_{rojo} = \frac{V_t - V_{led}}{I_{máx}} = \frac{5V - 1.9V}{20mA} = 155 \text{ Ohm}$$

$$R_{amarillo} = \frac{V_t - V_{led}}{I_{máx}} = \frac{5V - 2V}{20mA} = 150 \text{ Ohm}$$

$$R_{verde} = \frac{V_t - V_{led}}{I_{máx}} = \frac{5V - 2.4V}{20mA} = 130 \text{ Ohm}$$

Podemos disponer de tres resistencias de valor de 150 Ohm para la protección de los leds.

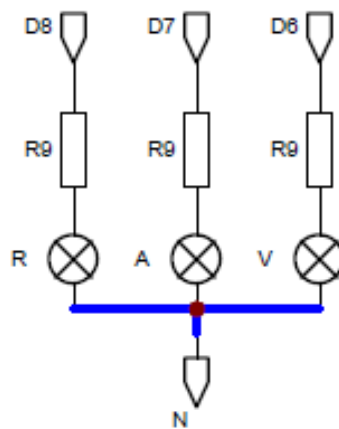


Ilustración 36: Conexión de leds en placa.

La respectiva conexión del interruptor y los leds se puede observar en el “Plano eléctrico 2”.

4.5. Bloque de control.

4.5.1. Control electrónico: IGBT.

El dispositivo electrónico con el que operamos se basa en un transistor bipolar de puerta aislada (IGBT, en nuestro caso el K30N60 y el K40N60), combina las ventajas de los BJT y los MOSFET. Tiene una impedancia de entrada elevada, como los MOSFET y bajas pérdidas en conmutación, como los BJT, por lo que puede trabajar a elevada frecuencia y con grandes intensidades.



Ilustración 37: Patillaje de las IGBT.

Los IGBT fueron desarrollados hace relativamente poco, pero su evolución ha sido veloz por las ventajas que presenta en la conducción y conmutación. Pueden soportar hasta tensiones de 1400V y corrientes de 300 A, el cual no es nuestro caso pues operamos con 100V y 9 A de máxima.

El control por tensión hace que el IGBT sea más rápido que el BJT, pero más lento que el MOSFET. La energía aplicada a la puerta que activa el dispositivo es reducida con valores de corriente entorno a los nano amperios, siendo posible controlarse con circuitos integrados.

El problema que presenta el IGBT es un coeficiente de temperatura negativo implicando que, a mayor temperatura, menor caída de tensión y por tanto un aumento de corriente, provocando un aumento de temperatura en la

unión. Este problema se debe de analizar en la conexión de varios IGBT en paralelo.

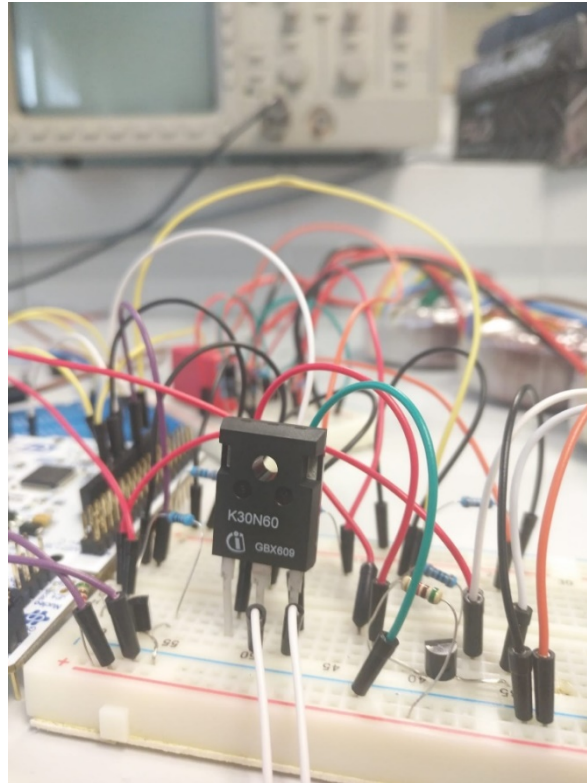


Ilustración 38: IGBT k30N60 con la que se opera.

La excitación de los IGBT se realiza por tensión, permitiendo su activación y desactivación de una forma sencilla. El estado de conducción se consigue cuando la tensión puerta-emisor sobrepasa la tensión umbral, lo que fuerza al IGBT a entrar en la región de trabajo óhmica. El desactivado de esta se realiza con una tensión menor a la “umbral” en la puerta-emisor. Las corrientes de puerta para los estados de encendido y apagados son reducidas, pero es necesario cargar la capacidad de entrada parásita para poner al IGBT en funcionamiento y descargarla para apagarlo.

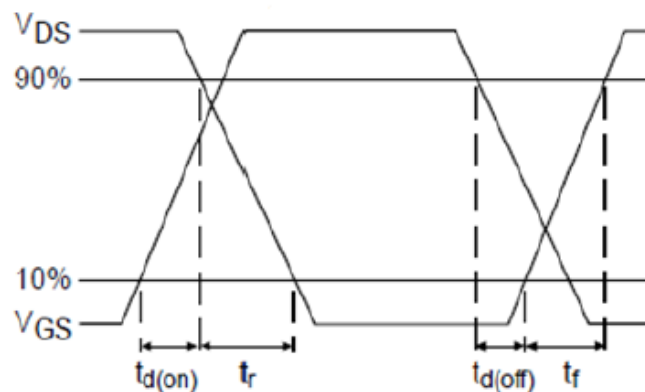


Ilustración 39: Tiempos de disparo de la IGBT.

Realizamos la comprobación del margen de trabajo para asegurar que no destruimos el transistor por trabajar en unas condiciones específicas. Como la mayoría de los transistores tendremos un tiempo de encendido (ton), de apagado (toff) y el de funcionamiento dentro de un ciclo de trabajo operacional, teniendo en cuenta la hoja de características del IGBT obtenemos los siguientes valores, tabla 1.

Tabla 1: Tiempos de permutación de la IGBT a 25°C, obtenida del Data-Sheet.

Símbolo	Denominación	Valor
<i>td (on)</i>	Turn On Delay Time	30ns
<i>tr</i>	Rise Time	35ns
<i>td(off)</i>	Turn Off Delay Time	270ns
<i>tr</i>	Fall Time	250ns
<i>ton</i>	Turn On	65ns
<i>toff</i>	Turn Off	520ns
<i>f</i>	Frequency	1KHz

La frecuencia de trabajo depende de la salida PWM del stm32, la cual se puede variar a valores más elevados, pero no es necesario para la implementación del sistema automático.

Teniendo en cuenta que la máxima potencia que puede disipar el IGBT según su Data-Sheet es de 200W, vamos a obtener el valor de la potencia generada en nuestras condiciones de trabajo.

La potencia disipada se calcula sumando las potencias respecto a cada intervalo de operación (on, off, cond).

$$P_t = \frac{W_{ton} + W_{toff} + W_{cond}}{T}$$

Donde

W_{ton} es la energía disipada durante el ton, en J.

W_{toff} es la energía disipada durante el toff, en J.

W_{cond} es la energía disipada durante la conducción, en J.

T es el periodo, en s.

P_t es la potencia disipada total, en W.

Procedemos al cálculo individual de cada energía, seguidamente juntamos todos los valores y analizamos los resultados obtenidos.

Durante ON

$$W_{ton} = 0.5 * V_{ds(sat)} * I_{dss} * ton$$

Donde

W_{ton} es la energía disipada durante el ton, en J.

$V_{ds(sat)}$ es la tensión de paso durante la saturación, en V.

$I_{c(sat)}$ es la corriente de paso durante la saturación, es decir, la corriente máxima de operación con la que debemos de trabajar se mide en A.

Ton es el tiempo de encendido en s.

Obtenemos los diferentes parámetros de la hoja de características y sustituimos.

$$W_{ton} = 0.5 * 2.7V * 10A * 65 * 10^{-9}s = 0.8775 * 10^{-6}J$$

Durante OFF

$$W_{off} = 0.5 * V_{dss} * I_{c(sat)} * t_{off}$$

Donde

W_{toff} es la energía disipada durante el t_{off} , en J.

V_{dss} es la tensión que deseamos bloquear, en V.

I_{dss} es la corriente que pasa en el bloqueo, se mide en A.

T_{off} es el tiempo de apagado en s.

Obtenemos los diferentes parámetros de la hoja de características y sustituimos.

$$W_{toff} = 0.5 * 100V * 200 * 10^{-6}A * 520 * 10^{-9}s = 52 * 10^{-10}J$$

Durante Conducción

$$W_{cond} = V_{ds(sat)} * I_{c(sat)} * t_{cond}$$

Donde

W_{cond} es la energía disipada durante el t_{cond} , en J.

$V_{ds(sat)}$ es la tensión de paso durante la saturación, en V.

I_{dss} es la corriente que pasa en el bloqueo, se mide en A.

T_{cond} es el tiempo de conducción en s, restando a T los valores de t_{on} y t_{off} .

Obtenemos los diferentes parámetros de la hoja de características y sustituimos.

$$W_{cond} = 2.7V * 200 * 10^{-6}A * (1 * 10^{-3} - (65 + 520) * 10^{-9}) = 5.3968 * 10^{-7}J$$

Juntando todos los valores, obtenemos.

$$P_t = \frac{0.8775 * 10^{-6}J + 52 * 10^{-10}J + 5.3968 * 10^{-7}J}{1 * 10^{-3} s} = 0.00142238 W$$

Considerando que la potencia máxima de disipación son 200W podemos asumir tranquilamente que estamos trabajando en una zona segura con margen de operación.

Durante la conducción en la zona óhmica, obtenemos la resistencia $r_{ds(on)}$ que tiene el IGBT obtenida de la gráfica “Typical output characteristic” para una tensión de 2V V_{ce} y corriente I_c de 15 A del Data-Sheet, esta resistencia disipa energía en forma de calor, para una corriente máxima de 9 A, obtenemos:

$$P_{dis} = I_d^2 * R_{ds(on)} = 9^2 * 0.13333 Ohm = 10.79973 W$$

Consideramos también la disipación de calor durante el trabajo en la zona de “saturación”, en la que el IGBT opera como resistencia variable hasta saturarse. Suponiendo la frecuencia de trabajo la misma y con una corriente máxima de 9 A, obtendremos el valor de la potencia disipada de la siguiente forma:

$$X_L = 2 * \pi * f * L$$

Dónde

f es la frecuencia del pulso PWM, en Hz.

L es el valor de la inductancia interna de los IGBT con valores del Data-Sheet de 13 nH.

Operando:

$$XL = 2 * \pi * 50Hz * 13 * 10^{-9}H = 4.084 * 10^{-6} \Omega$$

Disipando una potencia:

$$P_{dis} = Id^2 * R_{ds}(sat) = 9^2 * 4.084 * 10^{-6} = 0.3308 mW$$

En resumen, la potencia disipada en la conmutación es de 1.4 mW, trabajando en la zona Óhmica, 10.8 W y en la zona saturada 0.3308 mW. Con diferencia, la mayor potencia se disipa en el trabajo de la zona óhmica que no deseamos obtener, aun así, tenemos un margen de 189.2 W para poder disipar antes de su respectiva destrucción para una temperatura de 25°C y en condiciones críticas desfavorables.

4.5.2. Sistema de excitación del IGBT.

Primero el montaje real para la comprobación práctica del esquema teórico para la señal PWM, “ilustración 40: Circuito de control del motor de cc”, en aplicación real, con valor de resistencias externas de control de 2.2k para el motor de cc y 0.2 k para la excitatriz del alternador.

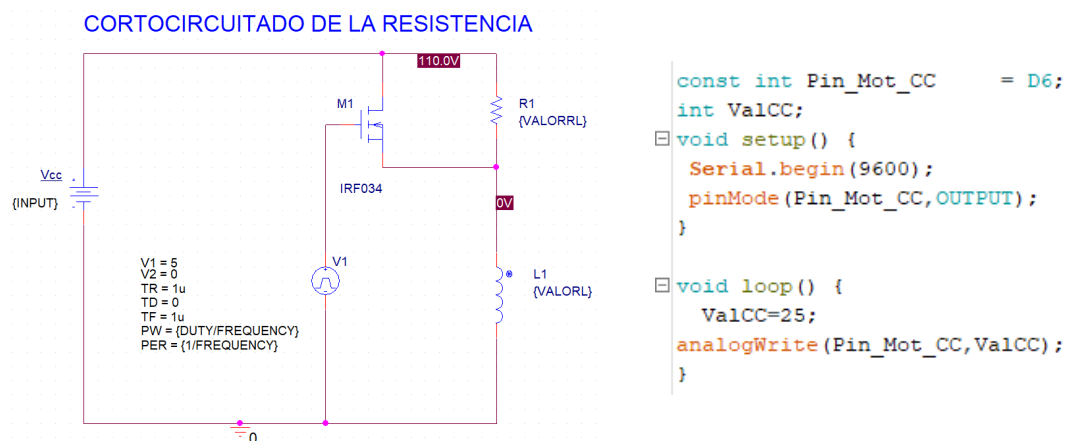


Ilustración 40: Cortocircuitado de una resistencia con el programa Arduino.

El control de salida se realiza mediante el PIN D3, para dos valores, 250 (ilustración 41) y 20 (ilustración 42), la salida debe de variar el valor de la corriente de paso.

La lectura de la variación PWM se realiza con un osciloscopio de precisión, permitiendo observar claramente como varía el ciclo de trabajo modificando los parámetros específicos.

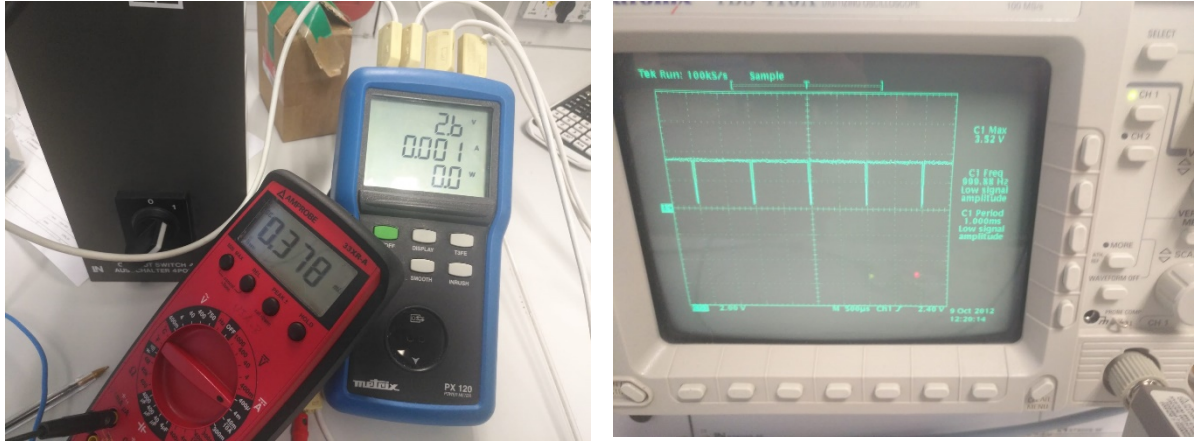


Ilustración 41. Señal PWM para 250 valores.

El programa implementado solo permite variar el valor de DUTY de la señal PWM hasta un margen de 0-255 valores, ilustración 40, auxiliándonos de dos multímetros para medir la tensión en V junto a la corriente en miliamperios, obtenemos con un valor de corriente inicial de 0.378 A para un DUTY de 20 valores, ilustración 42, al no cortocircuitar completamente la resistencia por lo tanto la corriente es reducida.

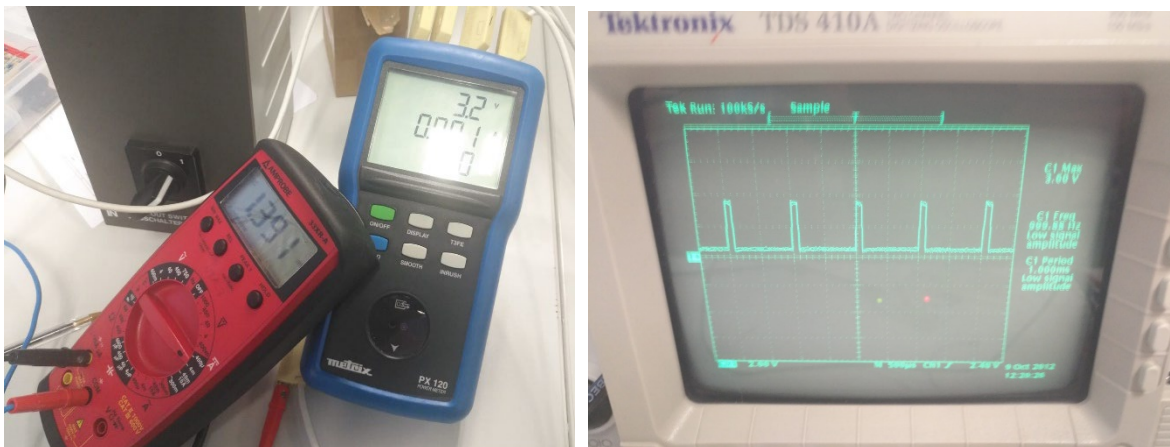


Ilustración 42: Señal PWM para 20 valores.

Tras variar el pulso a 250 valores numéricos y cargarlo obtenemos una corriente de valor 1.391 mA, ilustración 42, 1 mA superior a la corriente del caso anterior, por lo tanto, sabemos que el comportamiento es correcto.

Debemos de ir variando el valor del pulso para trabajar con la corriente de excitación, respectivamente la velocidad de giro del motor de cc.

Observando el Data-Sheet, podemos ver que la tensión que debemos producir en la puerta ($V_{ge(th)}$) es de mínima 4 V y máxima 7 V, es decir, con 7V se satura con total seguridad, pero nuestra placa stm32 propone una señal PWM con valor máximo de tensión de 3,3 V no modificables, por lo tanto, debemos de desarrollar un driver de excitación para elevar la tensión a valores que permitan operar con el IGBT. Observando el “Plano eléctrico 3”, hemos desarrollado para el control de ambos IGBT un elevador de tensión compuesto por dos resistencias, una pila externa de 9V y un BJT NPN con inversión de sentido, es decir, cuando el BJT no está saturado deja pasar 9V, en caso de saturación el BJT pone a tierra el Colector y dejando el paso de 0V, es decir, corte.

Por lo tanto, en los semiciclos positivos de PWM1 se corta el IGBT y en los negativos se activa.

CONEXIÓN A MOTOR CC.

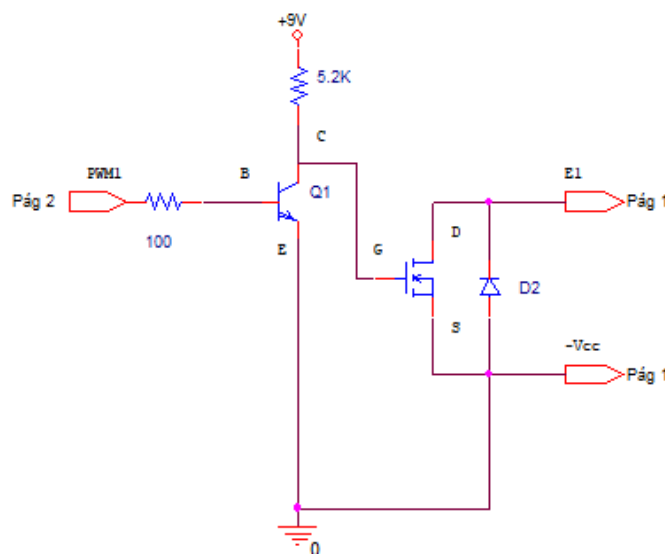


Ilustración 43: Driver elevador inversor para excitación del IGBT.

Vamos a presentar la ilustración 43, modificando el pulso PWM de la señal de la stm32 debemos de ir reduciendo o aumentando la tensión según tengamos que aproximarnos en mayor o menor proporción. Observando el “Data-Sheet” del IGBT de trabajo (k30N60 o k40N60), comprobamos que la tensión máxima para activar el transistor es rango de +/-30V con saturación de mínima 4V y máxima 7V, aumentando el valor facilitaremos el paso de etapa. Para esto debemos obtener un valor superior a los 3.3V obtenidos por la señal pulso.

Observando la ilustración 44 referida a un MOSFET y sirviendo de referencia pues tendremos las mismas zonas de operación, trabajaremos con tres zonas: lineal u óhmica (comportamiento como resistencia variable), saturación (cortocircuito entre C-E) y corte (Circuito abierto entre C-E), el paso de una a otra depende de la tensión de alimentación Gate-Emitter y la tensión Colector-Emitter, a su vez están definen la corriente de trabajo y los márgenes que no podemos rebasar para no quemar el IGBT. Debemos de trabajar en las zonas de corte y saturación.

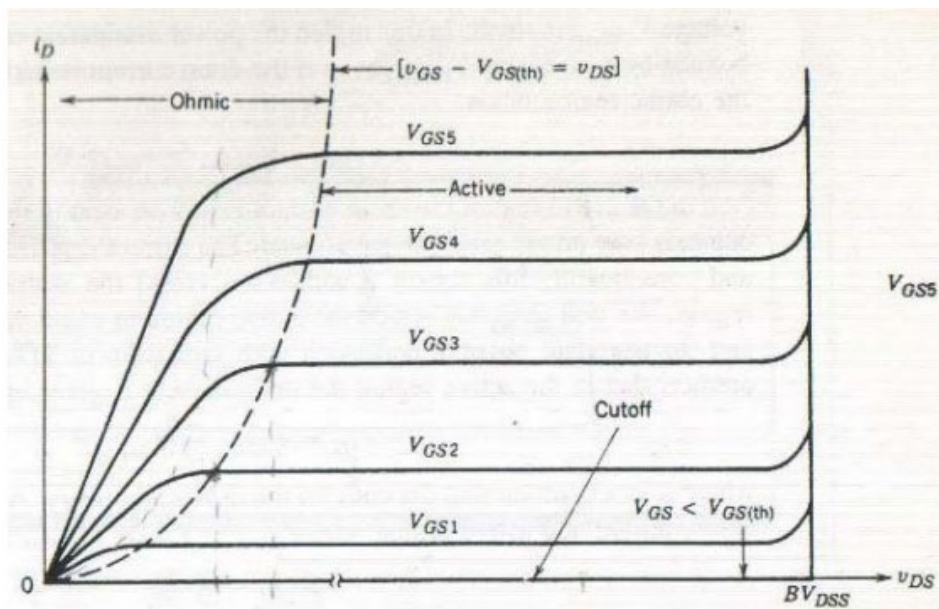


Ilustración 44: Zonas de saturación MOSFET.

Técnicamente no es recomendable tratar directamente con IGBT con procesadores actuales a razón de las diversas limitaciones de tensión y corriente en sus respectivas salidas. La tarjeta stm32 con la que se trabaja en este TFG varía su módulo con valor máximo de 3,3 V. Si deseamos realizar el trabajo directamente debemos de obtener transistores de nivel lógicos (más caros al estar diseñados específicamente para la conmutación con tensiones bajas), permitiendo la saturación veloz y correspondiente disipación.

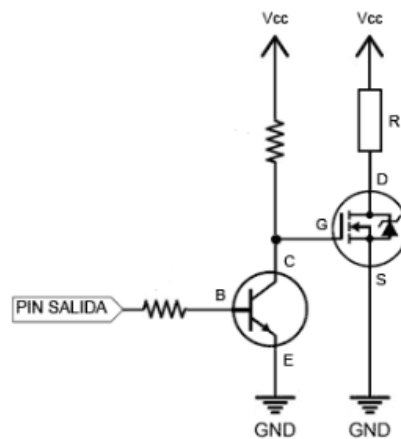


Ilustración 45: Driver de control de MOSFET.

Para evitar este problema debemos de limitarnos a cargas reducidas con corriente escasa, sustituir el IGBT o realizar una etapa previa de potencia (es la opción seleccionada por el estudiante), mediante el diseño de un “driver” para aumentar el valor de la señal de control, ilustración 45.

Las resistencias deben de reducir la corriente de trabajo, en el diseño se ha trabajado con un valor de 110 Ohm para la resistencia de señal en el pin de salida y 5200 Ohm para la alimentación de la tensión externa. Referida a la tensión externa y de las diversas opciones que se pueden escoger, el alumno ha decidido usar una pila de 9V por la facilidad de su obtención y su reducido precio económico. El montaje real se puede observar en la ilustración 46, en el que se ha trabajado con un BJT (PN2222A) seleccionado por la pertenencia del alumno con anterioridad. Cabe comentar que el BJT es canal N y con unas corrientes de operaciones máximas dispuestas en el Data-Sheet referente a su código en el apartado de Anexos 7.2 que no superamos, las potencias disipadas se estudian de la misma forma que el IGBT.

En la ilustración del esquema debemos de separar el apartado de alimentación del BJT con dos resistencias al apartado del Buck reductor, el cual contiene un diodo y resistencia en paralelo conectados a la patilla S del IGBT. El ejemplo se ha probado para tratar con unas tensiones elevadas, valores con los cuales el BJT resultaría perjudicado (es razón de esto la necesidad de esta doble alimentación).

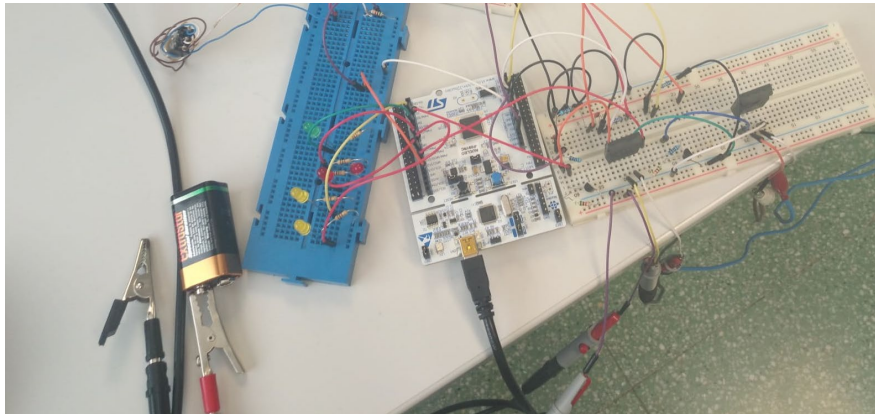


Ilustración 46; Implementación real del control de los IGBT.

La tensión de alimentación recitificada para ciclo completo sin tratar con un diodo para obtener señal continua de 50.5V, la salida del MOSFET es de 45.2 V, valor obtenido con una señal PWM de 10 valores numericos. La programación para comprobar la señal PWM del choper reductor es idéntica a la ilustración 40. La necesidad de trabajar con un paso intermedio invierte el valor numérico de señal PWM, es decir, para un DUTY de 1 necesitamos un valor numérico de 0, mientras que un DUTY de 0 se obtendría con un valor numérico de 255 en referente a la stm32 o sistema de programación que se esté utilizando.

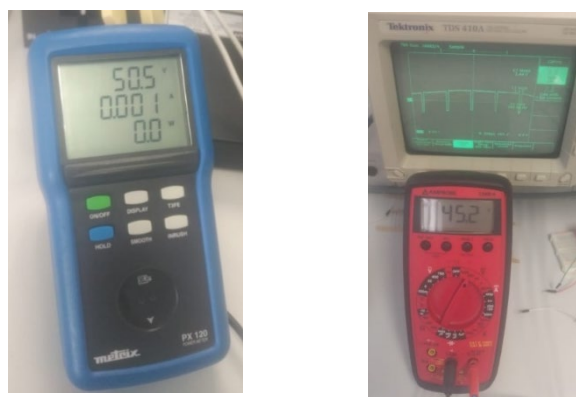


Ilustración 47: Valores de salida para una tensión de 50,5V y salida de 45,2V para un pulso de 245 valores.

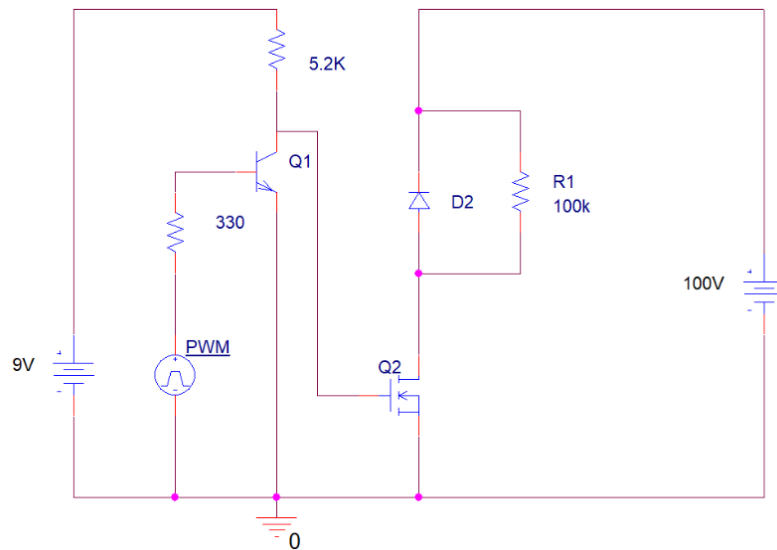


Ilustración 48: Esquema de comprobación de variación de tensión.

El conexionado específico de cada elemento se puede observar en la ilustración 48, diseñada por el alumno. La tensión de salida de R1 es la señal de alimentación de la excitatriz del alternador, tanto para el sistema de alta potencia como de baja potencia.

Tras el montaje de la ilustración 48 y cargar el programa en la stm32 como hemos comentado podemos apreciar la variación de la onda rectificada (no continua) para dos valores de pulsos opuestos en la ilustración 49.

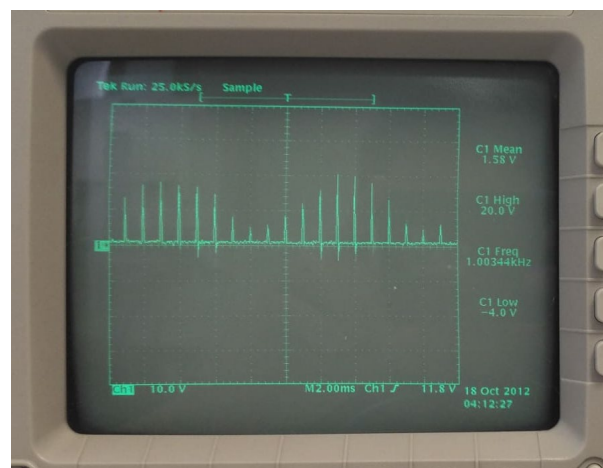
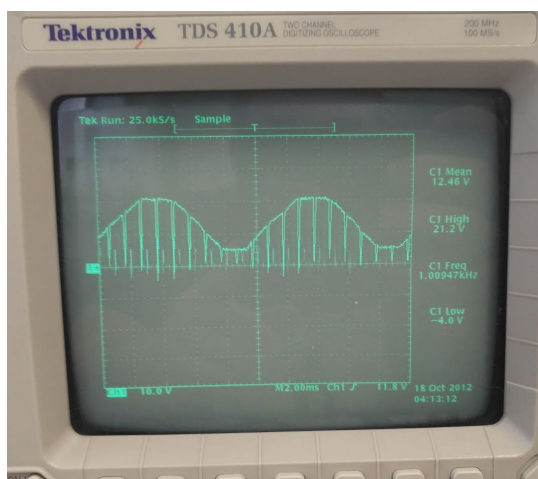


Ilustración 49: Valores de salida para un pulso de 250 y 10, respectivos.

Cabe comentar que la tensión de alimentación es una señal alterna rectificadora con un puente de diodos sin implicación de condensador para mantener la señal constante, ya que tras la alimentación del motor de corriente continua la señal se estabiliza en el valor cc por la componente inductiva del bobinado, produciendo el comportamiento esperado para una señal continua.

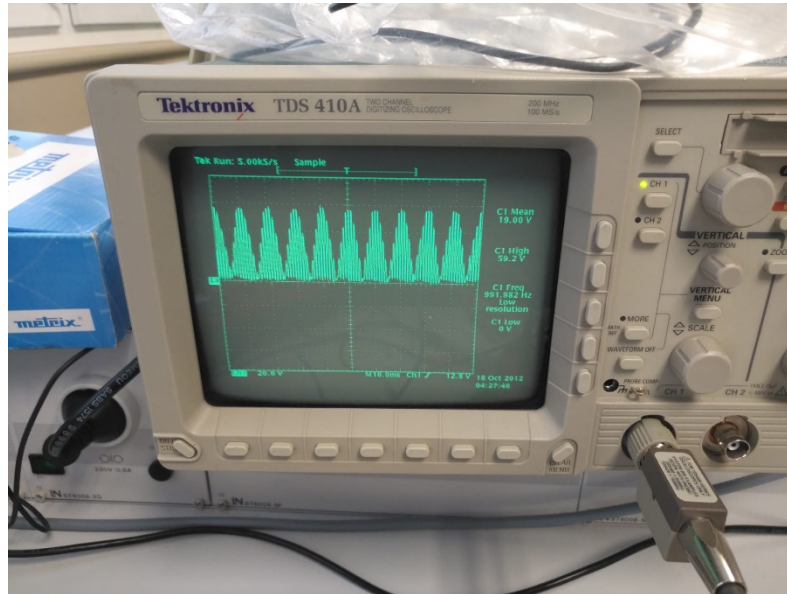


Ilustración 50: Señal modificada por ángulo de disparo de 240 valores.

4.6. Programación de la etapa de sincronización.

Una vez realizada la etapa de configuración y teniendo claro los flujogramas de referencia con los que nos vamos a guiar, procedemos a la respectiva programación.

Los diferentes programas que vamos a desarrollar se tienen que encargar del control de las señales PWM para el control del cortocircuito de la resistencia adicional al alternador encargado de variar la tensión de alimentación en la excitatriz, el cortocircuito del reóstato del motor de cc tipo compound para ir variando su valor, la comprobación de las cuatro condiciones establecidas para poder realizar el acoplamiento a red, la señalización del proceso de trabajo correspondiente, un sistema de seguridad para evitar estragos, un sistema de apagado y encendido del circuito. El conjunto de todos estos programas permite la adaptación del proyecto al circuito real.

Los estragos que pueden producirse en el proceso de sincronización y acoplamiento pueden ser la disposición del pulso de trabajo sobre los límites de funcionamiento, una lectura de una señal de entrada que no garantice el correcto funcionamiento del motor o su sobrealimentación sobrepasando los límites establecidos de funcionamiento (p.e. una señal de lectura de 260 V generados).

```
const int sensorPinRed   = A0;    //Pin que mide el valor de fase de red R.
const int sensorGen_R    = A1;    //Pin que mide el valor de fase generada R.
const int sensorGen_S    = A2;    //Pin que mide el valor de fase generada S.

const int Interr_PIN     = D13;    // Señal de control del interruptor.
const int Pin_Nara_1     = D2;    // Señal de control del LED NARANJA para visualización externa.
const int Pin_Mot_CC     = D3;    // Señal PWM para controlar el motor cc.
const int Pin_Alt        = D10;    // Señal PWM para controlar alternador.
const int Pin_Verde      = D4;    // Señal de control del LED VERDE para visualización externa.
const int Pin_Rojo       = D7;    // Señal de control del LED ROJO para visualización externa de ERROR.
const int Pin_Rojo_2     = D9;    // Señal de control del LED ROJO 2 para visualización externa de ERROR.
const int Pin_Nara_2     = D8;    // Señal de control del LED NARANJA 2 para visualización externa.
```

Ilustración 51: Definición de los Pins de trabajo.

Debemos de definir la función de los pins y la resolución de lectura/escritura en el bucle setup().

```
void setup() {
  Serial.begin(9600);
  analogReadResolution(12);
  analogWriteResolution(12);

  pinMode(Interr_PIN, INPUT);
  pinMode(Pin_Nara_1, OUTPUT);
  pinMode(Pin_Nara_2, OUTPUT);
  pinMode(Pin_Verde, OUTPUT);
  pinMode(Pin_Rojo, OUTPUT);
  pinMode(Pin_Rojo_2, OUTPUT);
  pinMode(Pin_Mot_CC, OUTPUT);
  pinMode(Pin_Alt, OUTPUT);

  Serial.print("GO");
  Serial.print('\n');
  delay(1000);
  digitalWrite(Pin_Verde, HIGH);
  var_cc=2047; var_alt=3276;
}
```

Ilustración 52. Función setup() de única ejecución.

Observamos como definimos los Pins de salida, mientras que los Pins de entrada analógicos están predefinidos solo obtenemos la lectura del interruptor externo. Disponemos la resolución de lectura y escritura de las señales PWM a 12 bits, lo que equivale a los 4096 valores numéricos. Serial.Begin(9600) para la activación del “monitor serie” y la resolución a la que trabaja.

Al comienzo del programa debemos de inicializar ciertas variables para facilitar el tratado de los diferentes Pins, se escribe encima de la función setup() y la predefinición de las funciones.

4.6.1. Tensión de referencia.

Debemos de obtener la tensión y frecuencia de red y generada para la fase R, para eso hemos realizado la adaptación a tensión de trabajo en el apartado posterior trabajando con los márgenes de 0V a +3.3V, correspondientes a tensión de -359.19V y 359.19V en el primario a -1.65V / +1.65V para los 3.3V máximos. Estos valores se establecen para la captación por la stm32 en un margen de 0 a 4095 valores. La tensión leída trabaja con un offset de 1.655 V de referencia real, lo que corresponde en la lectura de la stm32 como un valor próximo a los 2055 valores numéricos.

$$\text{Valor numérico} = \frac{1.65V}{3.3V} * 2054.206 = 2055V$$

Es decir, todo valor leído que tiene en el offset de 2055 valores numéricos obtendrá como máximo el valor de 4095 correspondiente a la lectura máxima de red. Siempre y cuando el valor de offset no se modifique, nos permitirá poder operar con este punto para el estudio de todas las operaciones como la lectura de la tensión, frecuencia, etc.

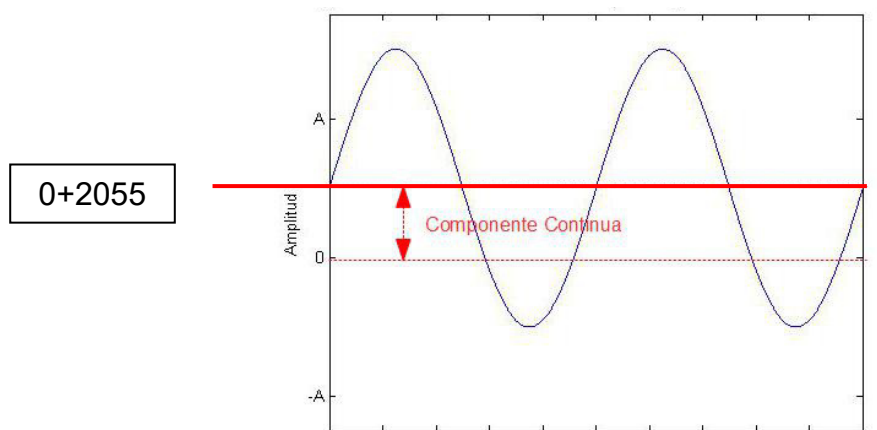


Ilustración 53: Punto de referencia continua para la lectura de la señal de red y generada.

El valor mínimo nunca bajará de 0 pulsaciones referenciadas, debemos de tenerlo en cuenta en los cálculos. Para cuando trabajemos con el valor máximo de red, restarle la parte proporcional del punto de referencia.

Primero introducimos los valores de las constantes, junto a los Pins con los que vamos a trabajar, en este caso son A0 para el valor de red, A1 para el valor de fase R generada y A2 para el valor de fase S generada como variables constantes e inamovibles.

Los valores "bool" corresponden a números binarios con valor "true" o "false" (0 o 1 lógico), los valores "float" son números enteros, los valores "int" son números reales (el prefijo "const" inmoviliza el valor independientemente de la programación), los valores "unsigned long" son números enteros sin espacio fijo de memoria para la lectura de los tiempos micros()). Todas las variables que deseemos utilizar deben de estar definidas.

Una vez implementados los parámetros de inicio, realizaremos una función en Arduino para la lectura, para facilitar la comprobación por el usuario externamente.

```
float LeerValor (float a, float val_min, float val_max){
  SensorValue = analogRead(a);
  return (((SensorValue-2050)*(val_max-val_min)/(4096-2050) + val_min));
}
```

Ilustración 54: Lectura de valores para uso referencial.

En la ilustración 54, observamos como la variable de salida es un “float” correspondiente a los parámetros de trabajo, las entradas de la función “LeerValor” son el pin como parámetro entero, junto a los márgenes equivalentes comentados para transformar el valor numérico en la tensión correspondiente de red.

Tras la captación de la señal analógica de red, en el margen de 0 a +1.65V y transformados a un valor entre 2050 y 4095, realizamos la operación matemática, la cual trabaja con los límites de los márgenes comentados, realizamos una interpolación.

Obtenemos.

$$V_{real} = \frac{(Valor\ del\ sensor - V_{num\ mín}) * (V_{máx\ de\ red} - V_{mín\ de\ red})}{(Valor\ num\ máx - Valor\ num\ mín)} + val_min$$

Sustituimos los valores y obtenemos la función de lectura, solo nos queda llamarla en la realización del bucle para saber los valores en el bloque de “loop ()”, ilustración 55, pudiendo obtener el valor proporcional de red, tensión de pico.

```
Red = LeerValor(sensorPinRed, valorminimo_pico, valormaximo_pico);
FaseR = LeerValor(sensorPinFaseR, valorminimo_pico, valormaximo_pico);
FaseS = LeerValor(sensorPinFaseS, valorminimo_pico, valormaximo_pico);
```

Ilustración 55: Llamada de la función "Leer Valor".

4.6.2. Inicialización del programa.

Tras el comienzo del programa mediante un interruptor externo, suposición inicial que da paso a la entrada digital D13, debemos de poder operar en condiciones básicas normales de funcionamiento, esto es, con cierta velocidad en el eje del rotor y cierto valor de tensión generado. Es por eso por lo que se debe de plantear los criterios de subida y bajada en la Tabla 2.

Tabla 2: Modificación de pulso DUTY 1 y 2, respectiva respuesta de velocidad y F.E.M.

Definición	Símbolo	SI AUMENTA ↑↑	SI DISMINUYE ↓↓
Pulso motor cc	$\delta 1$	'w' disminuye ↓↓	'w' aumenta ↑↑
Pulso alternador	$\delta 2$	E disminuye ↓↓	E aumenta ↑↑

Las condiciones estándar de trabajo iniciales son al 20% de cada margen inicial, en consecuencia, un pulso del 80% (100-20) para el DUTY2 del motor de cc para trabajar con cierto valor de velocidad y un 20 % de DUTY1 para la excitación del motor para obtener cierto valor de f.e.m. en los bornes de salida. Cabe comentar que la alimentación del motor cc produce de por si cierto valor de giro cercano a la velocidad establecida de 1500rpm, pero su respectiva disposición inicial nos permite comprobar de antemano su funcionamiento.

Vamos a realizar una señalización correspondiente para cada etapa de trabajo, por ejemplo, para esta primera etapa de inicio dispondremos el led naranja con un parpadeo constante de medio segundo intermitente, una vez establecida la inicialización se mantendrá fijo la iluminación para pasar a su correspondiente comienzo por pulsador.

Recordando, una señal PWM es una señal digital a 5V durante un intervalo de tiempo programable, es decir, obtenemos un valor promedio variable en un intervalo de tiempo fijo, como se observa en la ilustración 56.

Los automatismos dejan activa una salida digital durante un tiempo y la mantienen desactiva durante el resto del periodo produciendo esa señal analógica. La modulación por ancho de pulso es una de las técnicas más empleadas en el ámbito estudiantil y laboral por su fácil modificación y amplia aplicación.

Debemos de considerar los tiempos de conmutación para analizar la potencia disipada en los correspondientes transistores, si en algún caso se sobrepasa el límite de disipación energética se podría llegar a quemar con casi total seguridad.

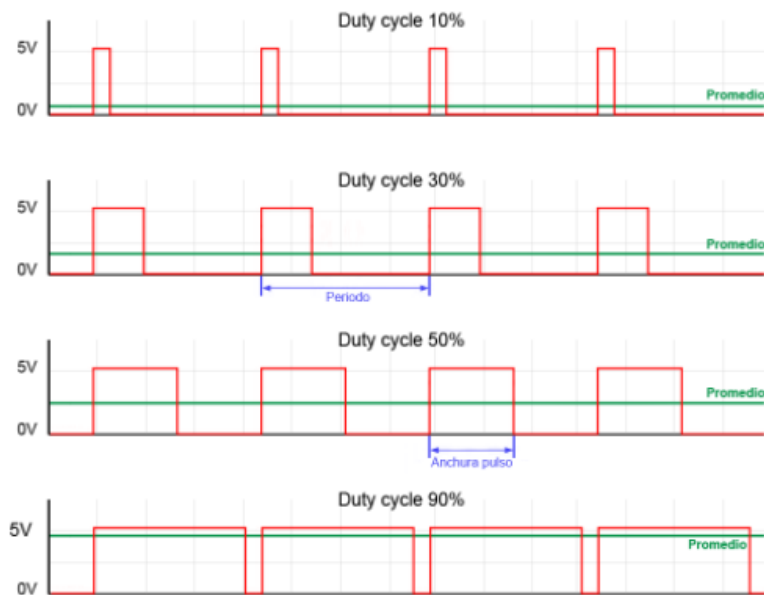


Ilustración 56: Señal PWM.

En Arduino el valor del PWM es una señal analógica variable para un intervalo de 0 a 4096 (según los bits de resolución), es decir, para un pulso DUTY nulo un valor numérico 0, contrariamente el valor extremo máximo, DUTY = 1, es el valor 4095. Mientras que en una placa stm32 tenemos un margen de 65535 valores ya que tenemos 16 bit para el Timer que nos proporciona una mayor resolución y, en consecuencia, una mayor precisión del valor en la salida como valor máximo, nosotros operaremos con margen de 12 bits.

Con definir los Pins de trabajo PWM, teniendo en cuenta el periodo de trabajo de estos que pueden ser 490Hz o 980Hz dependiendo de la salida para Arduino o las modificaciones que se pueden aplicar (las cuales no nos atañe), e ir incrementando o reduciendo los valores de pulsación para condiciones específicas tendremos logrado el objetivo. Con la placa stm32, la cual trabaja a una oscilación de 32.768 kHz en el cristal interno, obtenemos una salida normalizada entrono a los 1000 Hz de trabajo por predefinición en Arduino.

Podemos aumentar la frecuencia de los Timer reduciendo la resolución determinada, en nuestro trabajo nos concierne más la precisión y robustez que la velocidad de respuesta del sistema como hemos comentado.

Para poder utilizar las funciones PWM de Arduino con la stm32 debemos de implementar una librería correspondiente, al realizar la configuración tal y como hemos comentado debería de predefinirse automáticamente para su funcionamiento sin la necesidad de vinculación a los Timers correspondientes.

Según comentemos cada apartado en el que aparezca la variable específica, referenciaremos a esta ilustración como los sistemas de control del programa. Debemos de diferenciar las tres partes de lectura de frecuencia (la lectura de la señal S no es necesaria, pero la hemos dispuesto para su compresión), de tensión, los cuatro pasos para acoplamiento correcto, las condiciones de frecuencia y tensión (posteriormente las explicaremos).

Principalmente, la función del programa, tal y como hemos comentado, debe de establecer la señal de pulso al 80% de funcionamiento para el alternador, para la señal de control del motor cc al 20% respectivamente. Para asegurar un establecimiento correcto inicial, planteamos las condiciones iniciales en 100 pasos mediante la variación de la salida PWM, ilustración 58. El valor de pulso, un intervalo de 0 a 4095 respectivamente de 0 a 5V, tiene que variar al 80% para “ValAlt” en un intervalo de 100 pasos, en cada paso queremos incrementar un 0.8/100 porciento, por lo tanto, aplicamos los incrementos para la modificación correspondiente de “ValAlt” y multiplicada a la salida por el valor máximo de pulso. En el intervalo final ($i = 100$) obtendremos $0,8 \cdot 4095$.

```

void START_UP () {
    float v1=0, v2=0;
    for (int i=0; i<=100; i++) {

        digitalWrite(Pin_Nara_1, HIGH);
        v1 = ((0.008*i)*4095);
        Serial.print("v1: ");
        Serial.print(v1);
        Serial.print('\n');
        analogWrite(Pin_Alt, v1);

        v2 = (0.002*i*4095);
        Serial.print("v2: ");
        Serial.print(v2);
        Serial.print('\n');
        analogWrite(Pin_Mot_CC, v2);

        delay(20);
        digitalWrite(Pin_Nara_1, LOW);
        delay(20);
    }

    digitalWrite(Pin_Nara_1, HIGH);
    digitalWrite(Pin_Nara_2, HIGH);
    delay(1000);
    digitalWrite(Pin_Nara_1, LOW);
    digitalWrite(Pin_Nara_2, LOW);

    LEER_FRECUENCIA_RED = true;
    LEER_FRECUENCIA_GEN_R = true;
    LEER_FRECUENCIA_GEN_S = true;

    LEER_TENSION_RED = true;
    LEER_TENSION_GEN_R = true;
    LEER_TENSION_GEN_S = true;

    var_cc=v2; var_alt=v1;
}

```

Ilustración 57: Función Startup en Arduino.

Tras estos valores, esperamos 1 segundos para proceder a la adaptación de la tensión generada a la de red, señalizamos este proceso con los leds naranja 1 y 2.

Cabe comentar que la entrada para “pwmWrite”, primer parámetro el pin de respuesta y segundo el valor de pulso, deben de ser valores enteros, para eso debemos de realizar la conversión de un valor “float” (real) a uno “int” (entero), obteniendo el valor entero exclusivamente para poder operar en la función respectiva. Si disponemos directamente el valor real con decimales, la función machaca las decimas y opera con la parte entera exclusivamente.

```
void loop() {  
  
    pulsador =digitalRead(Interr_PIN);  
  
    if ( pulsador == 1 && START==true){  
        digitalWrite(Pin_Verde,LOW);  
        START_UP ();  
        Serial.print('\n');  
        Serial.print("START UP FIN");  
        Serial.print('\n');  
        CONDICION_1=true;START = false;  
    }  
}
```

Ilustración 58: Segundo bloque de reiteración en Arduino.

Esta función se invoca al principio del programa con la conmutación del interruptor, un mensaje en el “script” de Arduino para poder controlar el avance del programa y el establecimiento de la condición 1 a realizar junto a la salida del bucle inicial (START = false).

4.6.3. Primera y segunda condición: Igualación en frecuencias y tensión.

Una vez completada la etapa inicial, debemos de obtener una frecuencia generada próxima a la de red, es decir, los 50Hz específicos, para esta llamada debemos de ir modificando el valor PWM que controla el cortocircuito del reóstato para obtener una reducción en la corriente de excitación y, en consecuencia, un aumento de la velocidad de giro en el rotor.

Por lo tanto, debemos de medir la frecuencia de la f.e.m e ir aproximándola a la de red, eso lo obtenemos midiendo el tiempo de paso por un mismo valor para semiciclos positivos, este valor medido debe de ser próximo a los 0.02s que son en proporción los 50Hz de salida. A su vez

debemos de ir alcanzando el valor de la tensión de referencia, 230V, las acciones se deben de ir implementando prácticamente a la misma vez a razón de que el aumento de la carga generada produce una fuerza contraelectromotriz que reduce la velocidad de giro. Por lo tanto, debemos de realizar inicialmente la lectura de la tensión y frecuencia, seguidamente la modificación de los valores respectivos.

Siempre que se requiera leer la frecuencia debemos de activar la variable de control LEER_FRECUENCIA_RED o la variable respectiva a la generada. Tomamos 30 valores y de frecuencia y realizamos la media, podemos obtener más valores, pero no es necesario pues el resultado es preciso.

```
while (LEER_FRECUENCIA_GEN_R == true) {
    frec_max_R = 0; Dif_tiemp = 0; j = 0; frec_R = 0;
    while (j < 30) {
        tiempo3 = 0; tiempo4 = 0;
        sensorValue_R = analogRead (sensorGen_R);

        do {
            tiempo3 = micros();
            sensorValue_R = analogRead (sensorGen_R);
        } while (sensorValue_R >= 2050);

        do {
            sensorValue_R = analogRead (sensorGen_R);
        } while (sensorValue_R > 1900);

        do {
            sensorValue_R = analogRead (sensorGen_R);
        } while (sensorValue_R <= 2050);

        do {
            sensorValue_R = analogRead (sensorGen_R);
        } while (sensorValue_R < 2150);

        do {
            sensorValue_R = analogRead (sensorGen_R);
            tiempo4 = micros();
        } while (sensorValue_R >= 2050);

        Dif_tiemp = tiempo4 - tiempo3;
        frec_R = (1e6 / Dif_tiemp);

        //Serial.print('\n');
        //Serial.print("Dif_tiemp g ");
        //Serial.print(Dif_tiemp);

        if (frec_R < 55) {
            frec_max_R = frec_max_R + frec_R;
            j++;
        }
    }
}
```

El objetivo es la lectura entre un mismo valor asegurando que ha realizado un ciclo, es decir, dividido en cuatro etapas, una bajada, subida durante el semiciclo negativo y un subida y baja hasta el punto de partida del semiciclo positivo, seguidamente se toma el valor tardado en la variable "Dif_tiemp" y obtenemos la frecuencia de operación. Debemos de pasar la variable de microsegundo a segundos (se lee el valor en microsegundos "micros()"), las frecuencias pertenecientes a intervalos lógicos de 55/10 Hz). Seguidamente se hace la media y se finaliza la lectura de frecuencia negando la acción en "false". Este último bool se reactiva según necesitemos.

```
LEER_FRECUENCIA_GEN_R = false;
Serial.print('\n');
Serial.print("FRECUENCIA GENERADA FASE R: ");
Serial.print(frec_max_R / 30);
```

La función de delay(1), es la espera de un milisegundo, comentamos esta diferencia de subida y bajada para que el programa pueda diferenciar la toma de datos 1 y 2, podemos reducirlo una décima de microsegundo para realizar la lectura, no es necesaria teóricamente pero prácticamente realiza un “debug” para un correcto funcionamiento.

La variable de frecuencia se obtiene de la siguiente forma.

$$T = \frac{1}{f} \rightarrow \frac{1}{f} = \frac{Dif_tiempo_Tot}{1} * 10^{-6}$$

La toma de datos es referente a los 30 ciclos, los cuales se guardan en una variable para obtener el valor medio al dividir por la cantidad de iteraciones “n”, seguidamente pasamos el valor a segundo desde los microsegundos. Obtenemos la frecuencia al despejar.

$$f = \frac{1}{Dif_tiempo_Tot * n} * 10^6$$

Seguidamente procedemos a la lectura de la tensión, partiendo del valor inicial de control booleano como true, al igual que la lectura de la frecuencia, tras su medida solo se podrá volver a leer la tensión y frecuencia tras la variación del pulso de la señal de control. En este caso, hemos desarrollado una función para la lectura de la tensión aplicando el concepto de tensión eficaz en un periodo específico. Para eso necesitamos obtener la frecuencia y realizar la ecuación del valor cuadrático medio.

$$V_{ef} = \sqrt{\frac{\sum V^2}{T}} \quad (4)$$

Donde

- Vef es la tensión eficaz leída, en V.

- V es la tensión instantánea, en V.
- T es el periodo de la onda.

La función desarrollada se denomina Tensión_Actual, con valor de salida “float” y parámetros de entrada el pin de lectura y la frecuencia de referencia. El objetivo es ir guardando 100 valores en una matriz separados en su proporción respectiva a la frecuencia.

$$Tespera = \frac{10^6 \mu s}{100 * frecuencia} \quad (5)$$

El tiempo de espera es en microsegundos, por lo tanto, debemos de pasar de segundo a microsegundos, se toma un dato cada ese intervalo de tiempo. Seguidamente se realiza el cuadrado de cada valor guardado en la matriz restando el punto de referencia (recordamos que es el valor numérico 2050) y las sumamos, seguidamente realizar la raíz cuadrada y dividir en 100 intervalos acumulados. Obtenida la tensión eficaz para cada periodo específico podemos proceder a la realización de la condición 1: tensión y frecuencia.

```
float Tension_Actual (int a,float frecuencia){
    int vtension[100];
    unsigned long V_acum=0;
    float rms=0,Vsalida=0;
    do {
        while(analogRead (a) >=2050);
        for (byte cont=0;cont<100;cont++){
            vtension[cont]=analogRead(a);
            delayMicroseconds(1e4/frecuencia);
        }

        for(byte j=0;j<100;j++){
            V_acum=V_acum+pow((vtension[j]-2050),2);
        }
        rms =sqrt((V_acum/100));
        V_salida = rms;
        V_acum=0;

        return(V_salida);
    }
}
```

Ilustración 60: Función de lectura de tensión en Arduino.

Una vez obtenida la tensión y frecuencia de red y generada procedemos a la variación del pulso para obtener cada respectiva variable en concordancia a la magnitud de este.

Debemos de ir incrementando o decreciendo el valor de la resistencia para obtener una frecuencia de giro entorno a las 1500rpm mediante una variación del pulso PWM que cortocircuita el valor de la resistencia. Dependiendo del margen en el que operemos realizaremos la aproximación con una mayor velocidad o menor a razón de aproximar valores muy cercanos de la frecuencia origen. El programa trabaja con una tolerancia del 0.2%, es decir, valores referentes al 0.2% de $50\text{Hz}=0.1\text{ Hz}$ de diferencia entre la señal de red y generada. Al igual debemos de ir variando el valor de la tensión generada cortocircuitando otra resistencia, pues según varíe la carga generada disminuye la velocidad de giro del rotor, comprobación cíclica hasta que cumplan las condiciones. En caso de que el pulso de control supere los 4095 o mantenga valores inferiores a 0 debemos de señalar externamente. Al igual que si deseamos parar la sincronización debemos de desactivar el interruptor.

```

if(CONDICON_1==true) {
//*****
float FREC_REF =frec_max_R/30;float FREC_REF_RED=frec_max/30;
Err_Frec = (FREC_REF_RED-FREC_REF-1.5);

    Serial.print('\n');
    Serial.print("Err_Frec:    ");
    Serial.print(Err_Frec);
if (Err_Frec<=0.1&& Err_Frec>=-0.1){
    CONDICON_FRECUENCIA=true;
    Var_Tension = true;
    digitalWrite(Pin_Verde,HIGH);
    delay(50);
    digitalWrite(Pin_Verde,LOW);
    delay(50);
    digitalWrite(Pin_Verde,HIGH);
    delay(50);
    digitalWrite(Pin_Verde,LOW);
    delay(50);
}
}

```

Ilustración 61: Comienzo de la primera condición en Arduino.

Observamos que el comienzo de la condición se establece cuando `CONDICION_1 == true`, lo cual se cumple cuando realizamos la función `STRAT_UP()`.

Si cumple la condición señalizamos con el parpadeo del led verde y permitimos la modificación de la tensión, al igual que no es necesaria la lectura de la frecuencia, en caso de que no nos mantengamos en esos límites de error procedemos a la variación del parámetro `Contr_puls`, el cual incrementa o disminuye el pulso.

```
if (Err_Frec>0.1|| Err_Frec<-0.1){
    Var_Tension = false; CONDICION_FRECUENCIA = false;

    if(Err_Frec<-5){
        Contr_puls= -600;
    }

    if(Err_Frec>=5 && Err_Frec<-3){
        Contr_puls =-400;
    }

    if(Err_Frec>=3 && Err_Frec < -1){
        Contr_puls=-200;
    }
    if(Err_Frec>=-1 && Err_Frec < -0.35){
        Contr_puls=-120;
    }
    if(Err_Frec>=-0.35 && Err_Frec < -0.25){
        Contr_puls=-70;
    }
    if(Err_Frec>=-0.25 && Err_Frec < -0.15){
        Contr_puls=-20;
    }
    if(Err_Frec>=-0.15 && Err_Frec < -0.05){
        Contr_puls=-10;
    }
}
```

Ilustración 62: Modificación del pulso correspondiente al margen de error en el que operemos.

Para valores negativos de error, es decir, generamos una frecuencia superior a la de red debemos de reducir el valor de la resistencia para reducir el valor de la corriente de excitación, lejos de los 5 Hz de error correspondientes a 150 rpm de error para el valor extremos variamos el pulso 600 valores, correspondientes a un 14.65% de la señal de pulso, según nos acerquemos a la condición de 0.1 Hz de error para el valor más sensible, debemos de modificar en 10 unidades el pulso, correspondiente a un 0.2442% del DUTY. Para error positivo realizamos las mismas modificaciones de pulso, pero con valor positivo.

Seguidamente el valor de pulso se actualiza en la variable var_cc, se comprueba que no operemos fuera de los límites de trabajo, la condición del pulsador y realizamos el envío de la señal para modificar el pulso correspondiente, señalizamos con el led naranja 1 y permitimos la lectura de la frecuencia de red y la del generador R.

La modificación del DUTY2 para modificar la f.e.m. generada sigue el mismo principio, se calcula el error mediante la resta del valor de tensión de red y la media entre los parámetros generados de la fase R y S, si cumple las condiciones de 10 valores numéricos, correspondientes a 1V de diferencia, procedemos con la maniobra de condición cumplida mediante un parpadeo del led verde (al igual que en la maniobra de frecuencia) y el bool `CONDICION_TENSION = true`.

```
if (Var_Tension = true){
    Err_V = Tens_max-(Tens_max_S +Tens_max_R)*0.5;

    Serial.print('\n');
    Serial.print("Tens_max_S y Tens_max_R :      ");
    Serial.print(Tens_max_S);
    Serial.print(" y ");
    Serial.print(Tens_max_R);
    Serial.print('\n');
    Serial.print("Err_V:      ");
    Serial.print(Err_V);

    if (Err_V<=10 && Err_V>=-10){
        CONDICION_TENSION=true;

        digitalWrite(Pin_Verde,HIGH);
        delay(50);
        digitalWrite(Pin_Verde,LOW);
        delay(50);
        digitalWrite(Pin_Verde,HIGH);
        delay(50);
        digitalWrite(Pin_Verde,LOW);
        delay(50);
    }
}
```

Ilustración 63: Proceso de obtención de la F.E.M. en Arduino.

En caso de no cumplir las condiciones de 1V realizamos la modificación del DUTY2, para valores de error de -200 valores, es decir, la tensión generada es mayor que la de red, debemos de disminuir el valor que recae en la excitatriz del alternador, esto es, aumentando la resistencia en serie conectando externamente pues, al ser un divisor de tensión, al aumentar la resistencia aumenta la tensión que recae sobre ella disminuyendo la tensión de

excitación. Por lo tanto, el pulso debe de aumentar pues tenemos un “driver” inversor como recordamos que hace que la resistencia se cortocircuite menos veces.

Análogamente para valores de error positivos, es decir, la f.e.m. generada es menor, el pulso debe de decrecer con los mismos márgenes de pulso.

Seguidamente el valor de pulso se actualiza en la variable var_alt, se comprueba que no operemos fuera de los límites de trabajo, la condición del pulsador y realizamos el envío de la señal para modificar el pulso correspondiente, señalizamos con el led naranja 2 y permitimos la lectura de la tensión de red, la del generador R y la del generador S.

```

var_alt = var_alt + Contr_puls_V;

if(var_alt >=4096|| var_alt<=0){
    digitalWrite(Pin_Rojo_2,HIGH);
}

if(var_alt <4096 && var_alt>0){
    digitalWrite(Pin_Rojo_2,LOW);
}

if(pulsador==false ){
    Seguridad();
    var_alt =0;
}

analogWrite(Pin_Alt,var_alt);

digitalWrite(Pin_Nara_2,HIGH);
delay(200);
digitalWrite(Pin_Nara_2,LOW);
}

LEER_TENSION_RED   = true;
LEER_TENSION_GEN_R = true;
LEER_TENSION_GEN_S = true;

```

Ilustración 64: Modificación de DUTY 2 tras obtener el parámetro Contr_puls_V en Arduino.

Tras la modificación de ambas condiciones debemos de cerciorarnos que hemos cumplido ambas condiciones de frecuencia y tensión a la vez, si se cumplen ambas condiciones procedemos al parpadeo constante del led verde, el programa dispuesto tiene dos bucles de lectura para comprobar el correcto funcionamiento de este. Seguidamente, permutamos los valores bool para continuar a la siguiente condición y no repetir el recién procedimiento.

4.6.4. Tercera condición: Secuencia directa.

Tercera condición para la sincronización y acoplamiento, secuencia directa. Debemos de asegurar que la fase S esta adelantada 120° eléctricos a la fase R, es decir, la fase R esta retrasada 120° respecto a la fase S. Se comprueba con el uso de la función "Comprobar_Secuencia", recordando se realiza la resta para el comienzo de un ciclo en el valor 2050 y se comprueba si la resta es positiva o negativa. Si es positiva la secuencia es directa y CONDICION_3 es true, parpadeo del led verde 5 veces y proseguimos a la siguiente y última etapa.

```
bool Comprobacion_Secuencia (int P1, int P2){
    float Tension_R,Tension_S;
    bool SEC_DIR= false;
    do{
        Tension_R = analogRead(P1);
        Tension_S = analogRead(P2);
    }while (Tension_R <= 2050);

    if (Tension_R>Tension_S){
        cont++;
        if (cont ==5){
            SEC_DIR=true;
        }
        return(SEC_DIR);
    }
}
```

Ilustración 65: Función de comprobación de secuencia en Arduino.

Las variables de entrada son el pin R y S, la salida una variable bool de control para la secuencia directa. Cada comprobación realiza un parpadeo externo de leds naranja, en caso de que no cumpla, el parpadeo corresponde al led rojo hasta que se reinicie el programa.

4.6.5. Cuarta condición: Desfase nulo.

Cuarta y última condición para la sincronización a red, desfase nulo. Debemos de comprobar que la tensión de red y generada están en fase. Para eso realizamos la lectura constante de los valores de red y de la misma fase generada, en este caso, R. La resta entre ambos valores debe aproximarse a 0V, por razones de precisión trabajamos con un margen de 1V de error, correspondiente a cierta cantidad de potencia reactiva que desempeñará el motor tras el acoplamiento.

```
bool Comprobacion_Desfase (int P1, int P2){
float Tension_RED,VGEN_R; cont=0;
bool DESF_NULO= false;
do{
    Tension_RED = analogRead(P1);VGEN_R = analogRead(P2);
    pulsador =digitalRead(Interr_PIN);
    if((Tension_RED-VGEN_R)>-20 && (Tension_RED-VGEN_R)< 20){

        digitalWrite(Pin_Nara_1,HIGH);
        digitalWrite(Pin_Nara_2,LOW);
        delay(100);
        digitalWrite(Pin_Nara_2,HIGH);
        digitalWrite(Pin_Nara_1,LOW);
        delay(100);

        digitalWrite(Pin_Nara_1,HIGH);
        digitalWrite(Pin_Nara_2,LOW);
        delay(100);
        digitalWrite(Pin_Nara_2,HIGH);
        digitalWrite(Pin_Nara_1,LOW);
        delay(100);
        digitalWrite(Pin_Nara_2,LOW);
        DESF_NULO=true;

    }else if((Tension_RED-VGEN_R)<-10){

        DESF_NULO=false;
        digitalWrite(Pin_Rojo,HIGH);
        delay(50);
        digitalWrite(Pin_Rojo,LOW);
        delay(50);
        analogWrite(Pin_Mot_CC,var_cc+5);
    }
    else if((Tension_RED-VGEN_R)> 10){

        DESF_NULO=false;
        digitalWrite(Pin_Rojo_2,HIGH);
        delay(50);
        digitalWrite(Pin_Rojo_2,LOW);
        delay(50);
        analogWrite(Pin_Mot_CC,var_cc-5);
    }
}while (pulsador == 0);

return(DESF_NULO);
}
```

Ilustración 66: Función de control de desfase nulo en Arduino.

Para cada lectura comprobamos si estamos por debajo de 1V o por encima de este, en ambos casos modificaremos el pulso en 5 unidades correspondientes a un 0.12% de pulso constantemente para aproximarnos a ese valor. En caso de que de situarnos por debajo del valor realizamos la señalización mediante parpadeo del LED ROJO 1 y el aumento del DUTY1, en caso de disponernos por encima, LED ROJO 2 y el decremento del DUTY1, en caso de estar en el margen de acoplamiento parpadeo de los LEDs NARANJA 1 y 2. Es justo en el instante en el que nos dispongamos en este margen que debemos de realizar el acoplamiento a red, por lo tanto, dispondremos una variable “bool” denominada “DESF_NULO” para el respectivo control.

4.6.6. Función de Seguridad.

El objetivo es realizar una función que condicione el sistema en un estado de bloqueo por seguridad. Se niegan todas las condiciones de lectura de tensión y frecuencia, al igual que todas las etapas de control entre condiciones de sincronización.

```
void Seguridad(){
    analogWrite(Pin_Mot_CC,4096);
    analogWrite(Pin_Alt,1);

    START = false;
    CONDICION_1=false;
    CONDICION_2=false;
    CONDICION_3=false;
    CONDICION_4=false;

    LEER_FRECUENCIA_RED = false;
    LEER_FRECUENCIA_GEN_R = false;
    LEER_FRECUENCIA_GEN_S = false;

    LEER_TENSION_RED =false;
    LEER_TENSION_GEN_R =false;
    LEER_TENSION_GEN_S =false;

    while(digitalRead(Interr_PIN)==true){
        digitalWrite(Pin_Rojo,HIGH);
        delay(250);
        digitalWrite(Pin_Rojo,LOW);
        delay(250);
    }
}
```

Ilustración 67. Función de seguridad implementada en Arduino.

Seguidamente una señalización de LEDs Rojo mientras el pulsador este activo. Las condiciones para cumplir esta condición de seguridad deberían de ser cuando sobrepasemos los límites de pulso de algún DUTY para su respectiva modificación externa, mientras que en la implementación práctica se realiza cuando deseemos terminar el programa de reinicio con el interruptor en la etapa de condicionamiento de tensión y frecuencia.

5. Implementación y ensayos.

El motor y el alternador con el que operamos presentan las siguientes características.

Alternador

- Denominación: Máquina Síncrona Trifásica de Polos.
- Empresa constructora: Lucas-Nülle LN, Germany.
- Código: SE 2662-3M
- Potencia nominal: 0.3 kW
- Tensión nominal: 230/400V
- Corriente nominal: 0.7/0.45 A
- Velocidad nominal: 1500 rpm
- Factor de potencia: 1/0.8
- Tensión de excitación máxima: 140V
- Corriente de excitación máxima: 0.55 A



Ilustración 68: Alternador de baja potencia.

Motor

- Denominación: Máquina Compound de Corriente Continua.
- Empresa constructora: Lucas-Nülle LN, Germany.
- Código: SE 2662-3C
- Potencia nominal: 0.3 kW
- Tensión nominal: 220 V
- Corriente nominal: 1,4 A
- Velocidad nominal: 2000 rpm
- Tensión de excitación máxima: 220V
- Corriente de excitación máxima: 0.1A



Ilustración 69: Motor cc de baja potencia.

En ambos casos debemos de considerar todos los ensayos para no rebosar sus límites en corriente y tensión de alimentación. Realizando todas las conexiones sin ninguna clase de tensión externa que alimente al motor, con referencia a los planos y varios aparatos de medida que permitan operar dentro del margen de seguridad para el material.

Antes de disponerse en la selección del material, es esencial que el técnico entienda el funcionamiento de la sincronización y acoplamiento a red, funcionamiento del motor y alternador, una vez asimilados estos conceptos globales, el respectivo control y como afectan las diferentes variables en el resultado expuesto se procederá al montaje. Sin comprender correctamente cuál es la función de cada elemento es posible dañar el material al trabajar con tensiones o corrientes fuera del margen recomendado por los fabricantes o tratar con velocidades que sobrepasen el límite nominal de la máquina.

Toda conexión se debe realizar con los planos eléctricos presentes para ir realizando su correspondiente conexión, en este caso el plano eléctrico 1, 2 y 3.

5.1. Conexionado del sistema.

Partimos siempre de la comprobación individual de cada parte del sistema, por ejemplo, el alternador.

Tras centrar el servo-driver al alternador y fijar las tuercas de sujeción, realizar la conexión del alternador con los bornes en estrella y lectura de la corriente de la corriente de excitación mediante multímetro para operar en rango seguro F1-F2. Para operar con el alternador hacer uso del servo-controlador del laboratorio para producir el giro a una frecuencia de 50Hz (1500rpm). Esto se hace conectando los cables de seguridad para medida de temperatura del alternador al sistema de control, encendiendo el sistema y seleccionando la operación de “control de velocidad”, seguidamente apretamos el botón verde y podemos seleccionar la velocidad de giro que necesitemos mediante los pulsadores con diferentes tolerancias de salto (x1, x1.5, x2).

Mediante multímetros medimos la tensión de alimentación, la tensión de caída en el reóstato y la corriente de excitación para operar en el margen de seguridad.

Tras la generación de giro en el rotor procedemos a la alimentación de la excitatriz del alternador y mediante el control de la corriente de excitación aumentando o reduciendo la tensión de alimentación medimos la tensión de

fase (el multímetro de tensión de fase generada se mide conectando una fase cualquiera R-S-T, respectivamente es U2-V2-W2 con el neutro de la estrella, cualquier fase de U1-V1-W1), procedemos a la lectura de la tensión y observamos que efectivamente se comporta con valores fijos de tensión ac y valores “nulos” cc. La medida de la velocidad la podemos realizar con el tacómetro para asegurar el giro respectivo.

Procedemos al conexionado del motor cc para obtener el valor de velocidad de giro alimentando externamente con una tensión de cc (o una onda rectificada completamente, sin trabajar con una cc pues al conectarse al motor se regulará para trabajar a valor constante por las inductancias del motor).

Para la realización de las conexiones del motor usar la referencia “Máquinas de corriente continua, por Miguel Ángel Rodríguez Pozueta” (En el TFG se describen mediante esquemas las conexiones a seguir referenciadas a esta documentación de laboratorio, en general cualquier esquema de motor cc compound nos permite su correcta conexión).

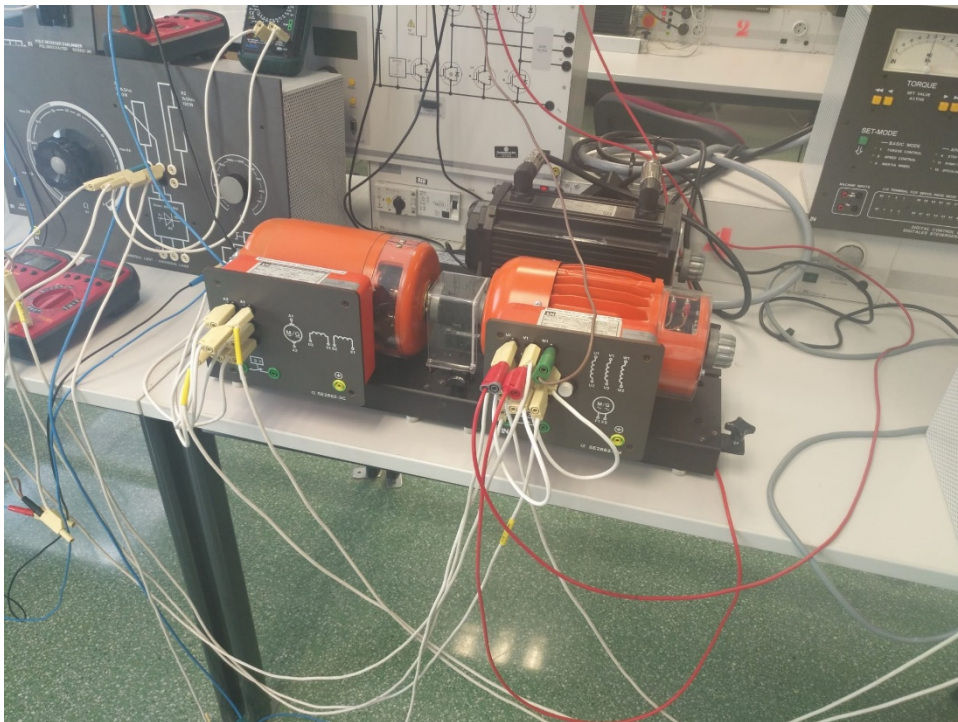


Ilustración 70: Sistema de baja potencia, alternador alineado con el motor cc.

Una vez comprobado individualmente, realizar el acoplamiento conjunto sin el servo-driver mediante alimentación por fuentes diferentes en ambos casos continua o rectificada junto a los reóstatos externos para el motor de corriente continua y la excitatriz del alternador, ilustración 71. Medir la tensión de alimentación en todo momento junto a la que decae en las resistencias externas para observar cómo su variación aumenta o disminuye la velocidad y tensión generada.

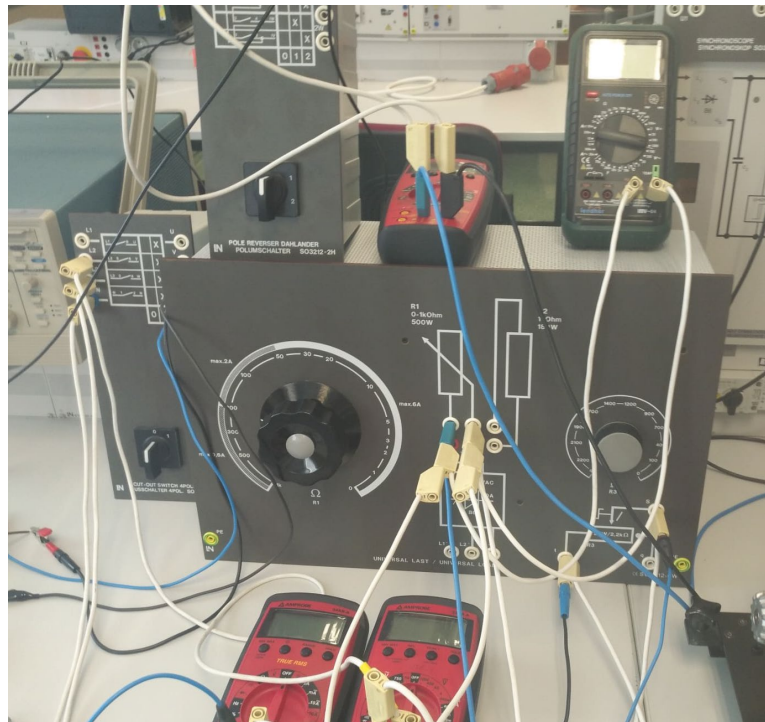


Ilustración 71: Reóstatos externos, izquierda para el alternador derecha para el motor cc.

Los multímetros dispuestos miden la tensión de fase generada (los dos inferiores correspondientes a la fase R y S), la de red (el multímetro rojo) y la tensión que recae en la resistencia del alternador (para comprobar que se cortocircuita correctamente).

Debemos observar al aumentar la tensión de la excitatriz la reducción leve de la velocidad de giro produciendo una caída de tensión en el motor cc.

Tras el funcionamiento correcto del sistema y familiarizarnos con el funcionamiento, procedemos a las conexiones de los drivers y la stm32. Individualmente realizamos las conexiones en las bases de las resistencias, filtros, punto de referencia sin alimentación. Se recomienda realizar las conexiones por bloques, partiendo del bloque de lectura y realizar una comprobación de su funcionamiento con el programa “LEER TENSIÓN Y FRECUENCIA DE RED” en el apartado de Anexos 7.4, alimentando individualmente las tres bobinas por separado con tensión de red monofásica (L1-N). Siguiente bloque el de lectura externa por leds y el último el de control de los transistores.

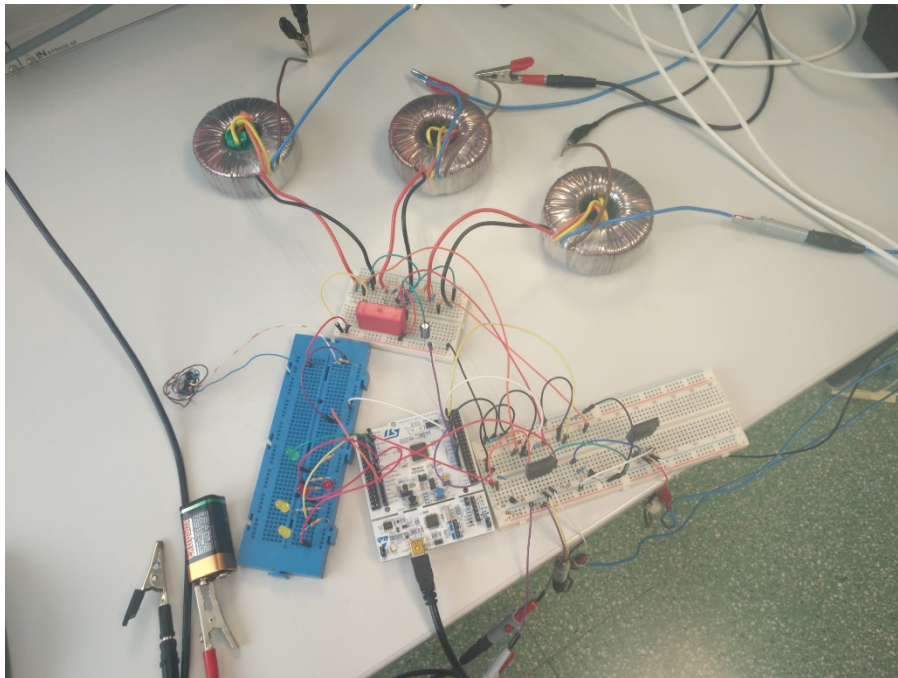


Ilustración 72: Conexión completo del sistema de sincronización automático.

Comprobar que todas las tierras estén interconectadas para tener un punto de referencia, tener claros los Pins de operación y la salida de señal PWM generada por sus salidas correspondiente, se puede comprobar con el programa ejemplo dispuesto en Anexos 7.4. de “Señal PWM” y lectura con un osciloscopio.

En el bloque de lectura debemos de asegurar que trabajamos con una tensión continua de base de 1.655V en caso de que varíe levemente tener en cuenta para los cálculos y la interpretación lógica de los resultados, debemos

de verificar la lectura de la tensión y frecuencia para un mismo valor, por ejemplo, los 230V y 50Hz de trabajo en los tres divisores de tensión y comprobarlo por el “Monitor Serie”, observar los pulsos de diferencia y adaptar la señal leída generada a la de red sumando o restando el valor específico de diferencia en el programa, específicamente en el cálculo de los ERRORES de trabajo de la tensión y la frecuencia.

En el bloque de LEDs trabajar con resistencias para no quemarlos en valores correspondientes a los cálculos realizados en este TFG.

El bloque de control trabaja con drivers elevadores, durante su instalación tener presente la nomenclatura de las patillas dispuesta por sus respectivos Data-Sheet en el apartado de Anexos 7.2. para realizar correctamente su instalación, comprobar que efectivamente se saturan al tratar con la señal PWM y la tensión de +9V externa.

La última conexión debe de ser la tensión de red y eficaz a los transformadores del bloque de lectura, interconectarlos con un seccionador para las fases R-S generadas y la de red, junto a sus respectivos neutros.

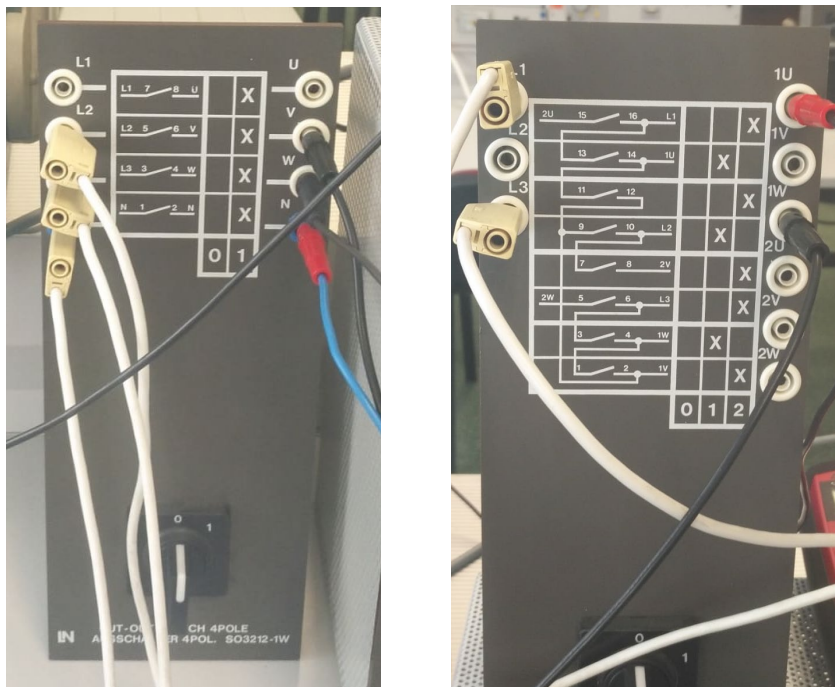


Ilustración 73: Seccionadores para la lectura de la tensión generada y red.

Adicionalmente hemos utilizado un sincronoscopio dispuesto en el laboratorio para asercionarnos del correcto funcionamiento del sistema, la conexión a led apagados se realiza con referencia a la ilustración 74, fase R de red (L1) con fase R generada, idénticamente para las fases S y T. La conexión se debe de realizar en oposición para no provocar un cortocircuito.

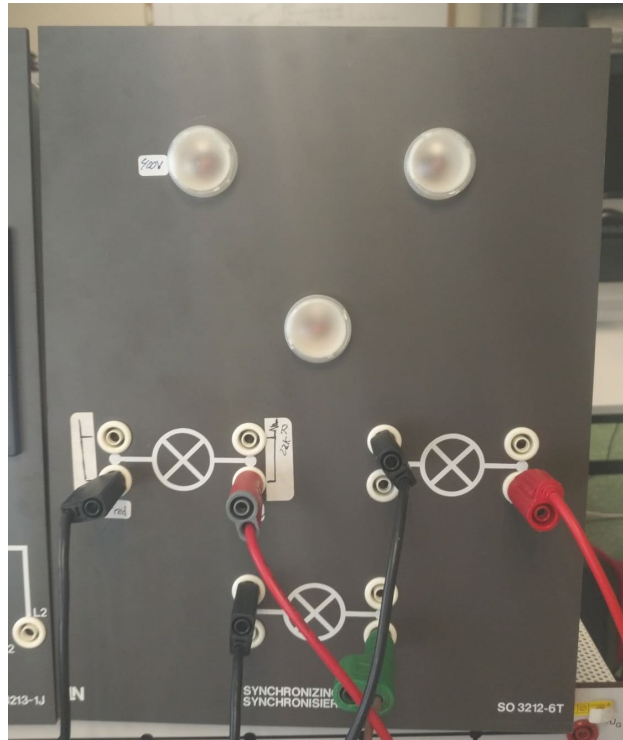


Ilustración 74. Sincronoscopio de lámparas.

No hacemos uso de un sincronoscopio de aguja pues el funcionamiento para obtención de tensión y frecuencia se realiza perfectamente, según nos acercamos al punto de sincronización el motor con el que opera el sincronoscopio de aguja digital provoca una leve caída de tensión que repercute en el alternador como un aumento de la tensión eficaz a excitación de la excitatriz cosntante, por lo tanto, varía la velocidad en el eje del rotor provocando una variación de la F.E.M. y a su vez de la velocidad provocando una insetabilidad en el sistema. Con el sincronoscopio digital de LEDs nos bastamos para observar el correcto funcionamiento.

Para cargar el programa lo realizamos desde el ordenador o mediante el dispositivo de alimentación con conexión USB.



Ilustración 75: Dispositivo de alimentación de la stm32.

La alimentación del motor de corriente continua la realizamos con el rectificador de onda dispuesto en la ilustración 76 izquierda, mientras que la excitación del alternador la realizamos con el rectificador de onda de la ilustración 76 derecha.

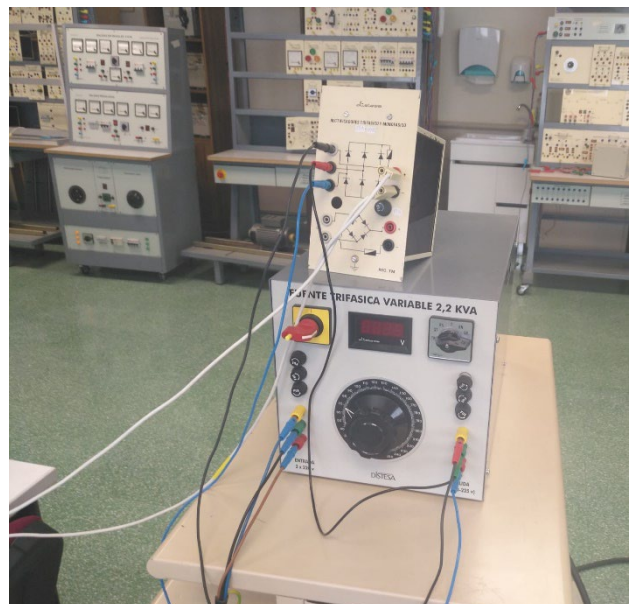
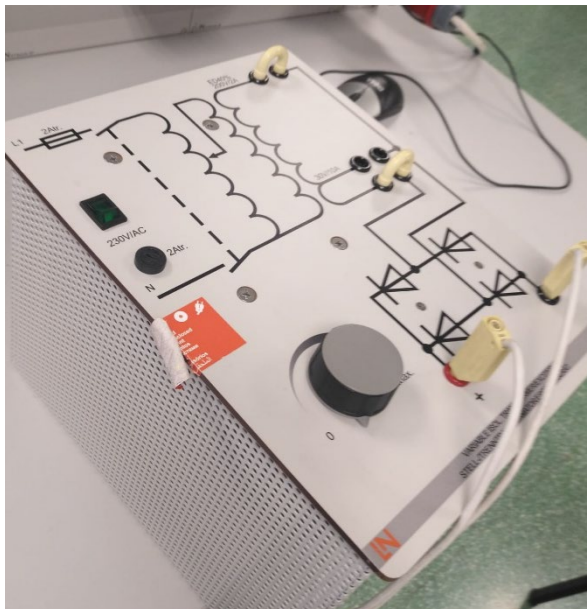


Ilustración 76: Toma de tensión para motor cc y excitatriz del alternador.

IMPORTANTE

No trabajar conectados al ordenador vía USB ni con el osciloscopio cuando operemos con la alimentación de la excitatriz del alternador, ilustración 76 derecha, usar la conexión con el dispositivo de la ilustración 75 (no usar el osciloscopio durante maniobra). Pues provocaremos una derivación a tierra y se activarán las protecciones diferenciales a tierra del laboratorio junto las de las mesas, provocando muy posiblemente la quema de la placa stm32, realizar todas las comprobaciones antes de trabajar con la tensión externa de la excitatriz con referencia a la placa (si no están conectadas sus referencias a tierra no se provocará el disparo a tierra).

Por último, realizamos las conexiones de los reóstatos a la placa para su respectivo control por cortocircuito según esquemas, seguidamente realizar la carga del programa, desconectar del ordenador, apagar el osciloscopio, conectar la alimentación (preferiblemente primero la excitación del alternador para no provocar una velocidad excesiva en el motor cc), permutar los contactores de lectura y proceder al acoplamiento mediante el interruptor de inicio.

Es importante tener en cuenta que trabajamos con poca corriente, pero debemos de estar precavidos por la tensión elevada con la que operamos, no debemos de tocar nada que pueda conectarnos en paralelo al sistema.

Tras el acoplamiento debemos de mantener fijo el valor del pulso para no aumentar o reducir la potencia reactiva o activa. Comprobar que efectivamente tratamos con valores de tensión fijos a la de red a frecuencia estable.

La desconexión la realizamos apagando la tensión de alimentación del motor de cc, la alimentación de la excitatriz del alternador, los interruptores de lectura de red y generada, finalmente, la placa stm32 de red.

El respectivo proceso de sincronismo se puede ver en el **vídeo adjunto** en el archivo de Anexos 7.4, “*Proceso de sincronización a red de potencia infinita*”.

6. Conclusiones.

Concluimos la redacción del “Sistema de sincronización y conexión de alternadores a una red de potencia infinita”, con el análisis de las diferentes necesidades práctica y teóricas exigidas para su respectivo desarrollo, desde la redacción de planos eléctricos, programación de microprocesadores, adaptación de los esquemas al montaje real, estudio de la protección y seguridad del material hasta la implementación conjunta del sistema con un resultado satisfactorio.

El punto de partida para la correcta maniobra de sincronización automática trata los estudios y las pruebas realizadas como necesidad estricta demostrando la importancia en el estudio teórico con anterioridad al montaje práctico. Los valores de tensión de trabajo seleccionados en base a los ensayos de las respectivas máquinas realizadas permiten alcanzar las condiciones exigidas por la red con cierto margen de trabajo en un tiempo relativamente escaso.

La implementación del sistema en el banco de baja potencia, apreciable en el vídeo y en la respectiva memoria en la que se menciona, permite un amplio margen de opciones para diferentes implementaciones futuras, desde la implementación en PCB del sistema automático, depuración de los márgenes y procesos de selección de los ángulos de disparo hasta el control de potencia activa y reactiva que se puede insertar en el sistema, controlados externamente.

7. Anexos.

7.1. Esquemas y Planos.

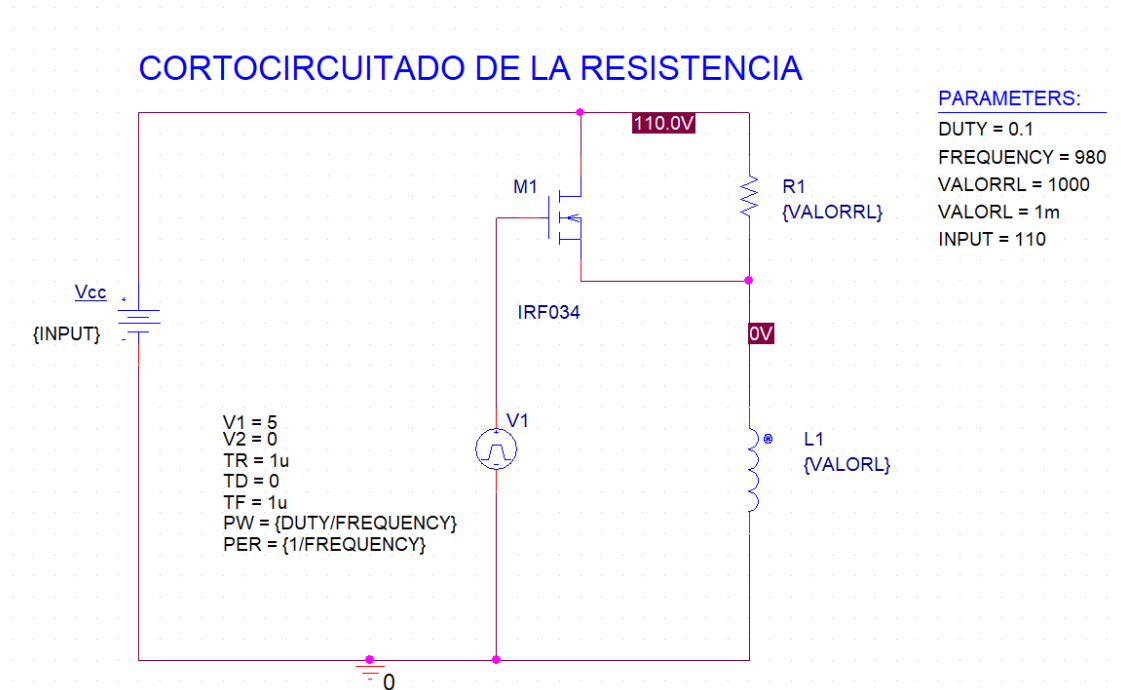


Ilustración 77: Cortocircuitado de la resistencia.

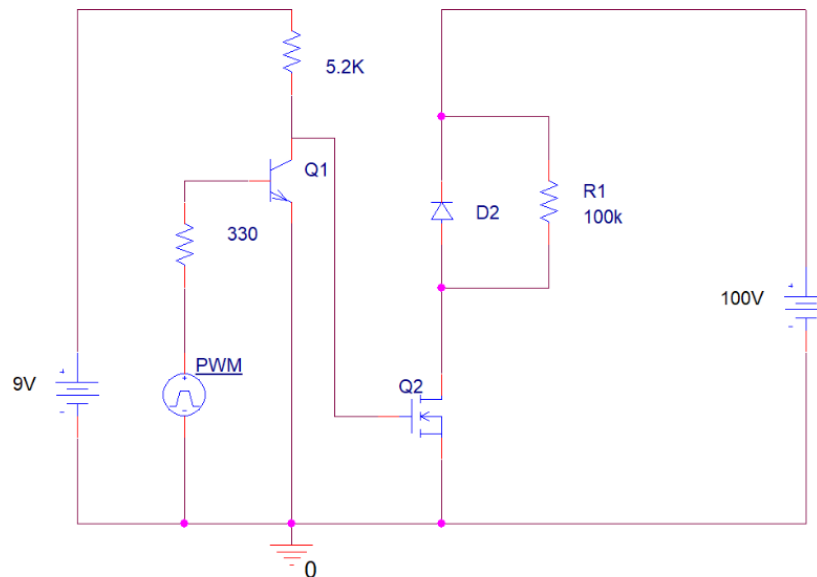
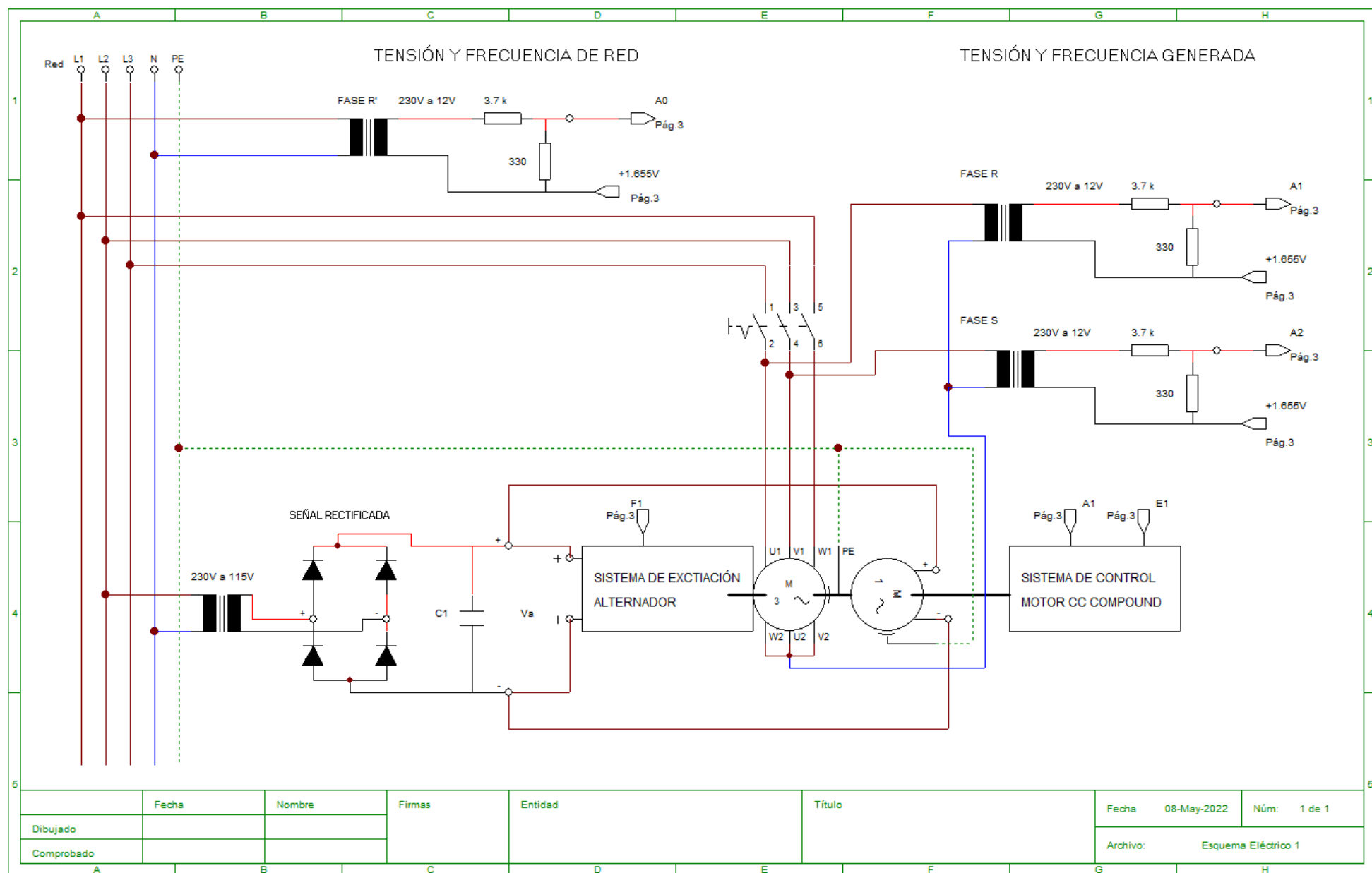
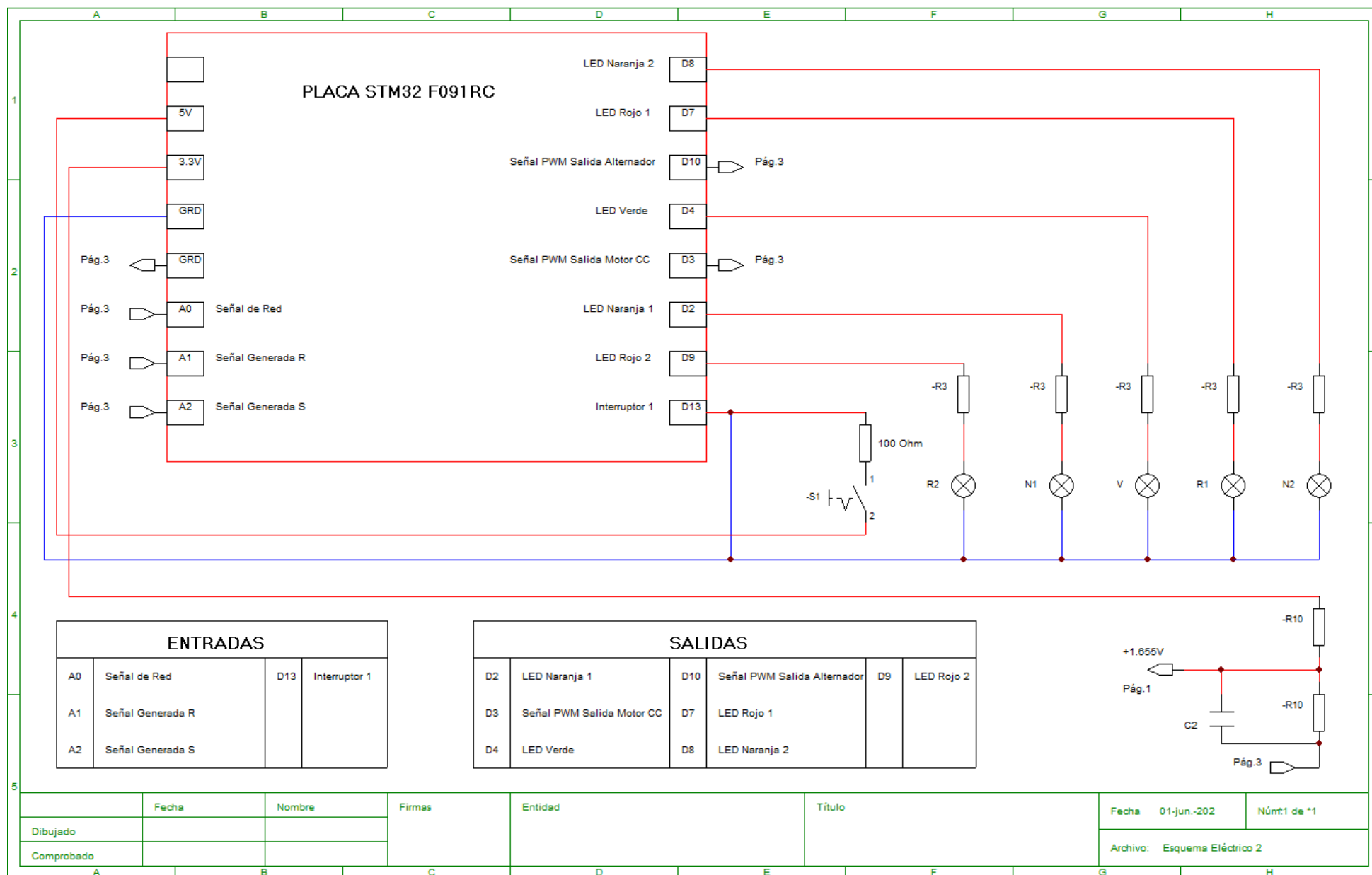
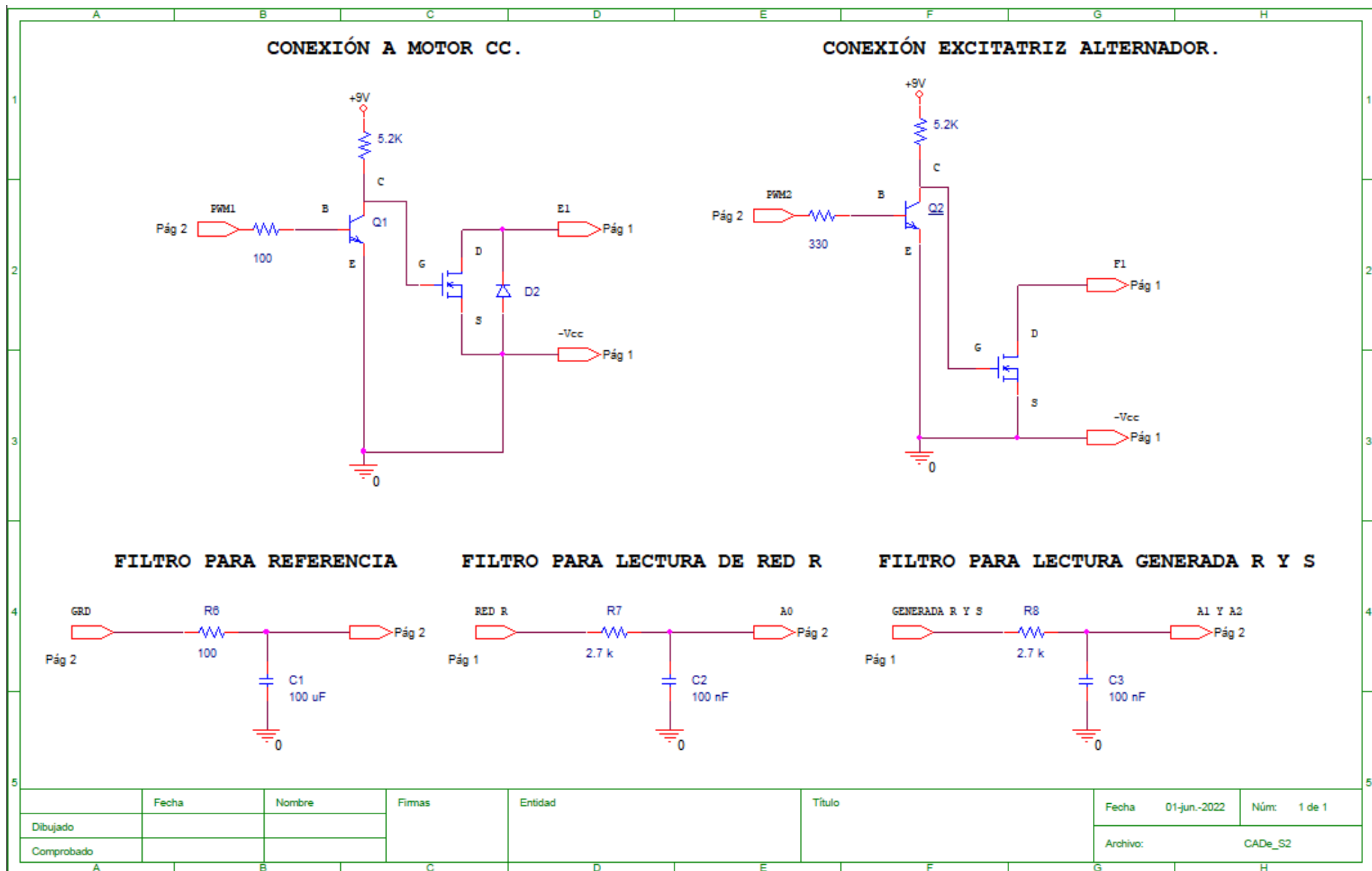


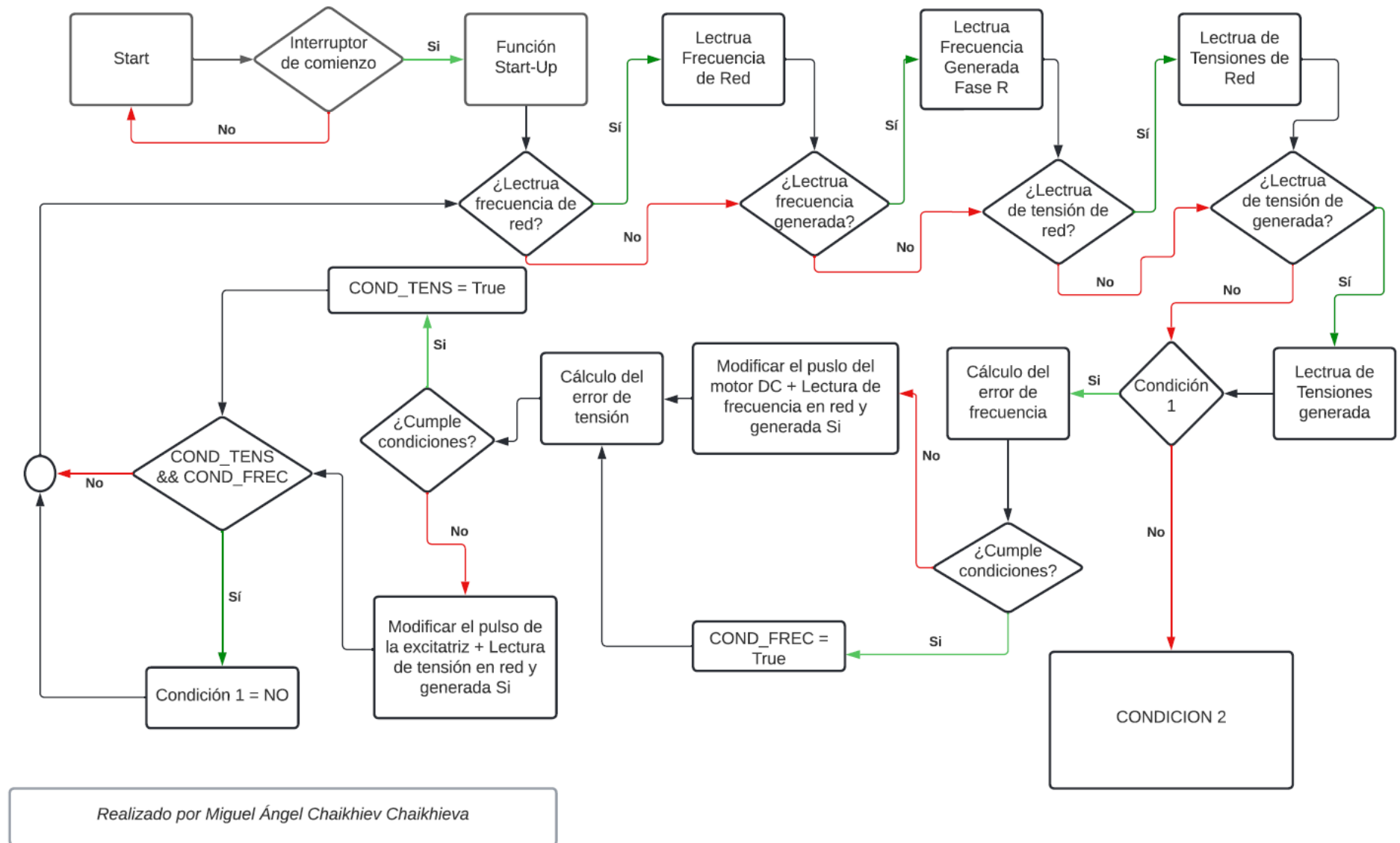
Ilustración 78. Driver de control del transistor.

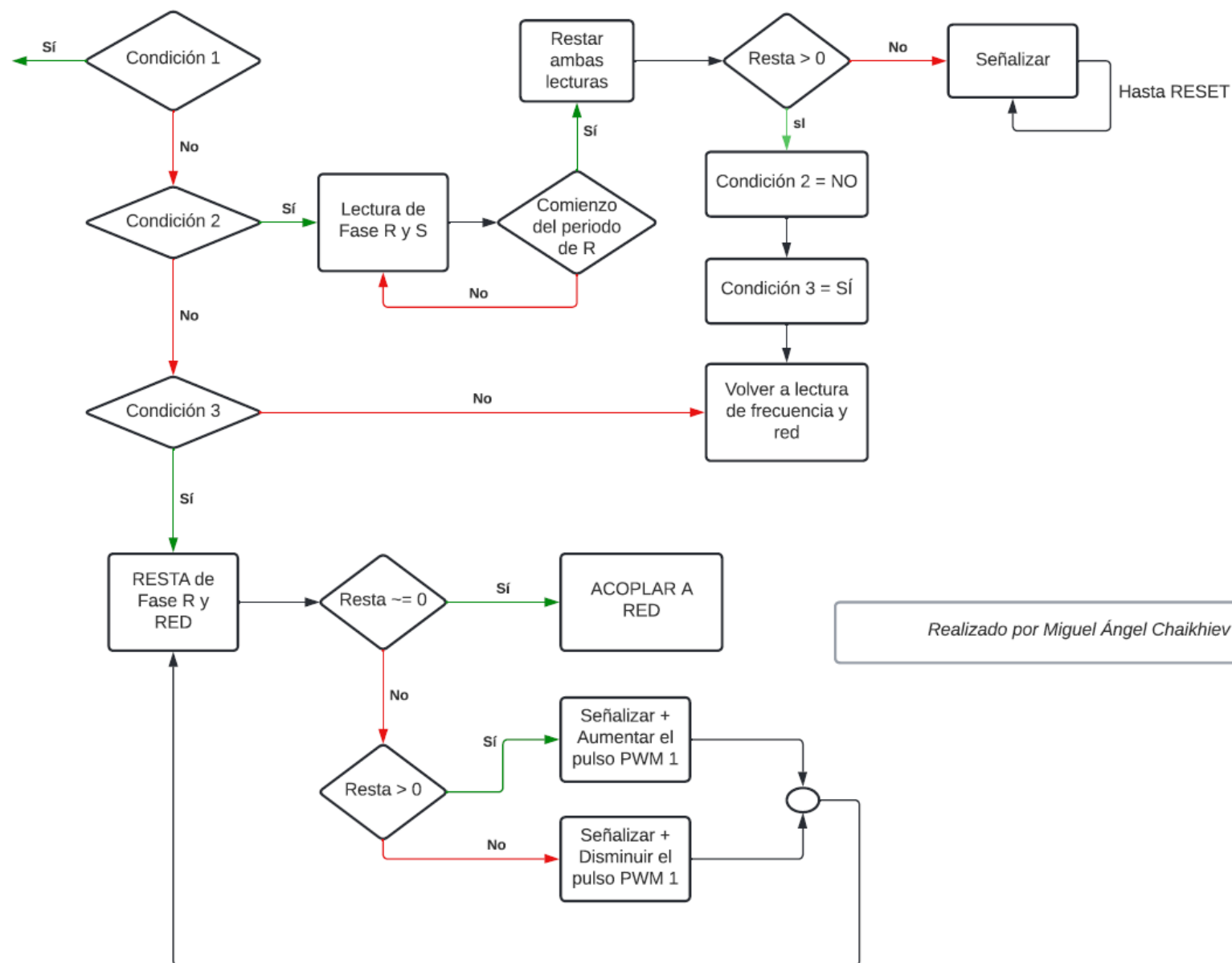


Plano 1: Conexión de las máquinas eléctricas, junto a la toma de lectura.









Realizado por Miguel Ángel Chaikhiev Chaikhieva

7.2. Data-Sheet.



High Speed IGBT with Diode

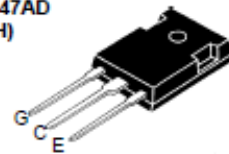
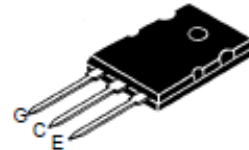
IXSH 30N60BD1
IXSK 30N60BD1
IXST 30N60BD1

Short Circuit SOA Capability

$V_{CES} = 600 \text{ V}$
 $I_{C25} = 55 \text{ A}$
 $V_{CE(sat)} = 2.0 \text{ V}$
 $t_{fi} = 140 \text{ ns}$



Symbol	Test Conditions	Maximum Ratings	
V_{CES}	$T_J = 25^\circ\text{C to } 150^\circ\text{C}$	600	V
V_{CGR}	$T_J = 25^\circ\text{C to } 150^\circ\text{C}; R_{GE} = 1 \text{ M}\Omega$	600	V
V_{GES}	Continuous	± 20	V
V_{GEM}	Transient	± 30	V
I_{C25}	$T_C = 25^\circ\text{C}$	55	A
I_{C90}	$T_C = 90^\circ\text{C}$	30	A
I_{CM}	$T_C = 25^\circ\text{C}, 1 \text{ ms}$	110	A
SSOA (RBSOA)	$V_{GE} = 15 \text{ V}, T_J = 125^\circ\text{C}, R_G = 10 \Omega$ Clamped inductive load, $V_{CL} = 0.8 V_{CES}$	$I_{CM} = 60$	A
t_{SC} (SCSOA)	$V_{GE} = 15 \text{ V}, V_{CE} = 360 \text{ V}, T_J = 125^\circ\text{C}$ $R_G = 33 \Omega$, non repetitive	10	μs
P_C	$T_C = 25^\circ\text{C}$	200	W
T_J		-55 ... +150	$^\circ\text{C}$
T_{JM}		150	$^\circ\text{C}$
T_{stg}		-55 ... +150	$^\circ\text{C}$
M_g	Mounting torque	1.13/10	Nm/lb.in.
Maximum lead temperature for soldering 1.6 mm (0.062 in.) from case for 10 s		300	$^\circ\text{C}$
Weight	TO-247/TO-268	6/4	g
	TO-264	10	g

TO-247AD
(IXSH)TO-268 (D3)
(IXST)TO-264
(IXSK)

G = Gate C = Collector
E = Emitter TAB = Collector

Features

- International standard packages: JEDEC TO-247, TO-264 & TO-268
- Short Circuit SOA capability
- Medium frequency IGBT and anti-parallel FRED in one package
- New generation HDMOS™ process

Applications

- AC motor speed control
- DC servo and robot drives
- DC choppers
- Uninterruptible power supplies (UPS)
- Switch-mode and resonant-mode power supplies

Advantages

- Space savings (two devices in one package)
- Easy to mount with 1 screw (isolated mounting screw hole)
- Surface mountable, high power case style
- Reduces assembly time and cost
- High power density

Symbol	Test Conditions	Characteristic Values ($T_J = 25^\circ\text{C}$, unless otherwise specified)		
		min.	typ.	max.
BV_{CES}	$I_C = 750 \mu\text{A}, V_{GE} = 0 \text{ V}$	600		V
$V_{GE(th)}$	$I_C = 2.5 \text{ mA}, V_{CE} = V_{GE}$	4		V
I_{CES}	$V_{CE} = 0.8 \cdot V_{CES}$ $V_{GE} = 0 \text{ V}$			$200 \mu\text{A}$ 3 mA
I_{GES}	$V_{CE} = 0 \text{ V}, V_{GE} = \pm 20 \text{ V}$			$\pm 100 \text{ nA}$
$V_{CE(sat)}$	$V_{GE} = 15 \text{ V}$			2.0 V 2.7 V
				$I_C = I_{C90}$ $I_C = I_{C25}$

IXYS reserves the right to change limits, test conditions, and dimensions.

98517A(7/00)

© 2000 IXYS All rights reserved

1 - 5

Ilustración 79: Data-Sheet IGBT 1.

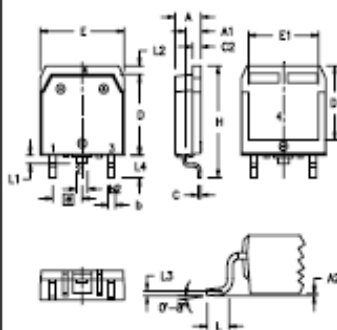


IXSH 30N60BD1 IXSK 30N60BD1 IXST 30N60BD1

Symbol	Test Conditions	Characteristic Values ($T_J = 25^\circ\text{C}$, unless otherwise specified)		
		min.	typ.	max.
g_{fs}	$I_C = I_{CQD}$; $V_{CE} = 10\text{ V}$, Pulse test, $t \leq 300\text{ }\mu\text{s}$, duty cycle $\leq 2\%$	10		S
C_{iss}	$V_{CE} = 25\text{ V}$, $V_{GE} = 0\text{ V}$, $f = 1\text{ MHz}$		3100	pF
C_{oss}			240	pF
C_{res}			30	pF
Q_g	$I_C = I_{CQD}$; $V_{GE} = 15\text{ V}$, $V_{CE} = 0.5 V_{CES}$		100	nC
Q_{ge}			30	nC
Q_{gc}			38	nC
$t_{d(on)}$	Inductive load, $T_J = 25^\circ\text{C}$		30	ns
t_{ri}			30	ns
$t_{d(off)}$	$V_{CE} = 0.8 V_{CES}$; $R_{GS} = 4.7\text{ }\Omega$		150	ns
t_{ri}			140	ns
E_{off}	Note 1.		1.5	2.5 mJ
$t_{d(on)}$	Inductive load, $T_J = 125^\circ\text{C}$		30	ns
t_{ri}			35	ns
E_{on}	$I_C = I_{CQD}$; $V_{GE} = 15\text{ V}$		0.5	mJ
$t_{d(off)}$			270	ns
t_{ri}	$V_{CE} = 0.8 V_{CES}$; $R_{GS} = 4.7\text{ }\Omega$		250	ns
E_{off}			2.5	mJ
R_{thJC}	TO-247 TO-264			0.62 K/W
R_{thCK}			0.25	K/W
R_{thCK}			0.15	K/W

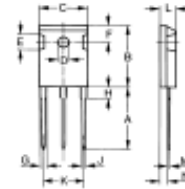
Symbol	Test Conditions	Characteristic Values ($T_J = 25^\circ\text{C}$, unless otherwise specified)		
		min.	typ.	max.
V_F	$I_F = I_{CQD}$; $V_{GE} = 0\text{ V}$ Note 2	$T_J = 150^\circ\text{C}$ $T_J = 25^\circ\text{C}$		1.7 V 2.5 V
I_{FSR}	$I_F = 50\text{ A}$; $V_{GE} = 0\text{ V}$; $T_J = 100^\circ\text{C}$ $V_R = 100\text{ V}$; $-di_F/dt = 100\text{ A}/\mu\text{s}$		2	2.5 A
t_{rr}	$I_F = 1\text{ A}$; $-di_F/dt = 100\text{ A}/\mu\text{s}$; $V_R = 30\text{ V}$ $T_J = 25^\circ\text{C}$		35	50 ns
R_{thJC}				.09 K/W

Notes: 1. Switching times may increase for $V_{CE}(\text{Clamp}) > 0.8 \cdot V_{CES}$, higher T_J or increased R_{GS} .
2. Pulse test, $t \leq 300\text{ }\mu\text{s}$, duty cycle $d \leq 2\%$

TO-268AA (IXST) (D³ PAK)

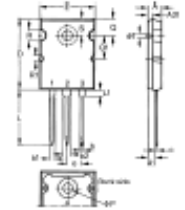
Dim.	Millimeter	Inches
	Min. Max.	Min. Max.
A	4.9 5.1	.193 .201
A ₁	2.7 2.9	.106 .114
A ₂	.02 .25	.001 .010
b	1.15 1.45	.045 .057
b ₁	1.9 2.1	.75 .83
C	.4 .65	.016 .026
D	13.80 14.00	.543 .551
E	15.85 16.05	.624 .632
E ₁	13.3 13.6	.524 .535
e	5.45 BSC	.215 BSC
H	18.70 19.10	.736 .752
L	2.40 2.70	.094 .106
L1	1.20 1.40	.047 .055
L2	1.00 1.15	.039 .045
L3	0.25 BSC	.010 BSC
L4	3.80 4.10	.150 .161

TO-247 AD (IXSH) Outline



Dim.	Millimeter	Inches
	Min. Max.	Min. Max.
A	19.81 20.32	.780 .800
B	20.80 21.46	.819 .845
C	15.75 16.26	.610 .640
D	3.55 3.65	.140 .144
E	4.32 5.49	.170 .216
F	5.4 6.2	.212 .244
G	1.65 2.13	.065 .084
H	- 4.5	- .177
J	1.0 1.4	.040 .055
K	10.8 11.0	.426 .433
L	4.7 5.3	.185 .209
M	0.4 0.8	.016 .031
N	1.5 2.49	.087 .102

TO-264 AA (IXSK) Outline



Dim.	Millimeter	Inches
	Min. Max.	Min. Max.
A	4.82 5.13	.190 .202
A1	2.54 2.89	.100 .114
A2	2.00 2.10	.079 .083
b	1.12 1.42	.044 .056
b1	2.39 2.69	.094 .106
b2	2.90 3.09	.114 .122
c	0.53 0.83	.021 .033
D	25.91 26.16	1.020 1.030
E	19.81 19.96	.780 .786
e	5.45 BSC	.215 BSC
J	0.00 0.25	.000 .010
K	0.00 0.25	.000 .010
L	20.32 20.83	.800 .820
L1	2.29 2.59	.090 .102
P	3.17 3.66	.125 .144
Q	6.07 6.27	.239 .247
Q1	8.38 8.69	.330 .342
R	3.81 4.32	.150 .170
R1	1.78 2.29	.070 .090
S	6.04 6.30	.238 .248
T	1.57 1.83	.062 .072

Min. Recommended Footprint



IXYS

IXSH 30N60BD1 IXSK 30N60BD1 IXST 30N60BD1

Fig.1 Saturation Characteristics

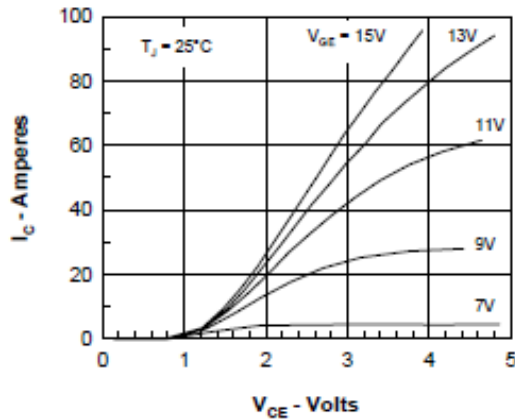


Fig.3 Collector-Emitter Voltage vs. Gate-Emitter Voltage

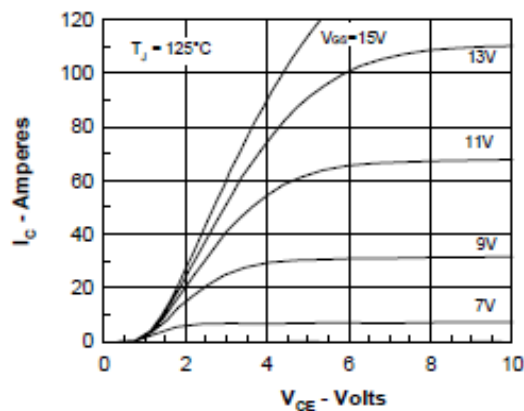


Fig.5 Input Admittance

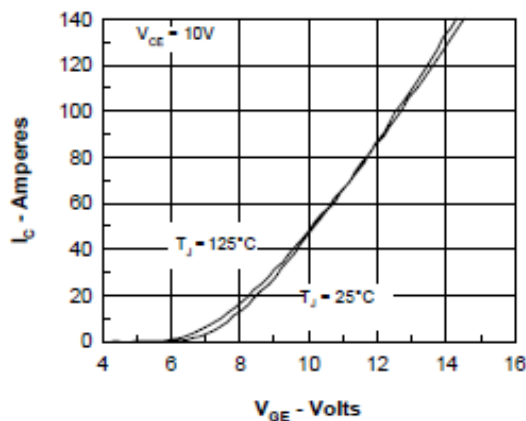


Fig.2 Output Characteristics

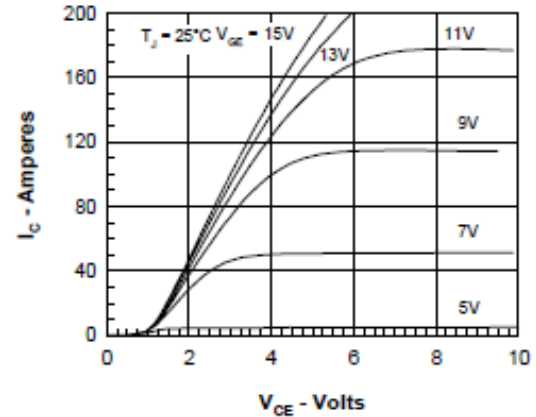


Fig.4 Temperature Dependence of Output Saturation Voltage

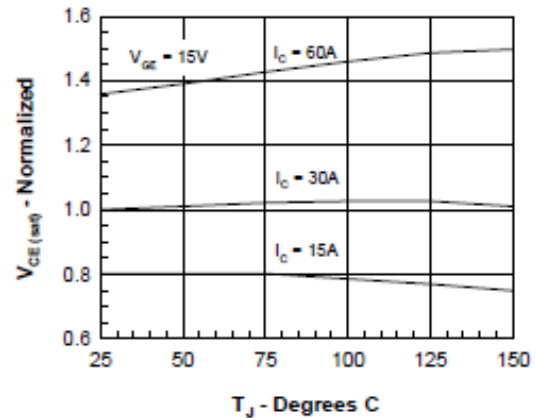
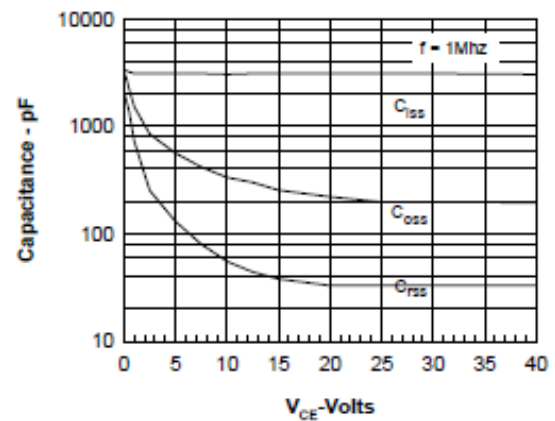


Fig.6 Temperature Dependence of Breakdown and Threshold Voltage



NPN switching transistor

PN2222A

FEATURES

- High current (max. 600 mA)
- Low voltage (max. 40 V).

APPLICATIONS

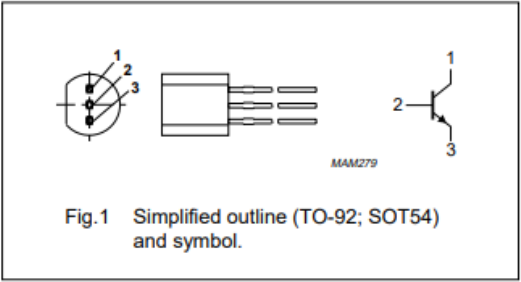
- General purpose switching and linear amplification.

DESCRIPTION

NPN switching transistor in a TO-92; SOT54 plastic package. PNP complement: PN2907A.

PINNING

PIN	DESCRIPTION
1	collector
2	base
3	emitter



LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CBO}	collector-base voltage	open emitter	—	75	V
V_{CEO}	collector-emitter voltage	open base	—	40	V
V_{EBO}	emitter-base voltage	open collector	—	6	V
I_C	collector current (DC)		—	600	mA
I_{CM}	peak collector current		—	800	mA
I_{BM}	peak base current		—	200	mA
P_{tot}	total power dissipation	$T_{amb} \leq 25\text{ }^{\circ}\text{C}$	—	500	mW
T_{stg}	storage temperature		−65	+150	$^{\circ}\text{C}$
T_j	junction temperature		—	150	$^{\circ}\text{C}$
T_{amb}	operating ambient temperature		−65	+150	$^{\circ}\text{C}$

Ilustración 82: Data-Sheet BJT PN2222A, 1.

NPN switching transistor

PN2222A

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th\ j-a}$	thermal resistance from junction to ambient	note 1	250	K/W

Note

1. Transistor mounted on an FR4 printed-circuit board.

CHARACTERISTICS

$T_j = 25\text{ °C}$ unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
I_{CBO}	collector cut-off current	$I_E = 0; V_{CB} = 60\text{ V}$	–	10	nA
		$I_E = 0; V_{CB} = 60\text{ V}; T_{amb} = 125\text{ °C}$	–	10	μA
I_{EBO}	emitter cut-off current	$I_C = 0; V_{EB} = 3\text{ V}$	–	10	nA
h_{FE}	DC current gain	$I_C = 0.1\text{ mA}; V_{CE} = 10\text{ V}$	35	–	
		$I_C = 1\text{ mA}; V_{CE} = 10\text{ V}$	50	–	
		$I_C = 10\text{ mA}; V_{CE} = 10\text{ V}$	75	–	
		$I_C = 10\text{ mA}; V_{CE} = 10\text{ V}; T_{amb} = -55\text{ °C}$	35	–	
		$I_C = 150\text{ mA}; V_{CE} = 1\text{ V}$	50	–	
		$I_C = 150\text{ mA}; V_{CE} = 10\text{ V}$	100	300	
		$I_C = 500\text{ mA}; V_{CE} = 10\text{ V}$	40	–	
V_{CEsat}	collector-emitter saturation voltage	$I_C = 150\text{ mA}; I_B = 15\text{ mA}$	–	300	mV
		$I_C = 500\text{ mA}; I_B = 50\text{ mA}$	1	–	V
V_{BEsat}	base-emitter saturation voltage	$I_C = 150\text{ mA}; I_B = 15\text{ mA}$	0.6	1.2	V
		$I_C = 500\text{ mA}; I_B = 50\text{ mA}$	–	2	V
C_c	collector capacitance	$I_E = i_e = 0; V_{CB} = 10\text{ V}; f = 1\text{ MHz}$	–	8	pF
C_e	emitter capacitance	$I_C = i_c = 0; V_{EB} = 500\text{ mV}; f = 1\text{ MHz}$	–	25	pF
f_T	transition frequency	$I_C = 20\text{ mA}; V_{CE} = 20\text{ V}; f = 100\text{ MHz}$	300	–	MHz
F	noise figure	$I_C = 100\text{ μA}; V_{CE} = 5\text{ V}; R_S = 1\text{ k}\Omega; f = 1\text{ kHz}$	–	4	dB
Switching times (between 10% and 90% levels); see Fig.2					
t_{on}	turn-on time	$I_{Con} = 150\text{ mA}; I_{Bon} = 15\text{ mA}; I_{Boff} = -15\text{ mA}; T_{amb} = 25\text{ °C}$	–	35	ns
t_d	delay time		–	15	ns
t_r	rise time		–	20	ns
t_{off}	turn-off time		–	250	ns
t_s	storage time		–	200	ns
t_f	fall time		–	60	ns

Ilustración 83: Data-Sheet BJT PN2222A, 2.



STM32F091xB STM32F091xC

ARM[®]-based 32-bit MCU, up to 256 KB Flash, CAN, 12 timers,
ADC, DAC & comm. interfaces, 2.0 - 3.6V

Datasheet - production data

Features

Core: ARM[®] 32-bit Cortex[®]-M0 CPU,
frequency up to 48 MHz

Memories

- 128 to 256 Kbytes of Flash memory
- 32 Kbytes of SRAM with HW parity

CRC calculation unit

Reset and power management

- Digital & I/Os supply: $V_{DD} = 2.0\text{ V to }3.6\text{ V}$
- Analog supply: $V_{DDA} = V_{DD}$ to 3.6 V
- Power-on/Power down reset (POR/PDR)
- Programmable voltage detector (PVD)
- Low power modes: Sleep, Stop, Standby
- V_{BAT} supply for RTC and backup registers

Clock management

- 4 to 32 MHz crystal oscillator
- 32 kHz oscillator for RTC with calibration
- Internal 8 MHz RC with x8 PLL option
- Internal 40 kHz RC oscillator
- Internal 48 MHz oscillator with automatic trimming based on ext. synchronization

Up to 88 fast I/Os

- All mappable on external interrupt vectors
- Up to 69 I/Os with 5V-tolerant capability and 19 with independent supply V_{DDIO2}

12-channel DMA controller

One 12-bit, 1.0 μs ADC (up to 16 channels)

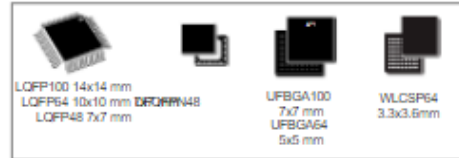
- Conversion range: 0 to 3.6 V
- Separate analog supply: 2.4 V to 3.6 V

One 12-bit D/A converter (with 2 channels)

Two fast low-power analog comparators with programmable input and output

Up to 24 capacitive sensing channels for touchkey, linear and rotary touch sensors

Calendar RTC with alarm and periodic wakeup from Stop/Standby



12 timers

- One 16-bit advanced-control timer for 6 channel PWM output
- One 32-bit and seven 16-bit timers, with up to 4 IC/OC, OCN, usable for IR control decoding or DAC control
- Independent and system watchdog timers
- SysTick timer

Communication interfaces

- Two I²C interfaces supporting Fast Mode Plus (1 Mbit/s) with 20 mA current sink, one supporting SMBus/PMBus and wakeup
- Up to eight USARTs supporting master synchronous SPI and modem control, three with ISO7816 interface, LIN, IrDA, auto baud rate detection and wakeup feature
- Two SPIs (18 Mbit/s) with 4 to 16 programmable bit frames, and with I²S interface multiplexed
- CAN interface

HDMI CEC wakeup on header reception

Serial wire debug (SWD)

96-bit unique ID

All packages ECOPACK[®] 2

Table 1. Device summary

Reference	Part number
STM32F091xB	STM32F091CB, STM32F091RB, STM32F091VB
STM32F091xC	STM32F091CC, STM32F091RC, STM32F091VC

Ilustración 84: Data-Sheet STM32F091RC, 1.

STM32F091xB STM32F091xC

Description

Table 2. STM32F091xB/xC family device features and peripheral counts

Peripheral		STM32F091Cx		STM32F091Rx		STM32F091Vx	
Flash memory (Kbyte)		128	256	128	256	128	256
SRAM (Kbyte)		32					
Timers	Advanced control	1 (16-bit)					
	General purpose	5 (16-bit) 1 (32-bit)					
	Basic	2 (16-bit)					
Comm. interfaces	SPI [I ² S] ⁽¹⁾	2 [2]					
	I ² C	2					
	USART	6	8				
	CAN	1					
	CEC	1					
12-bit ADC (number of channels)		1 (10 ext. + 3 int.)		1 (16 ext. + 3 int.)			
12-bit DAC (number of channels)		1 (2)					
Analog comparator		2					
GPIOs		38	52		88		
Capacitive sensing channels		17	18		24		
Max. CPU frequency		48 MHz					
Operating voltage		2.0 to 3.6 V					
Operating temperature		Ambient operating temperature: -40°C to 85°C / -40°C to 105°C Junction temperature: -40°C to 105°C / -40°C to 125°C					
Packages		LQFP48 UFQFPN48		LQFP64 UFBGA64 WLCSP64		LQFP100 UFBGA100	

1. The SPI interface can be used either in SPI mode or in I²S audio mode.

Ilustración 85: Data-Sheet STM32F091RC, 2.

3.14.2 General-purpose timers (TIM2, 3, 14, 15, 16, 17)

There are six synchronizable general-purpose timers embedded in the STM32F091xB/xC devices (see [Table 7](#) for differences). Each general-purpose timer can be used to generate PWM outputs, or as simple time base.

TIM2, TIM3

STM32F091xB/xC devices feature two synchronizable 4-channel general-purpose timers. TIM2 is based on a 32-bit auto-reload up/downcounter and a 16-bit prescaler. TIM3 is based on a 16-bit auto-reload up/downcounter and a 16-bit prescaler. They feature 4 independent channels each for input capture/output compare, PWM or one-pulse mode output. This gives up to 12 input captures/output compares/PWMs on the largest packages.

The TIM2 and TIM3 general-purpose timers can work together or with the TIM1 advanced-control timer via the Timer Link feature for synchronization or event chaining.

TIM2 and TIM3 both have independent DMA request generation.

These timers are capable of handling quadrature (incremental) encoder signals and the digital outputs from 1 to 3 hall-effect sensors.

Their counters can be frozen in debug mode.

TIM14

This timer is based on a 16-bit auto-reload upcounter and a 16-bit prescaler.

TIM14 features one single channel for input capture/output compare, PWM or one-pulse mode output.

Its counter can be frozen in debug mode.

Ilustración 87: Data-Sheet STM32F091RC, 3.

STM32F091xB STM32F091xC

Description

Table 2. STM32F091xB/xC family device features and peripheral counts

Peripheral		STM32F091Cx		STM32F091Rx		STM32F091Vx	
Flash memory (Kbyte)		128 256		128 256		128 256	
SRAM (Kbyte)		32					
Timers	Advanced control	1 (16-bit)					
	General purpose	5 (16-bit) 1 (32-bit)					
	Basic	2 (16-bit)					
Comm. interfaces	SPI [I ² S] ⁽¹⁾	2 [2]					
	I ² C	2					
	USART	6		8			
	CAN	1					
	CEC	1					
12-bit ADC (number of channels)		1 (10 ext. + 3 int.)		1 (16 ext. + 3 int.)			
12-bit DAC (number of channels)		1 (2)					
Analog comparator		2					
GPIOs		38		52		88	
Capacitive sensing channels		17		18		24	
Max. CPU frequency		48 MHz					
Operating voltage		2.0 to 3.6 V					
Operating temperature		Ambient operating temperature: -40°C to 85°C / -40°C to 105°C Junction temperature: -40°C to 105°C / -40°C to 125°C					
Packages		LQFP48 UFQFPN48		LQFP64 UFBGA64 WLCSP64		LQFP100 UFBGA100	

1. The SPI interface can be used either in SPI mode or in I²S audio mode.

Ilustración 86: Data-Sheet STM32F091RC, 4.

7.3. Webgrafía y Bibliografía.

- [1] Kuznetsov: *Fundamentos de electrotecnia*, Editorial Mir.
- [2] Jesús Fraile Mora: Máquinas Eléctricas. 6ta Edición Daniel W. Hart - Electrónica De Potencia, 2001.
- [3] Andrés Puente, E. Regulación Automática I., 1.989.
- [4] Franklin, G.; Powell, J.D. y Emami-Naeini, A. Control de sistemas dinámicos con retroalimentación. Addison-Wesley Iberoamericana. 1.991.
- [5] Theodore Wildi. Maquinas eléctricas y sistemas de potencia. Prentice Hall, 2007.
- [6] Karl J. Åström y otros. PID Avanzado. Prentice Hall, 2009.
- [7] Katsuhiko Ogata. Ingeniería de control moderna. Prentice Hall, 2005.
- [8] Benjamin C. Kuo. Sistemas de control automático. Prentice Hall, 1996.
- [9] Electronic Design. Agosto 2008.
- [10] Conexión interruptor Arduino por Luis Llamas.
<https://www.luisllamas.es/leer-un-pulsador-con-arduino/>
- [11] Electrónica de Potencia, Daniel W. Hart, Valparaíso University, A Viacom Company Upper Saddle River, NJ 07458.
- [12] Electrónica de potencia: Circuitos, dispositivos y aplicaciones. Muhammad H. Rashid.

[13] Circuitos Alternadores, Automoción: Sistemas de transición, seguridad y confortabilidad.

<https://blogsaverroes.juntadeandalucia.es/profesorjaviergarcia/files/2017/04/Alternador-circuitos-funcionamiento.pdf>

[14] Voltaje de saturación del regulador y excitación del alternador.

https://nosso.com/esp/biblioteca_detalle/51

[15] Sincronizador automático de alternadores, Martín Pérez Pérez y Rubñen Pino Parra.

http://repobib.ubiobio.cl/jspui/bitstream/123456789/2245/1/Perez_Perez_Martin_Andres.pdf

[16] Medir tensiones de 220V/230V con arduino y transformador por Luis Llamas.

<https://www.luisllamas.es/medir-tensiones-de-220v-230v-con-arduino-y-transformador/>

[17] NTP 169: Condiciones de cableado en máquinas (obsoleto).

https://www.insst.es/documents/94886/326853/ntp_169.pdf/22ca3b72-5846-419a-b349-4877a91a024d?version=1.0&t=1617977208603

[18] Instalaciones Eléctricas de Baja Tensión, 2ª Edición por Antonio Colmentar Santos y Juan Luis Hernández Martín, RaMa

[19] Modelo matemático motor DC conexión independiende por Francisco Javier Meza Weber y Paolo Domingo Ramos Morales.

<https://dspace.ups.edu.ec/bitstream/123456789/10257/1/UPS-GT001352.pdf>

[20] Máquinas Eléctricas II: Tema 2. Máquinas de corriente continua por Miguel Ángel Rodríguez Pozueta.

https://ocw.unican.es/pluginfile.php/2806/course/section/2597/02_Maquinas%20de%20Corriente%20Continua.pdf

[21] Crear funciones en Arduino. Programación Modular.

<https://www.wexterhome.com/curso-arduino/crear-funciones-en-arduino/#:~:text=Ha%20llegado%20el%20momento%20de,los%20par%C3%A1metros%20de%20la%20funci%C3%B3n.>

[22] Desarrollo de prácticas de control utilizando la tarjeta stm32 discovery por Pablo Andrés Buestán Andrade.

[23] Control de un Servomotor mediante una tarjeta stm32 nucleo 32 por Kaime Ferre Martínez.

[24] Power System Stability and Control por P.Kundur, McGraw-Hill, Inc.

[25] Software para la evaluación y selección de generadores independientes.

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59012014000100002

[26] Estabilidad angular de un sistema eléctrico de potencia considerando la integración de reguladores de velocidad y voltaje por daniela anabel antamba ávila.

[27] Medición del ángulo de potencia de un generador sincrónico por Játiva Jesús, Escuela politécnica Nacional Quito, Ecuador.

[28] La Máquina Síncrona funcionando en red de potencia infinita y en paralelo contras: Motor síncrono por Miguel Ángel Rodríguez Pozueta.

[29] Sincronización de plantas eléctricas o generadores.

<https://www.electricaplicada.com/sincronizacion-de-plantas-electricas-o-generadores/#condiciones-para-sincronizacion-o-paralelo-de-generadores>

[30] Electroválvulas CEME

https://www.sfcalefaccion.com/pdfcatalogos/DIVISION_5%20electrovalvulas%20ceme.pdf

[31] Diseño y Construcción de un Equipo de Sincronización Automático para Máquinas Síncronas por Rafael Buenadicha Avilés.

[32] Monografía de sistema trifásico.

<https://www.monografias.com/trabajos104/sistema-trifasico/sistema-trifasico>

[33] A-T4, universidad de valencia, otros componentes electrónicos.

https://www.uv.es/~esanchis/cef/pdf/Temas/A_T4.pdf

[34] Getting Started with STM32- Timers and Timer Interrupts.

<https://www.digikey.com/en/maker/projects/getting-started-with-stm32-timers-and-timer-interrupts/d08e6493cefa486fb1e79c43c0b08cc6>