

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA

INVESTIGACION Y DESARROLLO DEL MICROCONTROLADOR “PINGÜINO”

Trabajo de Titulación para optar al Título de
TÉCNICO UNIVERSITARIO EN ELECTRÓNICA

Alumno:

José Manuel Vega Cuevas

Profesor Guía:

Ing. José Llantén Álvarez

2024

RESUMEN

KEYWORDS: Tarjeta de desarrollo Pingüino, Aprendizaje, Variar del Común uso, Arduino, Esp32, Instalación.

En el presente desarrollo investigativo del trabajo de título, se buscará exponer una opción la cual permita investigar más sobre tarjetas microcontroladoras y variar del común uso de la serie Arduino, ESP 32, entre otros. Se realizará una investigación sobre el funcionamiento, Lenguaje de programación, aplicaciones, Módulos compatibles, ventajas y desventajas, pruebas de rendimiento realizadas de forma práctica y costos relacionados a esta tarjeta de desarrollo.

También se buscará información de las diferentes plataformas en la cual opera la tarjeta microprocesadora "Pingüino" tanto su instalación, escritura, funciones que permiten el desarrollo de un programa optimizado.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: TARJETA DE DESARROLLO, ¿QUÉ SON?, ¿CÓMO SE COMPONEN? Y ¿CUÁL ES SU FIN?	
1. TARJETA DE DESARROLLO, ¿QUÉ SON?, ¿CÓMO SE COMPONEN? Y ¿CUÁL ES SU FIN?3	
1.1. TARJETA DE DESARROLLO “PINGÜINO” ¿EN QUÉ SE DIFERENCIAN?	4
1.2. TARJETA DE DESARROLLO “PINGÜINO” ¿CÓMO ESTÁN COMPUESTOS?	5
1.2.1. Regulador de voltaje lineal y conmutado	10
1.2.2. ¿Qué es una memoria Flash, PROM, EPROM?	14
1.2.3. ¿cuáles son Las características del procesador “Pingüino”?	18
1.3. PARTES ÚNICAS DE LA TARJETA DE DESARROLLO “PINGÜINO”	19
1.3.1. Mayor Alimentación en comparación de Arduino, Esp32.	19
1.3.2. Mayor variedad de conectores	22
1.3.3. Pines y conectores disponibles para su uso y para qué sirven	24
1.4. COMUNICACIONES Y PROTOCOLOS DE COMUNICACIÓN DE LA TARJETA DE DESARROLLO PINGÜINO	29
1.4.1. Comunicación serial UART	31
1.4.2. Comunicación serial I2C	32
1.4.3. Comunicación serial SPI	33
1.5. DIAGRAMA ELÉCTRICO	35
1.6. IMPLEMENTACIÓN EN APRENDIZAJE E INDUSTRIA.	36
1.6.1. ¿Que son las tarjetas Arduino? Características	37
1.6.2. ¿Qué es ESP32? Características	38
1.6.3. Tabla Comparativa	39
1.7. ¿CÓMO SE PROGRAMA	41
1.7.1. IDE Pingüino	41
1.7.2. MPIDE	42
1.7.3. MPLAB	44

CAPÍTULO 2: DESARROLLO PRACTICO DE LA TARJETA DE DESARROLLO PINGÜINO

2.	DESARROLLO PRACTICO DE LA TARJETA DE DESARROLLO PINGÜINO	46
2.1.	PLANTEO DEL PROBLEMA TEÓRICO.	46
2.1.1.	Planteo de la problemática “Circuito”	46
2.1.2.	Planteo del circuito	46
2.1.3.	Diagrama de flujo del circuito	52
2.1.4.	Componentes utilizados en el circuito.	52
2.2.	PLANTEO DEL PROBLEMA PRACTICO.	55
2.2.1.	ensamblado del proyecto	56
2.2.2.	resultados del proyecto.	58

CAPÍTULO 3: CONSIDERACIONES Y COSTOS DEL PROYECTO

3.	CONSIDERACIONES Y COSTOS DEL PROYECTO	60
3.1.	CONSIDERACIONES Y RECOMENDACIONES.	60
3.1.1.	Consideraciones en instalación.	60
3.1.2.	Consideraciones en programación.	61
3.2.	PROYECCIÓN ECONÓMICA DEL PROYECTO DE DESARROLLO PINGÜINO	
	“ESTUDIO ECONÓMICO ”	62
3.2.1.	Proyección “mano de obra”	62
3.2.2.	Proyección “Costos Materiales”	63
3.2.3.	Costos finales del proyecto con vista privada y estudiantil.	67
	CONCLUSIONES.	69
	BIBLIOGRAFÍA.	70
	ANEXO.	74

ÍNDICE DE FIGURAS

Figura 1-1.	Microchip procesador PIC32MX440F256H	5
Figura 1-2.	Conector de audio y usb	5
Figura 1-3.	Conector UEXT e ICSP	6
Figura 1-4.	Regulador de voltaje lineal con previa regulación conmutada	6
Figura 1-5.	Conexiones de salida de regulación de voltaje lineal al microchip	7
Figura 1-6.	Conexiones de salida de regulación de voltaje conmutado al conector usb	7
Figura 1-7.	Botones Reset y BUT	8
Figura 1-8.	Conexiones del almacenamiento SD	8
Figura 1-9.	Conector físico SD	9
Figura 1-10.	Conexiones de los pines hacia el procesador	9
Figura 1-11.	Regulador de voltaje conmutado	10
Figura 1-12.	Regulador de voltaje Lineal variable	10
Figura 1-13.	Gráfico de estabilidad y carga de regulación lineal	11
Figura 1-14.	Gráfico de eficiencia energética	12
Figura 1-15.	Gráfico eficiencia/frecuencia regulación conmutada	13
Figura 1-16.	Regulador conmutado y batería “Pingüino”	14
Figura 1-17.	Almacenamientos de datos en una memoria	15
Figura 1-18.	Transferencia de datos de almacenamiento	15
Figura 1-19.	Circuito representativo de una memoria	16
Figura 1-20.	Memoria flash tipo microSD	16
Figura 1-21.	Memoria PROM	17
Figura 1-22.	Procesador de un microcontrolador	18
Figura 1-23.	Microchip ATMEGA328	20
Figura 1-24.	Regulador de voltaje de la tarjeta microcontroladora Arduino UNO	20
Figura 1-25.	Microchip esp32-d2Wd	21
Figura 1-26.	Regulador de voltaje lineal tarjeta desarrolladora ESP32	21
Figura 1-27.	Microchip PIC32MX440F256H	22
Figura 1-28.	Regulador de voltaje conmutado y lineal Pingüino OTG	22
Figura 1-29.	Conector UEXT y ICSP	23
Figura 1-30.	Conector mini USB	23
Figura 1-31.	Conector indicativo ICSP	24
Figura 1-32.	Conector indicativo UEXT	25
Figura 1-33.	Conector indicativo Con1-Power	25
Figura 1-34.	Conector indicativo Con2-Analog	26

Figura 1-35.	Conector indicativo PWR	26
Figura 1-36.	Conector indicativo Con4-Digital	27
Figura 1-37.	Conector indicativo Con5-Digital	28
Figura 1-38.	Conector indicativo USB-OTG	28
Figura 1-39.	Conector indicativo MicroSD	29
Figura 1-40.	Visión física de conector serial	30
Figura 1-41.	Visión física de conector SCL	30
Figura 1-42.	Visión física de conector SPI	31
Figura 1-43.	Lectura de datos asíncrona	31
Figura 1-44.	Codificación NRZ	32
Figura 1-45.	Comunicación I2C	33
Figura 1-46.	Señal SCL “clock” y SDA “datos”	33
Figura 1-47.	Conexión protocolo SPI	34
Figura 1-48.	Comunicación por terminal SPI	34
Figura 1-49.	Diagrama completo tarjeta microcontroladora Pingüino	35
Figura 1-50.	Desarrollo de aplicación simple en IDE pingüino con máquina virtual UBUNTU	42
Figura 1-51.	Desarrollo de aplicación de prueba de tarjeta microcontroladora con MPIDE	43
Figura 1-52.	Interfaz de programación MPLAB	44
Figura 2-1.	Circuito de control	47
Figura 2-2.	Circuito de fuerza	48
Figura 2-3.	Circuito de fuerza editado	49
Figura 2-4.	Programa del Proyecto parte 1	50
Figura 2-5.	Programa del Proyecto parte 2	51
Figura 2-6.	Diagrama de flujo del proyecto	52
Figura 2-7.	Tip 31 BJT tipo NPN	53
Figura 2-8.	Fuente Variable Sorensen	53
Figura 2-9.	Contactador ECX18	54
Figura 2-10.	Relé de control de alta potencia	54
Figura 2-11.	Motor de inducción trifásica.	55
Figura 2-12.	Microcontrolador Pingüino OTG	56
Figura 2-13.	Microcontrolador Pingüino OTG armado	56
Figura 2-14.	Relés de alta corriente y contactores armado.	57
Figura 2-15.	Motor de inducción trifásica armado.	57
Figura 2-16.	Circuito de control en Protoboard.	58

ÍNDICE DE TABLAS

Tabla 1-1.	Características del CPU del microcontrolador pingüino	19
Tabla 1-2.	Pines del conector ICSP	24
Tabla 1-3.	Pines del conector UEXT	24
Tabla 1-4.	Pines del conector Con1-Power	25
Tabla 1-5.	Pines del conector Con2-Analog	26
Tabla 1-6.	Pines del conector PWR JACK	26
Tabla 1-7.	Pines del conector Con4-Digital	27
Tabla 1-8.	Pines del conector Con5-Digital	27
Tabla 1-9.	Pines del conector USB-OTG	28
Tabla 1-10.	Terminales de conexión microSD	29
Tabla 1-11.	Características de la tarjeta microcontroladora Arduino UNO	38
Tabla 1-12.	Características de la tarjeta microcontroladora ESP32 WROOM	39
Tabla 1-13.	Tabla comparativa	40
Tabla 2-1.	Control Lógico y terminales de activación	58
Tabla 3-1.	Costos de profesional	62
Tabla 3-2	Mercado de Protoboard	63
Tabla 3-3	Costos de motor de inducción trifásica	64
Tabla 3-4	Costos TIP31	64
Tabla 3-5.	Costos de Resistencias	65
Tabla 3-6	Costos contactores	65
Tabla 3-7	Costos Disyuntor trifásico	66
Tabla 3-8	Costos Pingüino OTG	66
Tabla 3-9	Costos Fuente de poder variable.	67
Tabla 3-10	Costos totales privado	67
Tabla 3-11	Costos totales estudiantil	68

SIGLAS Y SIMBOLOGIA

A. SIGLAS:

GND	: tierra
DC	: Corriente continua
UEXT	: Universal Extension conector
DMA	: Direct Memory Access
RAM	: Random Access Memory
SD	: Secure Digital (Memoria digital Segura “Memory SD”)
ADC	: Analog Digital Conversor
DAC	: Digital Analog Conversor
EPROM	: Erasable Programmable Read-Only Memory
USB	: Universal Serial Bus
OTG	: On The Go
Py	: Python
RST	: reset
BUT	: botón
PWM	: Modulador por ancho de pulso
PROM	: Programmable Read-Only Memory
UV	: Rayos ultravioleta
MISO	: Master Input Slave Output
MOSI	: Master Output Slave input
microSD	: micro memoria digital segura
SPI	: Serial Peripheral Interface
UART	: Receptor y transmisor universal asincrónico
RX	: Recibir
TX	: Transmitir
D”X”	: Terminal Digital X numero
SCK	: Serial Clock
SS	: Slave Select
SDA	: Puerto de aceptación de línea de datos
SCL	: Línea de reloj en serie
CLOCK	: Reloj
FP	:Factor de potencia.
FS	:Factor de servicio

B. SIMBOLOGÍA

s	: segundos
V	: Volt
A	: Amperios
Ω	:ohm
HP	:Unidad de potencia, Horse power
mA	:Mili Amper
bit/s	:Ratio de Bits por Segundo
B	:Byte
Hz	:hercios
°c	:grados Celsius
SPS	:Disparos por segundos

INTRODUCCIÓN

El objetivo de esta investigación es el poder implementar diferentes aplicaciones con esta tarjeta de desarrollo al esquema de enseñanza estudiantil, permitiendo aumentar la variedad de conocimiento o variedades de tarjetas de desarrollo que se presentan al momento de la enseñanza. En la actual enseñanza su enfoque es el aprendizaje con las tarjetas de desarrollo Arduino uno, ESP 32, Raspberry Pi, siendo este último un microordenador. El objetivo es poder trabajar con esta variante la cual tiene mayor grado de adaptabilidad en los esquemas eléctricos y electrónicos gracias a su mayor tensión de trabajo y su alta tasa de bits de transferencia basándose en el esquema del microchip PIC 32.

La tarjeta de desarrollo “Pingüino” presenta mayores beneficios en aplicaciones del tipo industrial en comparación de su competencia Arduino, la tarjeta de desarrollo Arduino UNO presenta una menor tasa de transferencia de bits por segundo y menores tensiones de alimentación que ofrecen sus salidas, ya que la arquitectura está basada en el microchip ATMEGA 328 siendo este un microcontrolador de 8 bits a comparación de los 32 bits de Pingüino . Además, su baja tensión de alimentación varía en el rango de 7 a 20 volts lo cual le dificulta en la integración de un suministro de mayor rango de voltaje (24V) requiriendo agregar un reductor de voltaje o una alimentación separada al esquema industrial para el correcto funcionamiento en el área. Otra característica que este presenta es un bajo soporte de corriente en las salidas, con valor de 20 mA.

El trabajo en nivel industrial requiere especificaciones mínimas para su funcionamiento las cuales cumple la tarjeta de desarrollo “Pingüino” con su pin de alimentación de 24V que permiten controlar circuitos de control estando integrado en la misma red eléctrica sin la necesidad de alguna fuente de alimentación externa a este, cuenta también de la ventaja de tener una mayor tasa de transferencia de bits por segundo permitiendo velocidades de ejecuciones mayores, a diferencia de otros, operando a 32 bits/s.

Para programar adecuadamente el “Pingüino” se requiere de un computador con sistema operativo Linux o Windows para su trabajo, en relación con sus modelos este cuenta con una gran variedad siendo algunos PIC 32 Pingüino, PIC 32 Pingüino micro, PIC 32 Pingüino MX220, PIC 32 Pingüino OTG, PIC 32 RetroBSD, PIC 32 T795. La diferencia de los modelos indicados anteriormente deriva desde su memoria flash hasta la frecuencia de sus relojes.

CAPÍTULO 1: TARJETA DE DESARROLLO, ¿QUÉ SON?, ¿CÓMO SE COMPONEN?
Y ¿CUÁL ES SU FIN?

1. **TARJETA DE DESARROLLO, ¿QUÉ SON?, ¿CÓMO SE COMPONEN? Y ¿CUÁL ES SU FIN?**

Las tarjetas de desarrollo son herramientas que permiten mediante programación automatizar cualquier proceso productivo o realizar cualquier proyecto el cual requiera un microcontrolador para su control. Los microcontroladores son sistemas computarizados de bajo consumo a comparación de ordenadores domésticos u otros. Permiten mediante su programación controlar señales digitales, análogas, Seriales, BLER, RFID, etc. , siendo necesario en las señales BLER y RFID agregar un módulo bluetooth o radio frecuencia respectivamente.

La necesidad de utilizar las tarjetas de desarrollo se origina de una necesidad de incrementar la productividad de sistemas de producción, o la constante necesidad de hacer más eficaz un sistema para evitar que se produzcan fallas que puedan ocasionar daños materiales o humanos de gran valor. Después de la introducción en el ámbito industrial su efectividad produjo un gran aumento de eficacia en las cadenas de producción, requiriendo que los próximos técnicos u ingenieros tengan conocimiento del uso de estas tarjetas. Los especialistas necesitan saber que se encuentra una gran variedad de tarjetas de desarrollo diferentes para cada necesidad específica los cuales pueden abarcar, por ejemplo:

La gama de tarjeta de desarrollo Arduino los cuales constan de características como 8 bits de transferencia de datos, un conversor análogo a digital y voltajes de salida de entre 3.3V a 5.5V DC con una corriente de 20mA por pin digital. Otra opción son las tarjetas de desarrollo ESP 32 que cuentan con salidas de 3.3V máximos y una corriente por pin de 40mA, su microcontrolador trabaja con 32bits a diferencia de la gama Arduino, siendo el ESP32 superior gracias a su reducido tamaño y bajo costo energético, pero consta de la desventaja que al ser un microcontrolador de tan bajo consumo y tamaño el rango el cual necesita para ser alimentado también es de bajo rango porque no cuenta con un reductor de voltaje agregado, este rango es de 2.2V a 3.6V para su correcto funcionamiento. Otros modelos de la gamma ESP32 cuentan con una mayor capacidad de alimentación de la tarjeta de desarrollo operando hasta 12v.

1.1 **TARJETA DE DESARROLLO “PINGÜINO” ¿EN QUÉ SE DIFERENCIAN?**

En toda la variedad de tarjetas de desarrollo disponibles en el mercado sus diferencias son notables entre sí, desde algunos que tienen una mayor capacidad de voltaje de alimentación, mayores tasas de transferencia de datos, integración de conectores únicos como el UEXT, ICSP, etc. La corriente de los pines de salida de la tarjeta de desarrollo Pingüino 185mA presentan una capacidad similar a la tarjeta de desarrollo Arduino UNO 200mA.

Las tarjetas de desarrollo “Pingüino” son tarjetas de hardware abierto que basan su arquitectura en el microcontrolador Pic 32MX440F256H el cual tiene características que se indican a continuación.

32 bits de transferencia de datos del procesador: el cual es más que el microprocesador ATMEGA328 que solo tiene 8 bits e igualando a las tarjetas de desarrollo ESP32 que también tienen 32 bits de transferencia, esto se ve reflejado en la complejidad de los programas. Si un microprocesador tiene un límite de 8 bits solo podrá controlar un total de 256 valores únicos a la vez. Esto le da una ventaja en velocidad al momento de ejecutar ordenes complejas, los 32 bits del microcontrolador proporcionan una velocidad de ejecución continua de 4 millones de valores a la vez. Un procesador de 32 bits también permite tener una mayor eficiencia energética en comparación al de 8 bits haciéndolo más rentable.

Conectores únicos: tales como el conector DMA “conector de acceso directo a la memoria” que es un conector el cual tiene acceso directo a la memoria del microcontrolador es utilizado por algunos periféricos para que tenga una comunicación directa entre el dispositivo y la memoria programada, la cual puede ser la memoria flash. Consta también con un conector UEXT el cual es un conector estándar que permite el uso de diferentes periféricos con una rápida conexión y desconexión entre las interfaces de comunicación, soporta I2C, SPI, UART, entre otros. Para poder cargar o conectar con su aplicación por defecto siendo esta “Pingüino IDE” se utiliza un terminal tipo USB mini.

Voltajes de Alimentación: La tarjeta de desarrollo Pingüino tiene una característica que la diferencia de su competencia la cual es su capacidad de ser alimentado mediante una señal DC constante variable entre 9V a 24V, este puede ser un detalle menor pero al ser una señal de voltaje igual a la utilizada en la industria permite conectar la tarjeta a la misma red, en comparación su competencia carece de esta ventaja siendo Arduino 9V-12v y ESP32 3.3V-12V respectivamente, esto genera que para poder utilizar las tarjetas de desarrollo Arduino y ESP32 es necesario tener una alimentación externa que los alimente con los respectivos voltajes.

1.2 TARJETA DE DESARROLLO “PINGÜINO” ¿CÓMO ESTÁN COMPUESTOS?

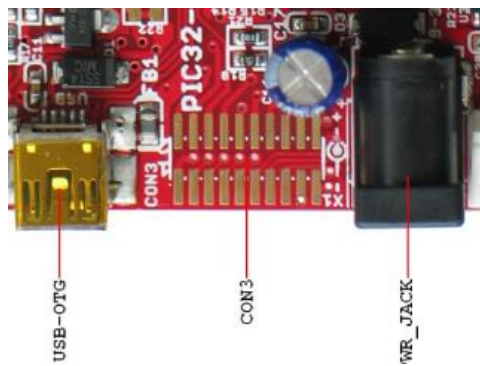
Las tarjetas de desarrollo “Pingüino” están compuestos por varias etapas que en su conjunto permiten el correcto funcionamiento, el principal de ellos es el procesador el cual se encarga de realizar todos los procesos lógicos que puede ejecutar el microprocesador desde la programación hasta la puesta en ejecución del programa a trabajar.



Fuente: <https://www.microchip.com/en-us/product/PIC32MX440F256H>

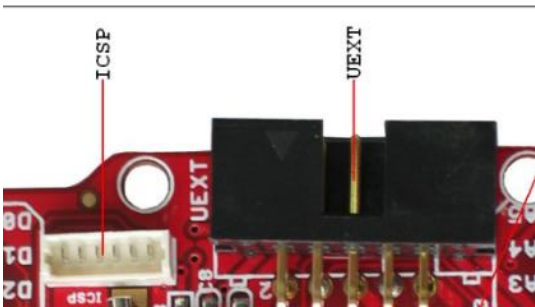
Figura 1-1. Microchip procesador PIC32MX440F256H

La tarjeta de desarrollo pingüino tiene características que lo asemejan a su competencia en el uso adecuado del mismo. Esta cuenta de conectores encargados desde la alimentación, comunicación con la plataforma de programación, conexión de audio como se puede observar en la figura 1-2. y dos conectores para módulos que se pueden observar en la figura 1-3.



Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGÜINO-OTG/>

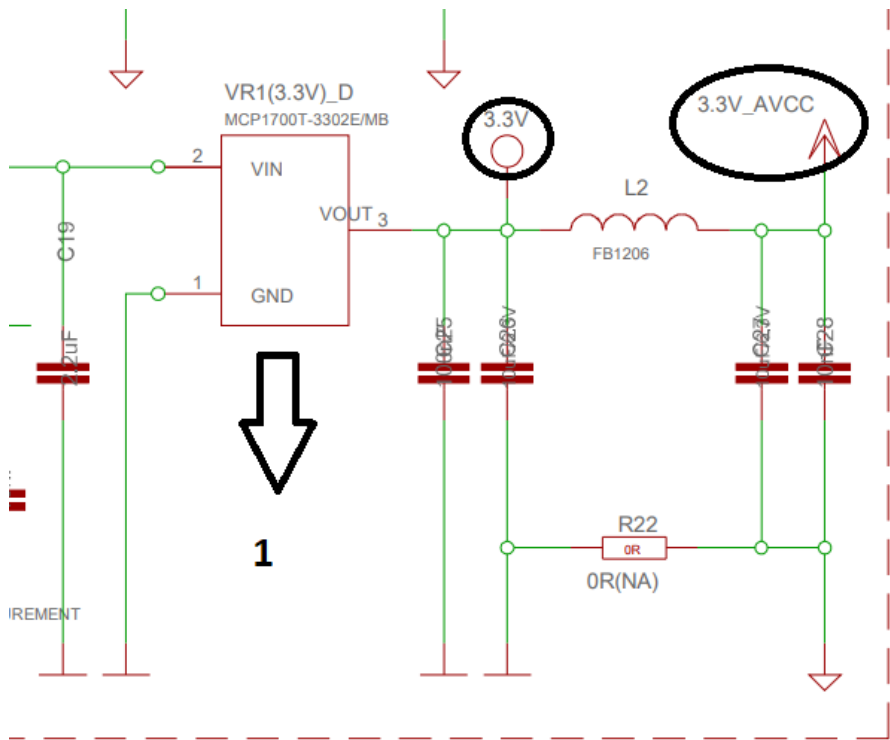
Figura 1-2. Conector de audio y usb



Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/>

Figura 1-3. Conector UEXT e ICSP

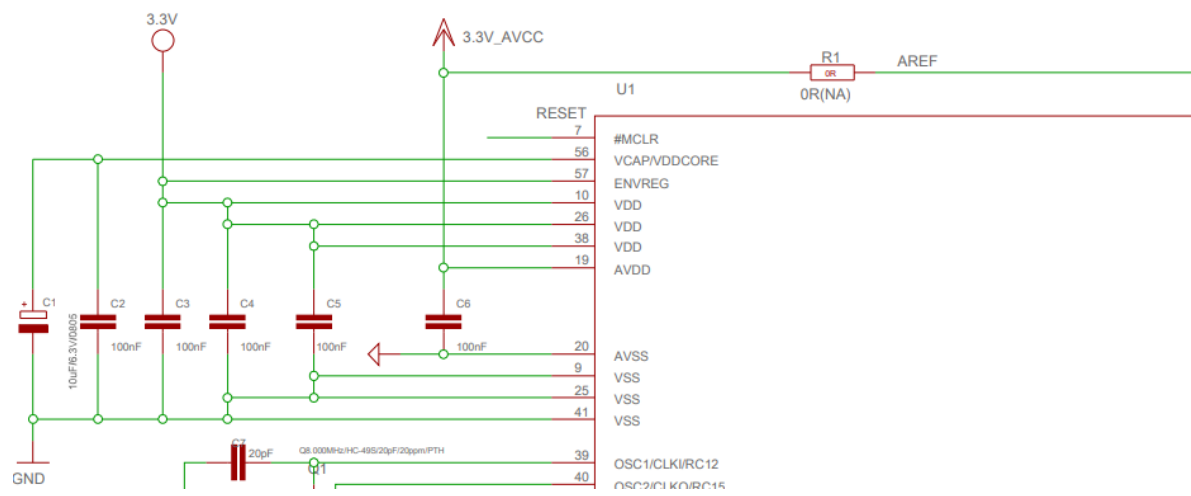
Otra etapa de la cual está compuesto esta tarjeta de desarrollo es el “Power Suply” como se observa en la figura 1-4. Esta etapa de la tarjeta es la encargada de suplementar correctamente de voltaje de alimentación al microcontrolador. En general los procesadores de las tarjetas de desarrollo necesitan un voltaje de bajo rango, en esta ocasión al ser un microchip PIC32MX440F256H consta de un rango entre 2.3V a 3.6V, las tarjetas de desarrollo normalmente se alimentan con un rango de 9V a 12V, pero en el caso de la tarjeta “Pingüino” tiene un rango de 9V a 24V correspondiente a lo comentado en el punto 1.1. para poder evitar que se cree un sobre voltaje e inutilice el microchip, la fuente de alimentación tiene incorporado un regulador de voltaje el cual convierte el rango de 9V-24V en 3.3V necesarios para el correcto funcionamiento como se puede observar en la figura 1-4. Se Representa el regulador lineal marcado con el numero uno para una mayor comprensión



Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO/open-source-hardware>

Figura 1-4. Regulador de voltaje lineal con previa regulación conmutada

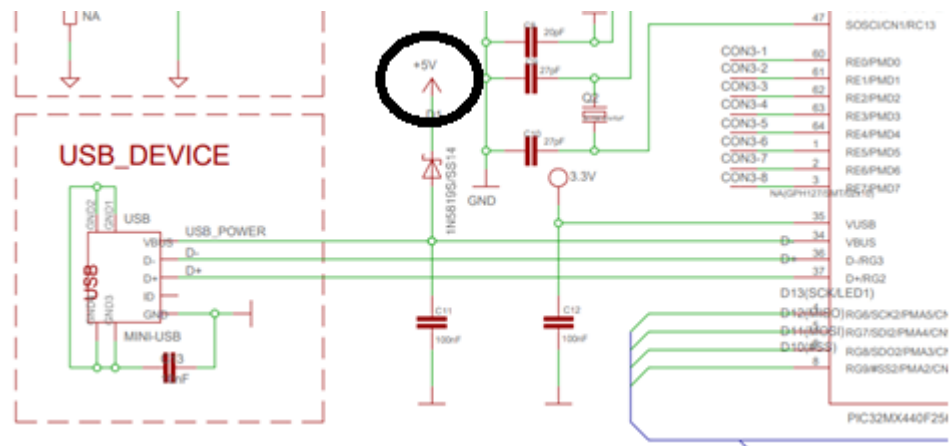
Según lo indicado en la Figura 1-4 es posible apreciar que la salida 3.3V_AVCC la cual se encerró dentro de un círculo para mayor comprensión, está directamente conectada a las alimentaciones necesarias del microchip de la Figura 1-5, este suministro de voltaje también es compartido a otros componentes como leds, pines de salida de voltaje, conectores tales como el UEXT.



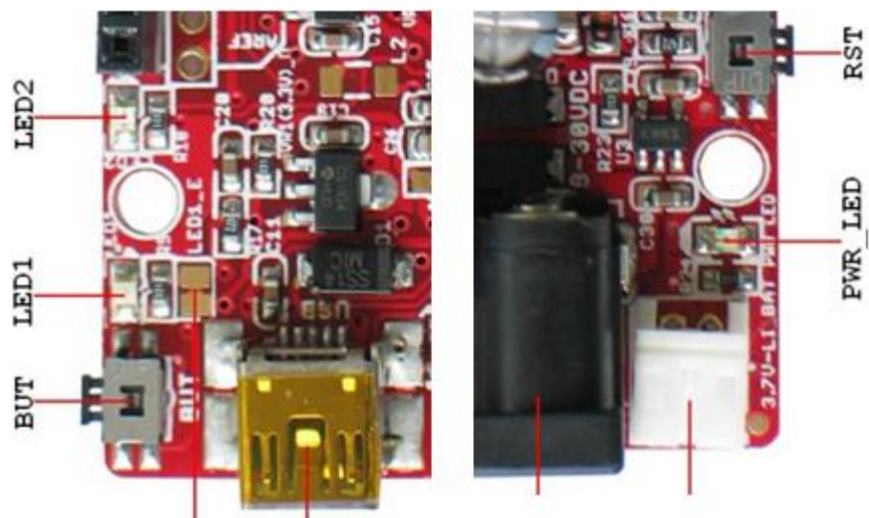
Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO/open-source-hardware>

Figura 1-5. Conexiones de salida de regulación de voltaje lineal al microchip

Según lo comentado previamente, la tarjeta Pingüino cuenta con un regulador conmutado el cual se encarga de suministrar voltaje de 5V como se puede observar en la figura 1-6. Marcado en un círculo negro para mayor comprensión este punto suministra a otros componentes necesarios del sistema tales como el usb y los pines de salidas de voltajes del conector.



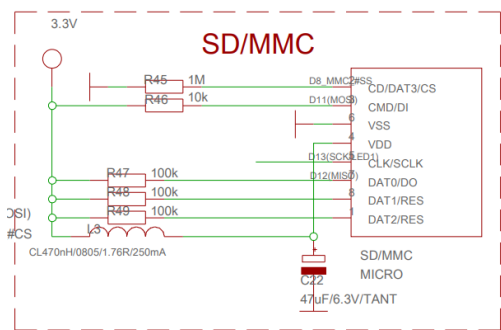
La tarjeta de desarrollo Pingüino cuenta con un botón de reset con la utilidad de interrumpir el programa en ejecución el cual se muestra en la figura 1-7. Reiniciar desde el principio con el botón cortará el suministro de voltaje y lo retornará a su valor inicial. Su uso produce una facilidad manual de probar los programas ejecutados sin la necesidad de interrumpirlos digitalmente mediante el compilador. El otro botón es el BUT el cual es un botón configurable para el usuario mediante programación, puede ser un reset, una interrupción, o interruptor. En la figura 1-7 se recortó y señaló una visualización completa de la distribución de estos botones para una mayor comprensión, se puede apreciar que tiene dos leds programables y un led indicativo del estado del controlador “PWR LED”.

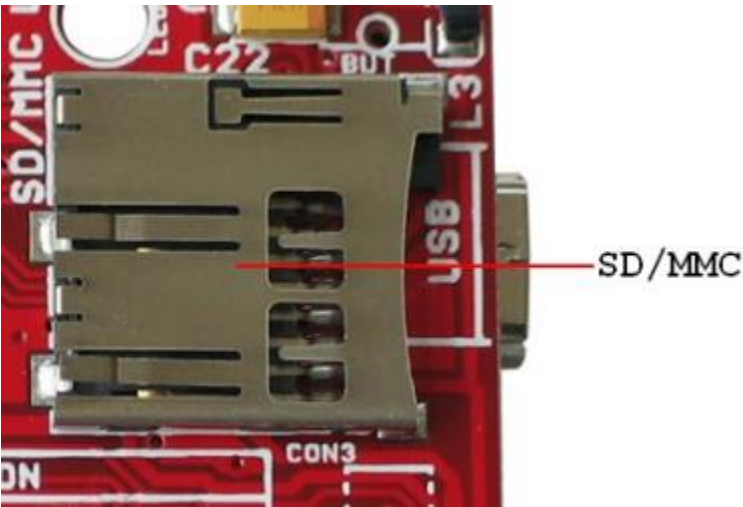


Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO/open-source-hardware>

Figura 1-7. Botones RST y BUT

Para complementar otra de las características que tiene la tarjeta Pingüino es la capacidad de tener una memoria microSD para almacenar información dependiendo lo que requiera la programación escrita, por ejemplo, se instala un módulo de cámara el cual almacene las capturas en el almacenamiento SD. La memoria SD tiene una directa conexión con el procesador, permitiendo el directo acople de este como se muestra en la Figura 1-8. En la figura 1-9. se puede apreciar el slot de la memoria SD al inverso de la placa normalmente.

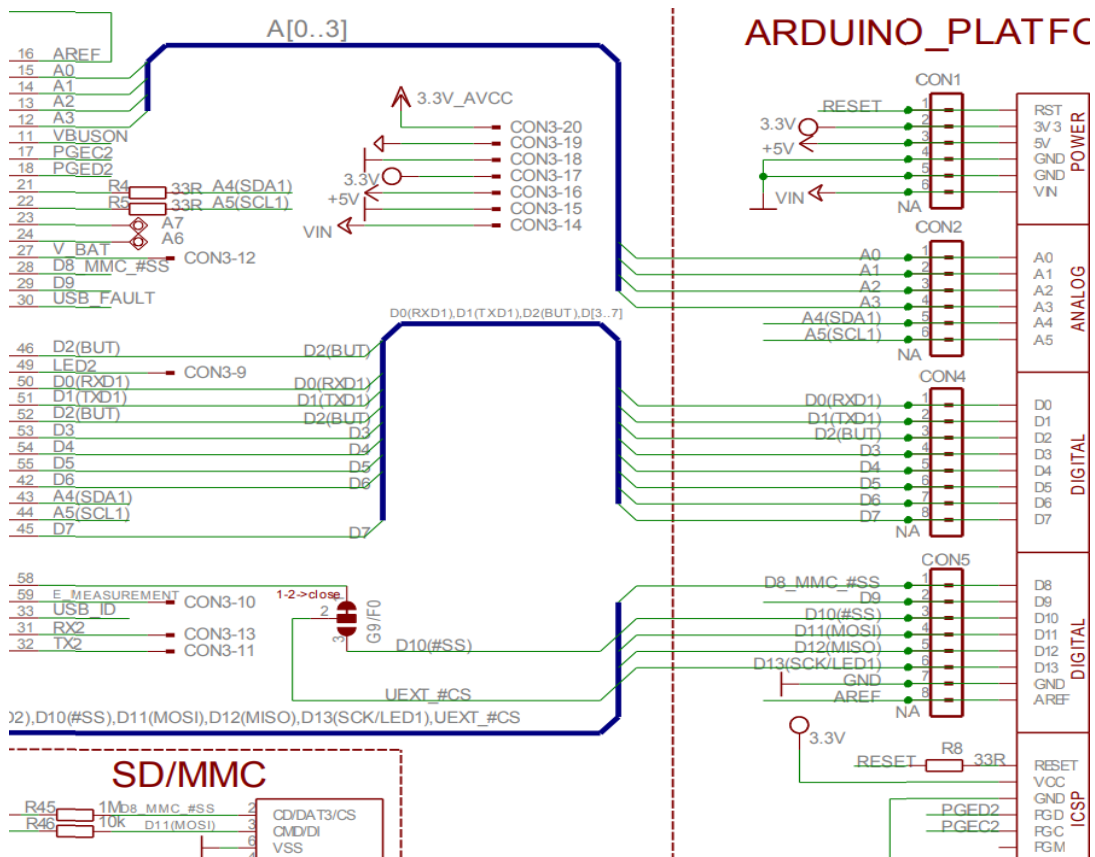




Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO/open-source-hardware>

Figura 1-9. Conector físico SD

La última característica por mencionar del reconocimiento de partes de la tarjeta de desarrollo son los pines que permiten una directa conexión de las salidas de procesador al circuito, en general esto permite controlar y ejecutar toda la programación escrita en la tarjeta de desarrollo Pingüino, sus conexiones abarcan desde comunicación UART, I2C, DAC, ADC, salidas y entradas digitales, alimentación de 5V y 3.3V, un reset del microchip, una señal de voltaje de referencia, etc. Se puede observar en la figura 1-10. Su conexión directa al procesador. “se editó para mayor claridad”



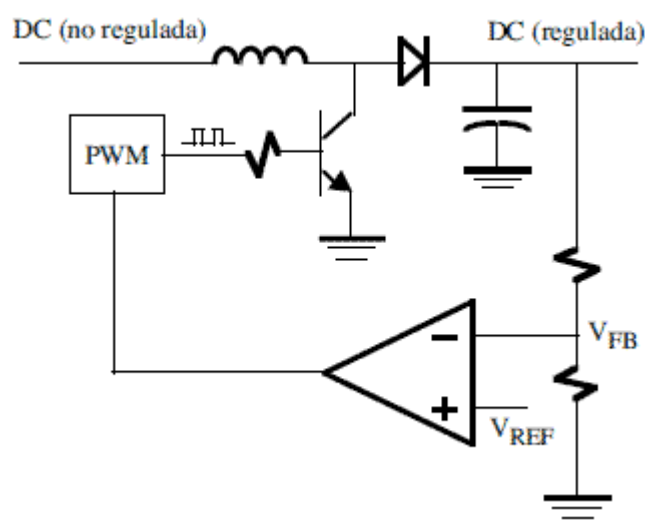
Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO/open-source-hardware>

Figura 1-10. Conexiones de los pines hacia el procesador

1.2.1 Regulador de voltaje lineal y conmutado

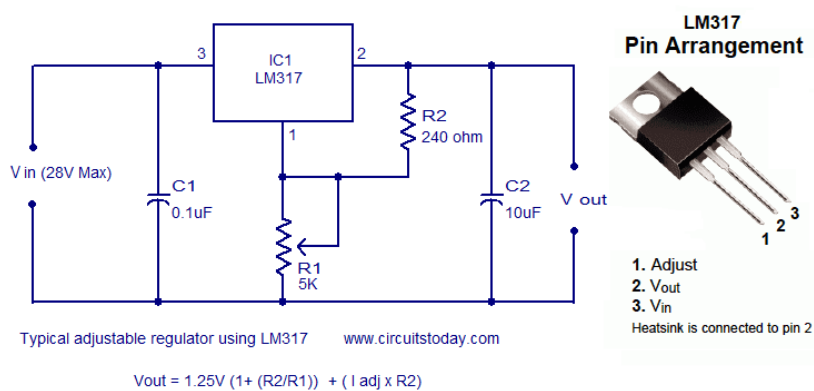
Las tarjetas de desarrollo de diferentes distribuidoras tales como el Pingüino, Arduino, ESP32 son tarjetas de bajo consumo, por lo cual requieren voltajes de bajo rango siendo estos de 2.6V a 3.3V requeridos en el microchip, para satisfacer la necesidad se emplea reguladores de voltajes que permiten bajar una entrada la cual puede ser de alto voltaje a uno de bajo voltaje.

Las tarjetas emplean dos configuraciones o tipos de reguladores de voltaje, estos pueden ser reguladores de voltaje lineales o reguladores de voltaje conmutado como se observa en la figura 1-11 siendo la conmutada y la figura 1-12 el regulador lineal.



Fuente: <https://unicrom.com/reguladores-de-conmutacion-convertidores-dc-d/>

Figura 1-11 Regulador de voltaje conmutado



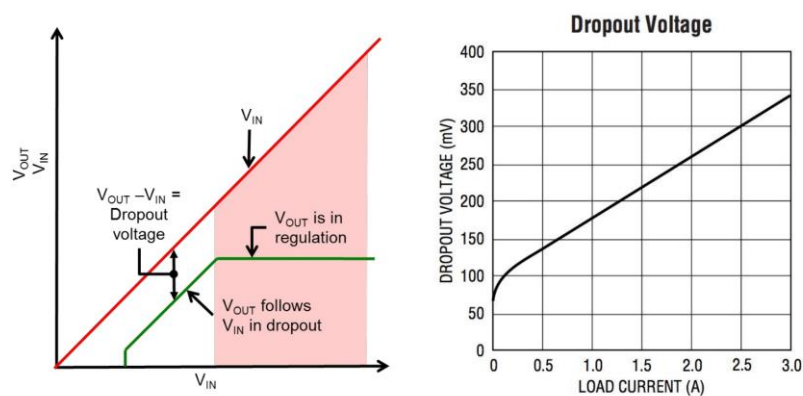
Fuente: <https://www.hwlibre.com/lm317/>

Figura 1.12 Regulador de voltaje Lineal variable

Los reguladores mostrados en la figura 1-11 y figura 1-12 presentan un diseño simple para comprender mejor el funcionamiento de estos, los diseños pueden variar según indique el plano de las tarjetas de desarrollo, estos pueden variar en busca de una mayor eficiencia o estabilidad. Cada uno tiene características únicas los cuales se usarán según se requiera o se preferirá de una u otro. El regulador conmutado regula el voltaje mediante el transistor que controla la carga del capacitor gracias el modulador de ancho de pulso que tiene incorporado, mientras que el regulador lineal controla el voltaje mediante combinación de resistencias sin necesidad que oscile la señal para que esta se regule en los valores nominales necesarios, ahora se presentarán las ventajas y desventajas de los reguladores mencionados.

1.2.1.1 Ventajas regulador de voltaje Lineal.

El regulador lineal tiene la ventaja la cual al ser una reducción DC-DC sin conmutación produce que su salida tenga bajo nivel de ruido, por ende, su señal es estable. Esta característica es ideal para procesadores que requieran precisión en el voltaje de alimentación para su correcto funcionamiento. La regulación lineal requiere un voltaje mínimo para funcionar, el cual es proporcional al voltaje de alimentación hasta un máximo. En la figura 1-13 se puede apreciar un gráfico de voltaje de ingreso contra el voltaje de salida y voltaje de salida contra la corriente de carga necesaria.

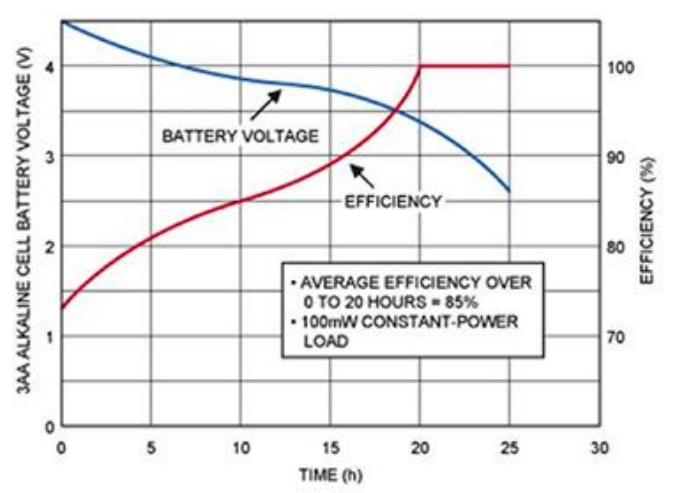


Fuente: <https://www.digikey.com/es/articles/understanding-the-advantages-and-disadvantages-of-linear-regulators>

Figura 1.13. Gráfico de estabilidad y carga de regulación lineal

Los reguladores lineales no tienen la misma eficiencia energética que un regulador de voltaje conmutado, pero el regulador lineal tiene la característica que mientras más cerca esté el valor de la salida del BUCK “reductor” en comparación de la entrada, mayor será su eficiencia energética acercándose desde un 95% hasta 99% de eficiencia energética. Tiene características menores pero que también influyen en el producto final tales como bajo ruido eléctrico, bajo

costo de implementación, una respuesta rápida a las variaciones, menor radiación electromagnética. En la presente figura 1-14 se presenta la gráfica comparativa de eficiencia energética contra voltaje entrada y salida



Fuente: <https://www.digikey.com/es/articles/understanding-the-advantages-and-disadvantages-of-linear-regulators>

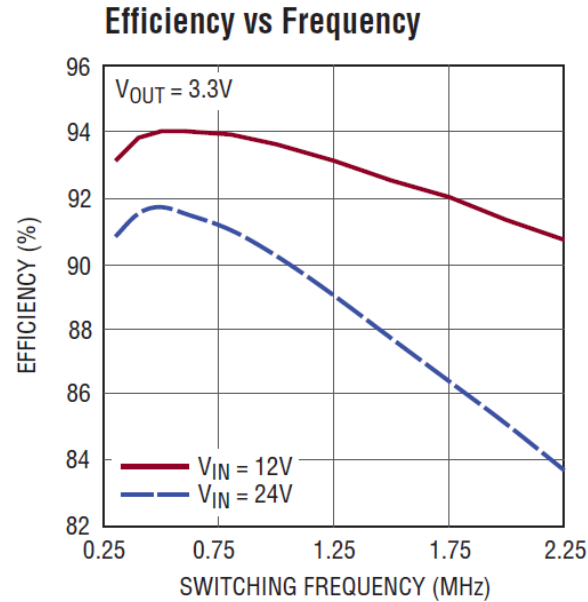
Figura 1-14. Gráfico de eficiencia energética

1.2.1.2 Desventajas del regulador de voltaje lineal.

Según lo indicado en la figura 1-14. La eficiencia energética que presenta este regulador solo es apreciable cuando se acerca al valor de voltaje de entrada, este efecto rara vez se produce ya que normalmente se suele alimentar con voltajes altos del rango 9V-24V en el caso de la tarjeta de desarrollo pingüino, para poder mitigar la carencia de eficiencia energética que se produce se agrega previamente un regulador de voltaje conmutado el cual consta de una mayor eficiencia reduciendo los valores nominales de 9V-24V a 5V permitiendo que la eficacia energética del regulador lineal aumente, cuentan adicionalmente de desventajas que afectan la operación ideal tales como una mala disipación de calor, estar limitados por bajo el límite de corriente permitido, un mayor tamaño, y mayores costos en alta potencia no apreciables en baja potencia como en este caso.

1.2.1.3 Ventajas del regulador de voltaje conmutado.

El regulador de voltaje conmutado presenta ventajas al momento de ejecución del microcontrolador, estos constan de una mayor eficiencia energética cuando mayor es el rango de diferencia entre la entrada y salido regulada del microcontrolador en comparación con el lineal. Esto es esencialmente eficaz al momento de poder distribuir los voltajes necesarios a los diferentes componentes del microcontrolador, con su uso en el microcontrolador pingüino se puede regular de 9 a 24 voltios en la figura 1-15. Se observa su eficiencia vs la frecuencia



Fuente: <https://electronics.stackexchange.com/questions/38189/why-is-a-lower-switching-frequency-more-efficient>

Figura 1-15. Gráfico eficiencia/frecuencia regulación conmutada

Otra característica la cual es una ventaja en su uso es tener una menor disipación de energía en calor, esto se produce gracias a lo indicado en la figura 1-15. la cual muestra que a menor frecuencia la eficiencia energética aumenta, en consecuencia, la proporción de energía que es convertida en calor es menor, cuenta además con un mayor rango de voltaje de regulación en comparación del lineal es capaz de regular voltajes más altos sin sufrir una pérdida de eficiencia significativa.

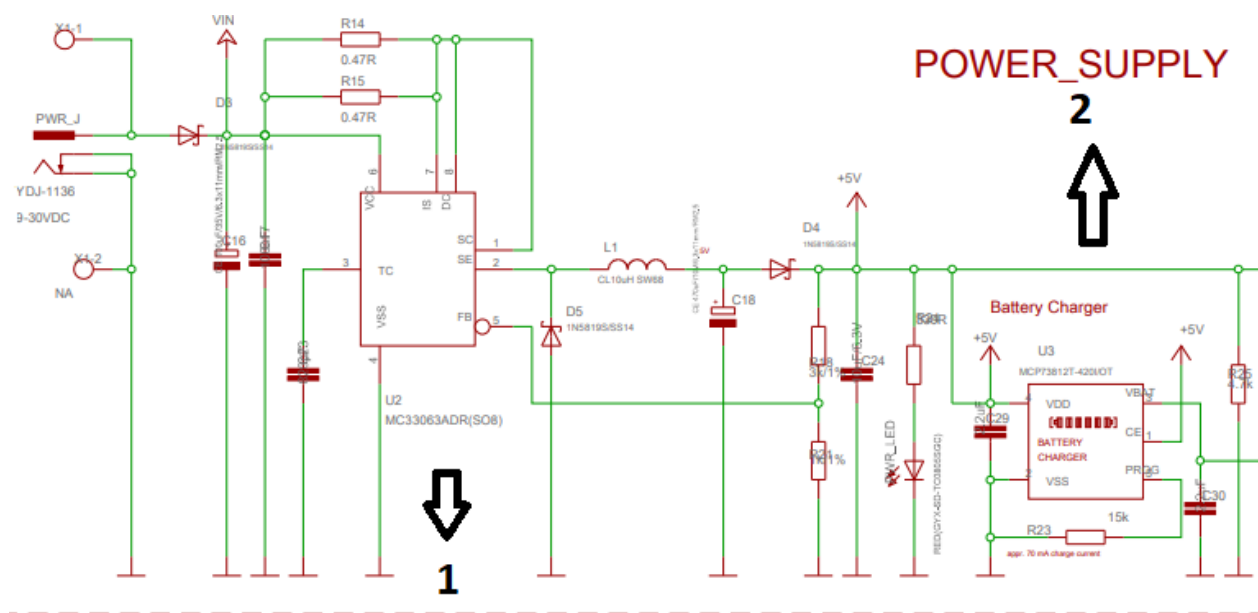
Cabe mencionar que se pueden adjudicar como ventajas, el hecho que cuentan con una menor caída de voltaje en comparación de los reguladores lineales, mayor capacidad de corriente la cual va en concordancia a los mayores niveles de voltaje que estos son capaces de soportar, y tamaños más compactados, la proporción de tamaño que utilizan los reguladores de voltaje conmutados es menor a comparación de los reguladores de voltaje lineales por lo general.

1.2.1.4 Desventajas del regulador de voltaje conmutado.

El regulador de voltaje conmutado presenta una serie de desventajas las cuales empeoran la eficiencia de la regulación.

El regulador de voltaje presenta una mayor susceptibilidad al ruido. Los reguladores conmutados son más susceptibles a generar ruido en la señal de salida debido a la constante conmutación que ejerce al regular el voltaje, esto puede interferir en aplicaciones de ruido sensibles y requerirá un filtrado o supresión. También estos conmutadores son susceptibles a generar interferencias electromagnéticas EMI “interferencia electro magnética” .

En conclusión, el diseño de las tarjetas generalmente implementa estos dos reguladores de voltaje, lineal y conmutado gracias a sus características. Utilizan el regulador conmutado para poder regular señales de un mayor voltaje 24V a 5V con una mayor eficiencia energética para ocuparlo como señal de bajo nivel para algunos componentes, después son almacenados en una batería para mantener el nivel y regularse de nuevo en una nominación de 3.3V como se puede observar en la figura 1-16. La cual es usada para alimentar el procesador y otros componentes; el uso de doble reguladores permite un aumento de la eficiencia energética una de alto rango y otra de bajo rango para evitar que la energía se disipe en calor. Se puede observar el regulador conmutado “1” y la batería “2” para mayor comprensión en la figura 1-16.

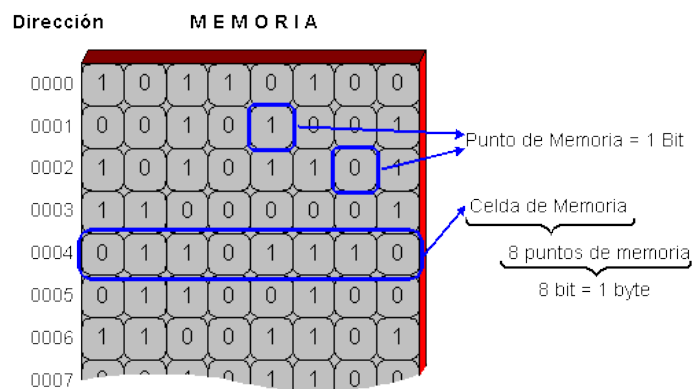


Fuente: <https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO/open-source-hardware>

Figura 1-16 Regulador conmutado y batería “Pinguino”

1.2.2 ¿Qué es una memoria Flash, PROM y EPROM?

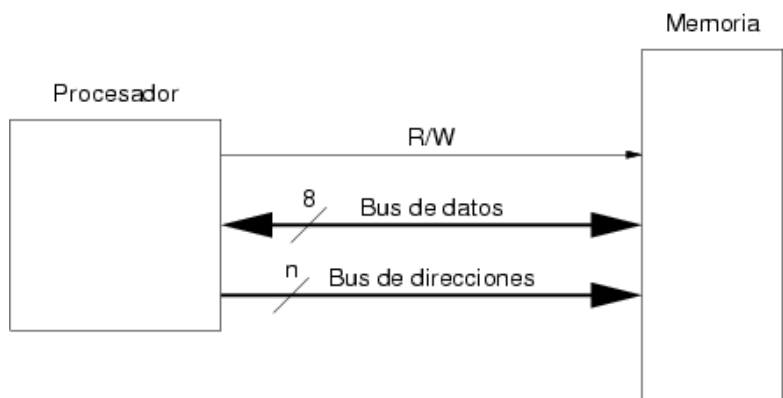
Las memorias son dispositivos electrónicos las cuales mediante estados eléctricos almacenan información, esto es extremadamente útil para usuarios de las tarjetas pinguino porque la ausencia de almacenamiento imposibilitaría el funcionamiento de estos, las memorias almacenan datos en forma de bits los cuales son almacenados en celdas de 8 bits o Byte, 16 bits llamadas WORD, 32 bits llamadas DOUBLE WORD. además, esta cuenta con una dirección que cumple la función de rastreador, para poder identificar en cual se almacena la información necesaria en la figura 1-17. Se observa la composición de una memoria de 1 Byte por celda. Existen de mayor tamaño siendo de doble palabra, etc.



Fuente: <https://josean2013.wordpress.com/2015/08/31/estructura-de-la-memoria/>

Figura 1-17. Almacenamientos de datos en una memoria

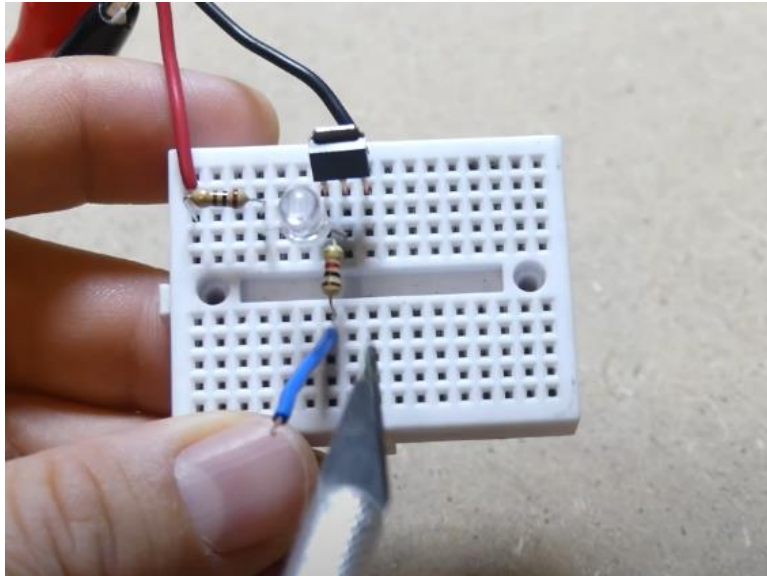
En mención a lo relacionado con las tarjetas “Pingüino” esta tarjeta tiene un bus de transferencia de datos de 32bits, estos 32 bits se refieren al ancho de datos que la memoria que puede acceder a la vez siendo un número aproximado de 4 millones, la comunicación la realiza con la EPROM o flash, respectivamente. En la figura 1-18 se puede apreciar cómo se comunica un procesador con una memoria en este caso 8 bits en el bus de datos.



Fuente: https://lc.fie.umich.mx/~rochoa/Manuales/ENSAMBLADOR/ARQUITECTURA_Sium/DRM.html

Figura 1-18. transferencia de datos de almacenamiento

El funcionamiento físico de una memoria digital se puede representar mediante un MOSFET, 2 resistores y un led. El MOSFET tiene la característica de poder almacenar información cuando se le suministra energía al MOSFET, siendo ON- OFF como estados representados eléctricamente. Los microcontroladores cuentan con millones de Transistores MOSFET en su interior, el tamaño aproximado de cada uno de estos MOSFET es del rango de nanómetros en los procesadores actuales, permitiendo un mayor almacenamiento o una mayor transferencia de datos a un menor gasto eléctrico, por la razón que mientras más pequeño es un MOSFET menor es su consumo, pero más sensible al ruido es. En la figura 1-19. Se observa una representación para visualizar como está compuesto la memoria.



Fuente: circuito realizado por el alumno

Figura 1-19. Circuito representativo de una memoria

Ahora comprendiendo el funcionamiento de una memoria, el funcionamiento de una memoria Flash se asemeja a lo explicado anteriormente las características que le son inherentes a este tipo de memoria se exponen a continuación.

En una memoria flash su información puede ser borrada eléctricamente sin necesidad de una intervención física por el usuario, esto da a lugar que la memoria flash sea más práctica en su uso diario en la constante programación y reprogramación, esto es vital en los controladores incluyendo el Pingüino.

Cuenta con una vida útil menor a la memoria EPROM. “memoria programable y borrrable de solo lectura”, la memoria flash, aunque larga, tiene una vida útil mucho menor siendo necesario tener que reemplazar el microcontrolador si viene incorporada o reemplazar la memoria SD si se encuentra separada al circuito principal, como se observó previamente la tarjeta pingüino tiene la capacidad de reemplazar la memoria SD. Cuenta por último con tiempo de accesos más rápidos y un tamaño menor en comparación de las memorias PROM “memoria programable de solo lectura” y EPROM. En la figura 1-20 se pudo observar una memoria flash tipo microSD utilizada en la tarjeta Pingüino.



Fuente: <https://www.solotodo.cl/products/87738-sandisk-ultra-microsdhc-32-gb-uhs-i-class-10-speed-class-1-sdsqunr-032g-cn3ma>

Figura 1-20. Memoria flash tipo microSD

Las memorias PROM por su abreviación Programmable Read Only Memory es un tipo de memoria la cual solo puede almacenar datos una sola vez, este tipo de memoria era utilizado en microcontroladores para datos que solo necesitaban programarse una sola vez en la vida, este tipo de memoria se dejó de utilizar ya que inhabilitaba mejoras que se le podrían hacer al controlador inutilizándolo, para poder realizar una mejora era necesario remplazar el procesador completo. Como se muestra en la figura 1-21 este tipo de memoria de solo lectura era necesario remplazar la memoria completa para actualizar el microcontrolador.



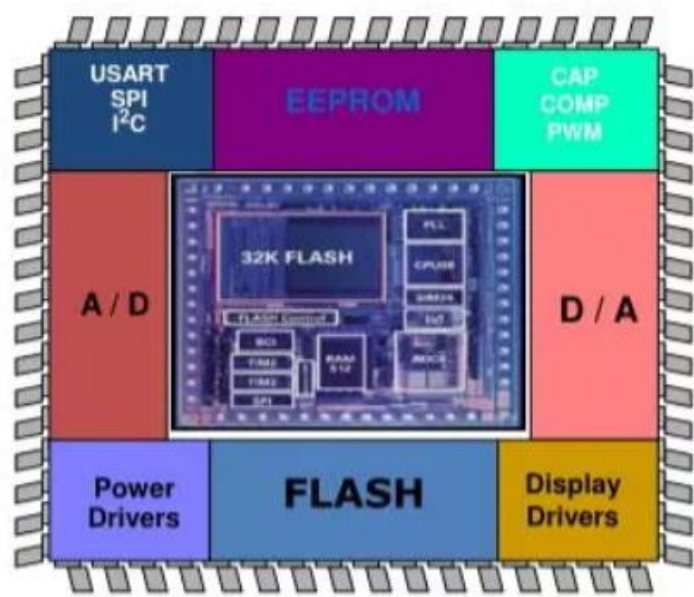
Fuente: <https://es.scribd.com/document/590479932/Que-es-la-memoria-PROM>

Figura 1-21. Memoria PROM

Las memorias EPROM y EEPROM son tipos de memorias más seguras y duraderas que las flash, estas memorias permiten almacenar datos esenciales una vez y dependiendo el tipo de memoria tener un borrado manual “exposición de chip a rayos UV” o eléctrico, este tipo de memoria es utilizada en ocasiones puntuales para almacenar información que no es necesaria reprogramar constantemente, se utiliza este tipo de memorias por varias razones en las que está la seguridad de la memoria y al ser EPROM solo borrarse mediante luz UV permiten seguridad en los datos, también tiene un proceso de programación sencillo y directo, y una fiabilidad a largo plazo gracias a la duración cuando se le compara a la memoria flash.

Las memorias EPROM y EEPROM “memoria programable y borrrable eléctrico de solo lectura” a las memorias PROM son parecidas en sus características, pero lo que la diferencia de su homóloga en caso de la EPROM es el borrado por luz UV que permiten reescribir la memoria, La EEPROM es parecida también, pero con la diferencia que tiene borrado eléctrico de memoria, esto le permite tener las ventajas de la memoria PROM con la facilidad de borrado de una memoria flash. Las memorias más usadas en los microcontroladores modernos incluyendo al Pingüino son las memorias EEPROM y la memoria flash, son usadas por la duración y seguridad “EEPROM” y la capacidad y velocidad “FLASH”.

En caso de Arduino pingüino esta tarjeta tiene adherido una memoria flash de 256KB, una memoria SRAM de 32KB “tipo de memoria única que permiten una comunicación rápida entre el procesador y la información”, agrega una memoria EPROM de 8KB. En la Figura 1-22 se puede apreciar la estructura de un procesador de un microcontrolador.



Fuente: <https://docplayer.es/15506145-Departamento-electronica-pic-tema-1-introduccion-a-los-microcontroladores.html>

Figura 1-22. Procesador de un microcontrolador

1.2.3 ¿Cuáles son Las características del procesador “Pingüino”?

El procesador Pic32MX440F256H es la base de la tarjeta de desarrollo, se encarga de realizar todas las tareas de control, automatización y ejecución que necesite el usuario mediante programación. Sus características las cuales lo diferencian de su competencia se representarán en la Tabla 1-1.

Según las características indicadas en la tabla 1-1 el procesador PIC32MX440F256H presenta características necesarias en la enseñanza de microcontroladores tales como comunicación UART, I2C, conversores análogos digitales.

Tabla 1-1 Características del CPU del microcontrolador pingüino

Característica	Valor
Voltaje de salida	3.3V-5V
Bits de transferencia	32 bits
Comunicación UART	Si, con 2 conectores RS232-485
Max corriente de salida por pin	18mA
Frecuencia max. reloj	80Mhz
Memoria flash	512KB
Memoria SRAM	32KB
Memoria EPROM	8KB
Temperatura de trabajo	-25°C hasta 85°C
Voltaje de alimentación	3.3V
Salidas digitales	14
Salidas análogas	6
Max voltaje análogo input	5.5V
ADC pines	2
Muestreo de conversión de señal Analog	500 KSPS
Pin de compatibilidad de módulos Pic24	1
Compatibilidad con USB 2.0	Si
Compatibilidad Maestro Esclavo	PMP de 8bits y 16bits de address
Max. Corriente de todos los conectores	185ma
Comunicación I2C	2 conectores
Conectores interruptores	5 conectores
Máximo valor de voltaje digital input	5.5V
Comunicación SPI conector	2 conectores

Fuente: Tabla creada por el alumno

1.3 **PARTES ÚNICAS DE LA TARJETA DE DESARROLLO “PINGÜINO”.**

La tarjeta de desarrollo Pingüino presenta características únicas que la diferencian de su competencia, estas características facilitan el manejo de esta tarjeta y le proporciona una mayor flexibilidad en la aplicación industrial. Las características que se le puede adjudicar se presentan a continuación.

1.3.1 **Mayor Alimentación en comparación de Arduino, Esp32.**

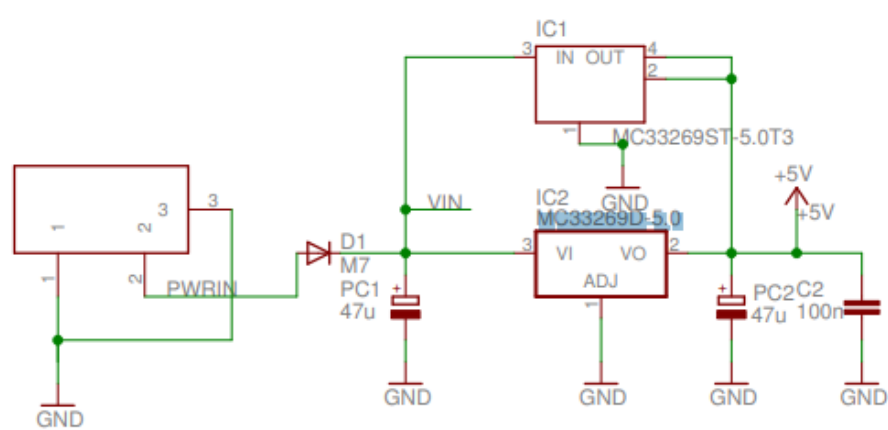
La tarjeta de desarrollo Arduino UNO consta de un regulador de voltaje MC33269D-5.0 que se pude apreciar en la figura 1-24. Este es un regulador de salida de 5V, los rangos de voltaje recomendado de trabajo del microchip ATMEGA328 representado en la figura 1-23 de la serie Arduino son de 3.3V-5.5V, este rango es ideal al regulador de voltaje MC33269D-5.0, pero este regulador de voltaje cuenta con la desventaja que al ser alimentado con rangos de voltajes superiores a 12V, el regulador de voltaje aumentará su temperatura dañando la placa de la tarjeta de desarrollo o dañándose el mismo regulador de voltaje, esto se produce debido a que el regulador de voltaje en la tarjeta microcontroladora no cuenta con un disipador térmico, sin

contar también que el regulador de voltaje el cual emplea la tarjeta de desarrollo Arduino UNO es un regulador de voltaje lineal, estos reguladores de voltaje se caracterizan en la razón que mientras más grande la diferencia del voltaje de alimentación con el de regulación, menor es la eficiencia y mayor el calor.



Fuente: <https://www.mouser.cl/new/microchip/atmelatmega328/>

Figura 1-23. Microchip ATMEGA328



Fuente: <https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf>

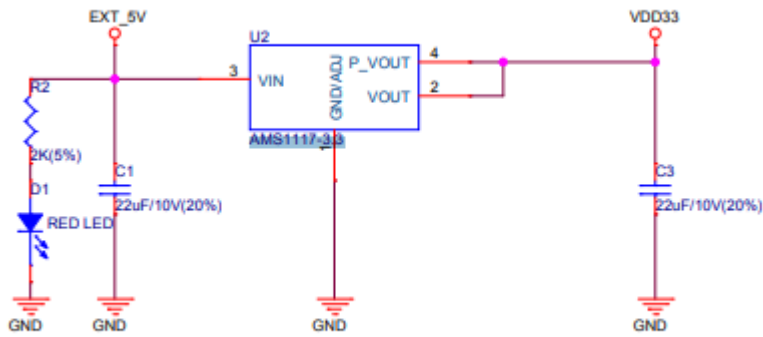
Figura 1-24. Regulador de voltaje de la tarjeta microcontroladora Arduino UNO

Mientras tanto la tarjeta de desarrollo ESP32 cuenta con un MICROCHIP ESP32-D2WD el cual se observa en la figura 1-25. Este microchip requiere un rango de voltaje de trabajo de 2.3V-3.3V. El regulador de voltaje que emplea la tarjeta microcontroladora es el AMS1117-3.3 que se puede observar en la figura 1-26. Este regulador de voltaje es un regulador de voltaje lineal con salida de voltaje de 3.3V, como se mencionó en la anterior tarjeta microcontroladora Arduino uno este tipo de componente es incompatible con altos rangos de voltaje, pero esta elección con esta tarjeta microcontroladora es necesaria ya que el objetivo de esta tarjeta microcontroladora es ser lo más reducida en tamaño posible, por ende los reguladores de voltaje lineales al requerir menos componentes, y ser de un tamaño pequeño termina siendo el ideal para esta tarjeta de desarrollo.



Fuente: <https://www.mouser.cl/ProductDetail/Espressif-Systems/ESP32-D2WD?qs=chTDxNqvsynYfhF3um46Mw%3D%3D>

Figura 1-25. microchip esp32-d2Wd



Fuente: https://dl.espressif.com/dl/schematics/esp32_devkitc_v4-sch.pdf

Figura 1-26. Regulador de voltaje lineal tarjeta desarrolladora ESP32

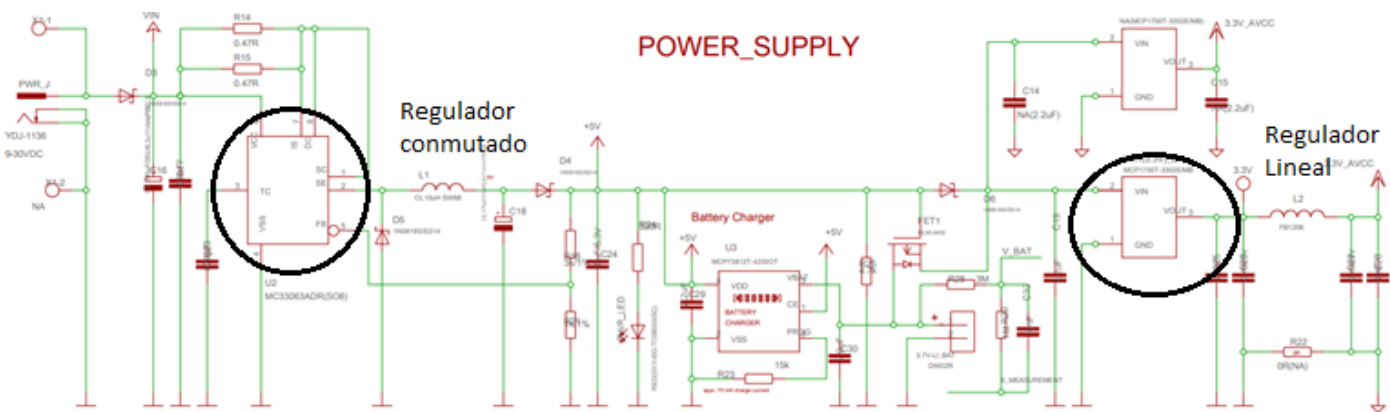
Finalmente, la tarjeta de desarrollo Pingüino OTG es una gama de tarjetas diseñadas por OLIMEX con arquitectura PIC32, como se indicó en la tabla 1-1. Esta tarjeta cuenta con un microchip PIC32MX440F256H como se observa en la figura 1-27. Este microchip requiere una alimentación de 3.3V, pero integrada a esta tarjeta tiene una característica que la diferencia de su competencia, esta tarjeta no cuenta con solo un regulador de voltaje sino dos reguladores de voltajes, esto le permite tener la característica de ser alimentada con un rango de voltaje mayor siendo este de 9V-24V su funcionamiento, recordando lo expuesto en el tema 1.2.1 en resumen es tener un regulador de voltaje conmutado que reduzca el voltaje a 5 voltios y se regule nuevamente con un regulador lineal. El regulador de voltaje conmutado MC33063ADR(SO8) como se observa en la figura 1-28. Presenta una mayor eficiencia energética en rangos de voltaje mayores, y el regulador de voltaje lineal MCP1700T que se observa en la figura 1-28.

El regulador lineal tiene una alimentación de 5V este siendo la previamente regulada por el regulador conmutado la eficiencia es mayor en la regulación lineal de salida 3,3V. La combinación de estos dos reguladores de voltajes permite una integración de esta tarjeta en circuitos de control de 24V sin la necesidad de un reductor externo. En la figura 1-28. Se observan dentro de círculos los reguladores conmutados y lineal, respectivamente.



Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-27 Microchip PIC32MX440F256H



Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-28. Regulador de voltaje conmutado y lineal Pingüino OTG

1.3.2 Mayor variedad de conectores.

La tarjeta de desarrollo Pingüino OTG presenta una serie de conectores que le permiten facilitar su uso, el primero es un conector USB 2.0 mini tipo B como se puede observar en la figura 1-30. Este conector es un remplazo del conector USB 2.0 tipo B del Arduino uno y el conector micro USB 2.0 del ESP32. A grandes rasgos solo varía la duración y tolerancia de corriente entre estos, la diferencia de conectores de alimentación y traspaso de datos también le otorga la posibilidad de accesibilidad de remplazo del cable transmisor, donde el ESP32 con su conector micro USB permite más fácil remplazar el cable transmisor, después la tarjeta pingüino OTG

con el conector mini USB, y por último el conector USB tipo B de Arduino siendo este el más difícil de obtener.

Otro conector que presenta la tarjeta de desarrollo Pingüino es un conector UEXT este es un tipo de conector de 10 pines, este conector está estandarizado y permite protocolos de comunicación UART, I2C, SPI. La estandarización le permite la compatibilidad con módulos de radio frecuencia, ZIGBEE, GPS, GSM. Olimex ofrece una variedad de módulos compatibles con este tipo de conector para facilitar las conexiones entre el módulo y la tarjeta microcontroladora.

La tarjeta de desarrollo Pingüino cuenta con un conector ICSP el cual es un conector enfocado en la depuración y la programación de firmware sin tener que retirar el microcontrolador. Este conector facilita la programación de una tarjeta de desarrollo ya instalada mediante otra tarjeta microcontroladora sin necesidad de retirarla. En la figura 1-29. Se puede observar el conector negro UEXT y ICSP blanco.



Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-29. Conector UEXT y ICSP



Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-30 Conector Mini USB

1.3.3 Pines y conectores disponibles para su uso y para qué sirven.

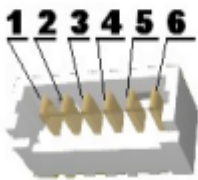
Los conectores y pines de la tarjeta de desarrollo Pingüino presentan una serie de características que ayudarán a entender su funcionamiento con las tablas presentadas a continuación, se entenderá que sus conexiones abarcan todos los pines y para qué es necesario cada pin. Se presentan los siguientes ejemplos con una breve explicación.

Se presenta a continuación en la figura 1-31 el conector ICSP encargado en poder programar el microcontrolador sin necesidad de retirarlo del circuito mediante otra tarjeta. Además, se observa en la tabla 1-2. La función de cada pin.

Tabla 1-2. Pines del conector ICSP

Pin	nombre
1	Reset
2	+3.3V
3	GND
4	PGED2
5	PGEc2
6	No conectado

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

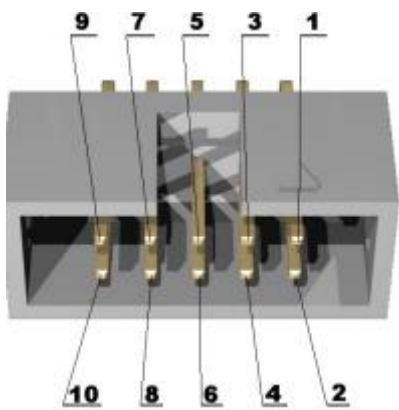
Figura 1-31. Conector indicativo ICSP

Se presenta en la figura 1-32. El conector UEXT encargado de realizar las comunicaciones de la tarjeta microcontroladora Pingüino con los módulos compatibles con protocolos I2C, UART, SPI. Su uso permite una mayor comodidad al momento de implementar módulos compatibles con este tipo de puerto. Además, se observa en la tabla 1-3. La función de cada pin.

Tabla 1-3. Pines del conector UEXT

Pin	Nombre
1	+3.3v
2	GND
3	TX2
4	RX2
5	SCL1
6	SDA1
7	MISO
8	MOSI
9	SCK/led1
10	UEXT

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

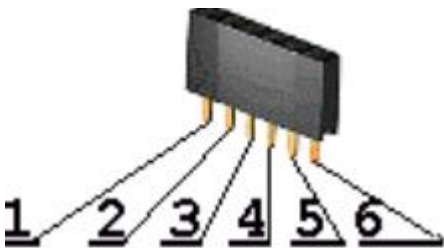
Figura 1-32. Conector indicativo UEXT

Se observa en la figura 1-33 los pines del CON1- POWER encargado en la alimentación del microcontrolador y salidas de voltajes para alimentar dispositivos, este conector presenta una salida de 3.3V la cual deriva del regulador de voltaje lineal que tiene incorporado y una salida de 5V del regulador de voltaje conmutado. Los pines VIN y GND pueden ser utilizados para suministrar voltajes a la tarjeta de desarrollo Pingüino. Además, su respectiva tabla 1-4 muestra sus respectivas funciones.

Tabla 1-4. Pines del conector Con1-Power

Pin	nombre
1	Reset
2	+3.3V
3	+5V
4	GND
5	GND
6	VIN

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-33 Conector indicativo Con1-Power

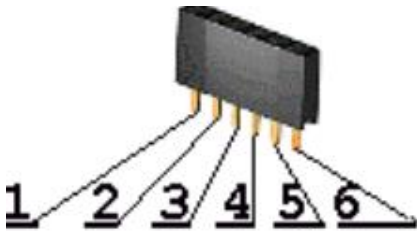
Se presenta a continuación en la figura 1-34 los pines CON2-ANALOG encargados de controlar las variables análogas y comunicación I2C. La señal Análoga tiene un rango de voltaje con el cual trabaja, superar este rango provocará que la entrada se queme, este rango de voltaje es de 0V-5.5V, este rango tiene una lectura de valor numérico donde 0 representa 0V hasta 1023 representando 5V, este límite permite transformar señales análogas en información digital, esta

aplicación es altamente necesaria al momento de trabajar con sensores. Además, en la tabla 1-5 presenta las entradas análogas “A” e I2C.

Tabla 1-5 Pines del conector Con2-Analog

Pin	nombre
1	A0
2	A1
3	A2
4	A3
5	A4-SDA1
6	A5-SCL1

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

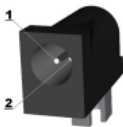
Figura 1-34 Conector indicativo Con2-Analog

Se presenta a continuación en la figura 1-35. El conector PWR JACK es un conector encargado de alimentar la tarjeta microcontroladora pingüino, este conector facilita la alimentación de la tarjeta agregándole una opción más para suministrarle voltaje. En la tabla 1-6 se representan sus respectivos terminales.

Tabla 1-6. Pines del conector PWR JACK

Pin	nombre
1	POWER INPUT
2	GND

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

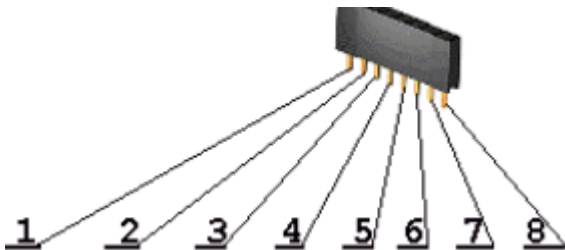
Figura 1-35 Conector indicativo PWR

Se presenta a continuación en la figura 1-36 los pines CON4-DIGITAL estos pines también llamados GPIO “GENERAL PIN INPUT OUTPUT” son conectores programables destinados para ejecutar el control del programa, son capaces de ser programados para actuar como entradas digitales tomando el valor de 0-1 o tomar el valor de salida siendo ON-OFF, el voltaje de trabajo de estos pines es de 3.3V-5V. Es posible también utilizar estos pines para realizar comunicación serial, esta comunicación utiliza el protocolo UART. Además, en la tabla 1-7. Se presenta la utilidad de cada uno de estos pines.

Tabla 1-7. Pines del conector Con4-Digital

Pin	Nombre
1	D0-RXD1
2	D1-TXD1
3	D2-BUT
4	D3
5	D4
6	D5
7	D6
8	D7

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

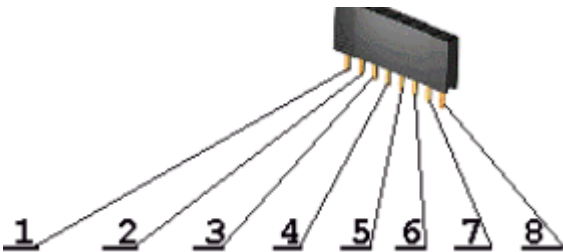
Figura 1-36 Conector indicativo Con5-Digital

Se presentan en la figura 1-37 los pines CON5-DIGITAL como se mencionó con el conector anterior estos pines son denominados GPIO, son usados para el diseño de circuitos volviéndolos entradas o salidas. Otra característica que presentan estos pines es la posibilidad de usar el pin 1 para realizar comunicación con la memoria SD siendo capaz de leer y escribir en la microSD. Tiene también conexión SPI en los pines MISO y MOSI, este tipo de comunicación permite el enlace entre un maestro y un esclavo. Además, en la tabla 1-8 presenta todas sus funciones.

Tabla 1-8. Pines del conector Con5-Digital

Pin	Nombre
1	D8-MMC_
2	D9
3	D10
4	D11-MOSI
5	D12-MISO
6	D13-LED1
7	GND
8	AREF

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-37 conector indicativo Con5-Digital

Se presenta en la figura 1-38 el conector USB-OTG encargado de realizar la comunicación entre la programación y la tarjeta de desarrollo. Este conector es un mini USB tipo B y suministra un voltaje de 5V a la tarjeta de desarrollo. Además, en la tabla 1-9 se presenta todas sus funciones.

Tabla 1-9. Pines del conector USB-OTG

Pin	nombre
1	+5v_
2	D-
3	D+
4	USB-ID
5	GND

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

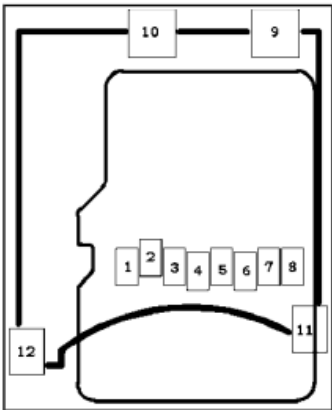
Figura 1-38. Conector indicativo USB-OTG

Se presenta en la figura 1-39 el slot SD/MMC que permite incluir una memoria en la tarjeta Pingüino, permitiendo aplicaciones más complejas como servidores de datos, almacenamiento de imágenes, etc. La localización de este slot de memoria se puede apreciar en la parte trasera del microcontrolador, este slot de memoria ofrece un acceso directo para trabajar con ella, otras tarjetas de desarrollo requerirían un módulo microSD para trabajar con ellas. Además, se presenta en la tabla 1-10 las funciones del slot SD/MMC.

Tabla 1-10 Terminales de conexión microSD

	Nombre
1	MCIDAT2
2	D8_MMC_#SS
3	D11-MOSI
4	+3.3V
5	D13-SCK/LED1
6	GND
7	D12-MISO
8	MCIDAT1
9	N/A
10	N/A
11	N/A
12	N/A

Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf



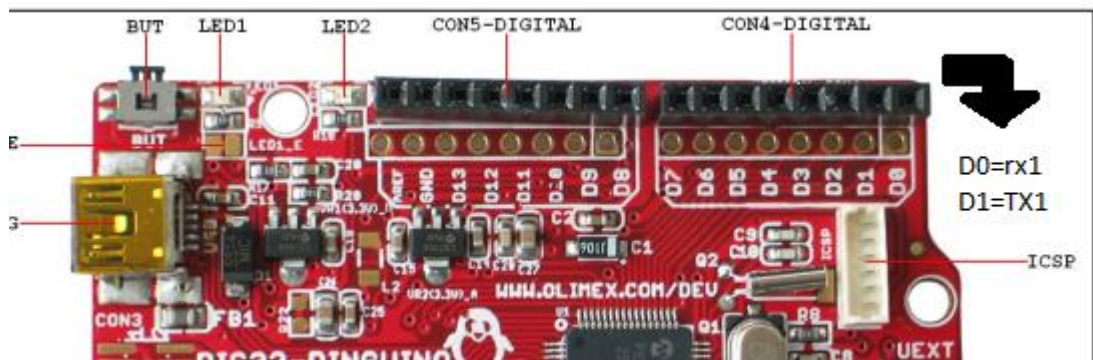
Fuente:https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-39 Conector indicativo MicroSD

1.4 **COMUNICACIONES Y PROTOCOLOS DE COMUNICACIÓN DE LA TARJETA DE DESARROLLO PINGÜINO**

La tarjeta de desarrollo Pingüino tiene una serie de comunicaciones digitales, estas comunicaciones digitales son usadas para la aplicación y comunicación de módulos de diferente utilidad, estas comunicaciones digitales utilizan protocolos para realizar la comunicación de forma más eficiente, estos se separan en diferentes tipos como se expone a continuación se separan por comunicación síncrona y asíncrona.

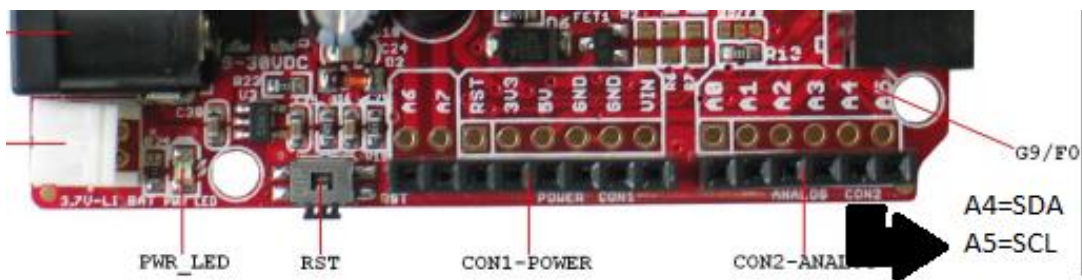
La comunicación Serial asíncrona “UART”, esta comunicación digital tiene dos terminales, siendo el conector RX y TX que utilizan el protocolo UART, estos se pueden apreciar en los terminales D0 y D1, este protocolo esta estandarizado y permite el uso de estos dos terminales para facilitar las comunicaciones con sus dispositivos. En la figura 1-40 se modificó la imagen para tener una mejor comprensión de los terminales UART.



Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-40. Visión física de conector serial

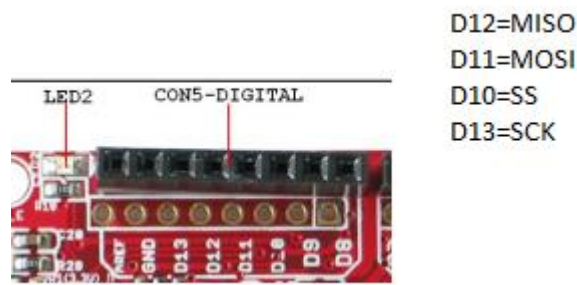
La comunicación serial síncrono “I2C” se caracteriza por tener dos terminales, siendo el conector SDA y SCL, estos terminales ocupan el protocolo de comunicación I2C, este protocolo está estandarizado para el uso de módulos en la aplicación de circuitos, los módulos más comunes que usan esta comunicación son las pantallas LCD, sensores de temperatura, humedad, etc. En la figura 1-41 se editó la imagen para una mejor comprensión de donde se localizan los terminales.



Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-41. Visión física de conector SCL

La comunicación serial síncrona “MOSI,MISO”, esta comunicación digital se caracteriza por tener 2 terminales, siendo el conector MOSI y MISO, D12 y D11 respectivamente, esta comunicación serial ocupa el protocolo de comunicación SPI, este protocolo es utilizado para estandarizar el uso de módulos en la aplicación de circuitos, esta comunicación serial está enfocada en el enlace de un maestro con un esclavo, esta comunicación serial requiere dos conectores extras siendo el conector SCK “Serial CLOCK” y SS “Slave SELECT” para poder conectar un microcontrolador como maestro y varios dispositivos como esclavos. Como se puede observar en la figura 1-42 se editó para tener una mayor comprensión del esquema.

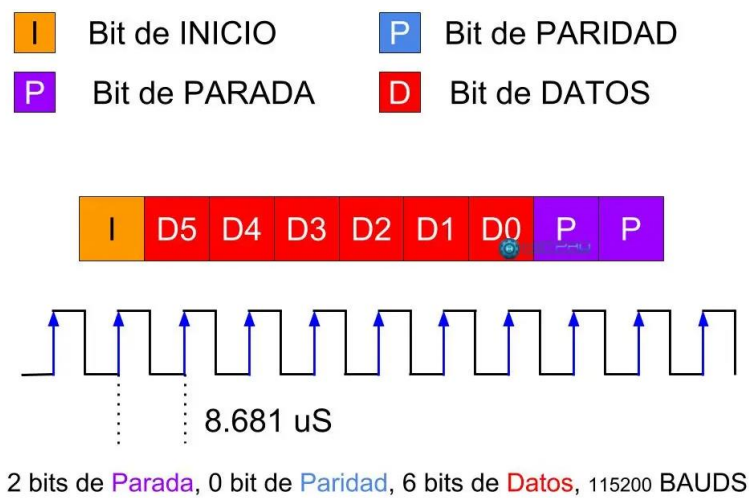


Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf

Figura 1-42. Visión física de conector SPI

1.4.1 Comunicación serial UART

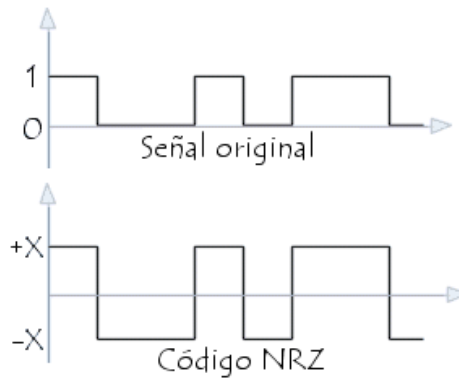
Esta comunicación serial se caracteriza por ser una señal asíncrona, una señal asíncrona es un tipo de señal la cual en el traspaso de datos entre el dispositivo y la tarjeta microcontroladora, en la señal a transmitir por ejemplo 1100011 no se identifica cual bit es cual mediante un reloj, las señales asíncronas ocupan un bit de inicio para poder identificar el inicio de una señal y ocupan un método diferente para poder identificar cual bits es cual, estos utilizan un reconocimiento de tiempo definido en Baudios, normalmente se utiliza el rango de 9600 Baudios y el cálculo para saber cuánto tiempo tiene cada pin es de $1/9600$ Baudios resultando un total 104us para cada lectura, en la práctica se le agregan algunos microsegundo más para poder leer en zonas más estables del cambio de bit. En la figura 1-43 se muestra como ocurre el proceso de reconocimiento mediante baudios/segundos.



Fuente: <https://hetpro-store.com/TUTORIALES/puerto-serial/>

Figura 1-43. Lectura de datos asíncrona

La señal de comunicación Serial con protocolo UART utiliza un tipo de codificación especial para poder transmitir sus datos, esta codificación es conocida como NRZ “NO RETURN ZERO” este tipo de codificación identifica como un valor digital 1 un nivel de voltaje alto e identifica un valor digital 0 un nivel de voltaje negativo, este método es usado para evitar interferencias que se pueda generar en la transmisión ayudando en la estabilidad de esta. En la figura 1-44. Se puede apreciar cómo se ejecuta el código NRZ.



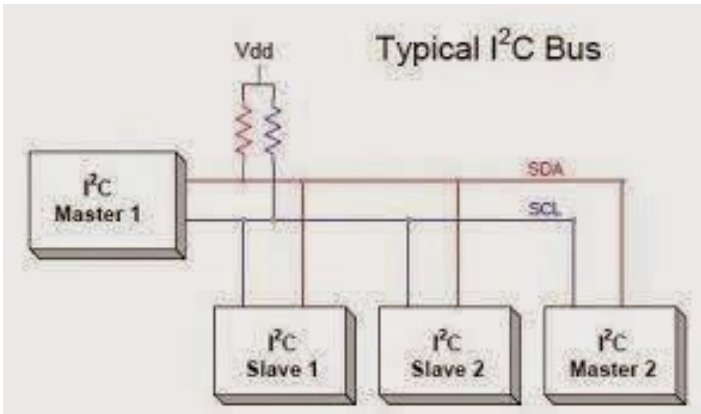
Fuente: https://www.researchgate.net/figure/Figura-38-Codificacion-NRZ-bipolar_fig23_45412587

Figura 1-44. Codificación NRZ

Por último, la comunicación serial asíncrona trabaja un tipo de protocolo denominado UART, este protocolo es usado como un estándar en este tipo de señal asíncronas, el protocolo define el nombre de los terminales siendo, RX y TX, define uso de baudios para controlar una señal, define el bit de inicio para leer una señal, y se caracteriza por ser una señal full dúplex. Las señales full dúplex son un tipo de señal que tiene la capacidad de enviar y recibir datos al mismo tiempo, un ejemplo sería la comunicación entre una tarjeta de desarrollo o un monitor serial.

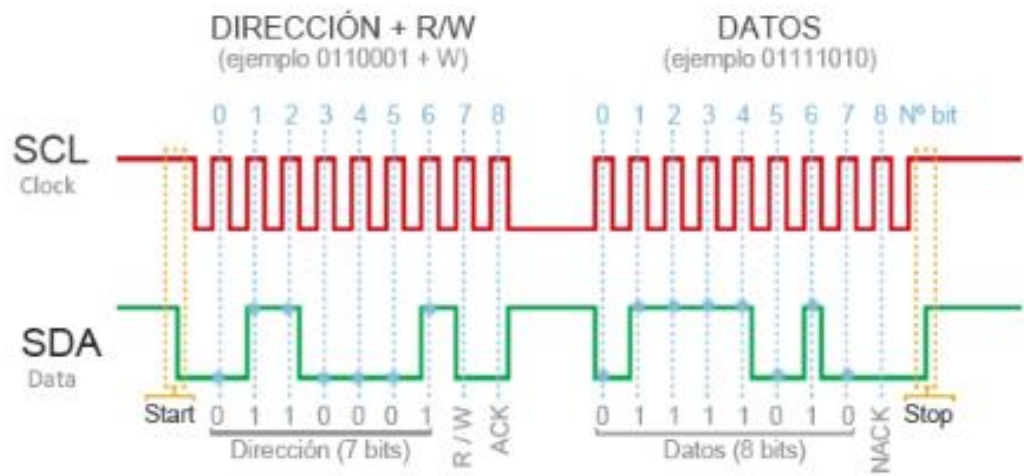
1.4.2 Comunicación serial I2C

Esta comunicación serial se caracteriza por ser una señal síncrona, una señal síncrona es un tipo de señal la cual al momento de transmitir datos usa un CLOCK para identificar los datos que se están transmitiendo, los dos cables que utiliza se separan en SCL siendo este el reloj de sincronización y SDA siendo este el terminal de transmisión de datos, por ejemplo si una señal transmite un valor de datos de 1100011 esta se transmitirá por el terminal SDA y mientras tanto se recibirá la señal sincrónica del reloj por el terminal SCL, la transmisión de datos I2C se caracteriza por enviar primero la dirección del dato y luego el dato, ambos siendo identificados por el clock sincronizador, tener una dirección permite tener un maestro en este caso el microcontrolador y varios esclavos estos siendo los módulos pertinentes. En la figura 1-45 se puede apreciar un esquema que muestra esquemáticamente cómo se comunica mediante I2C un maestro y tres esclavos.



Fuente: <https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>

Figura 1-45. Comunicación I2C



Fuente: <https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>

Figura 1-46 Señal SCL “clock” y SDA “datos”

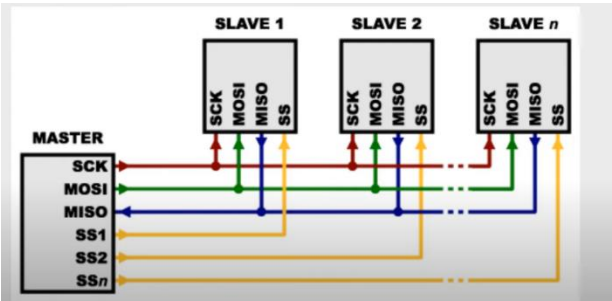
La señal de comunicación serial con protocolo I2C, utiliza un método de codificación parecido a la comunicación serial asíncrona UART vista anteriormente siendo NRZL donde los bits se identifican con un clock como se puede apreciar en la figura 1-46.

Por último, la señal serial síncrona trabaja con un protocolo denominado I2C, este protocolo estandariza todas las conexiones y transmisión de datos, estas conexiones son los conectores SDA, SCL y GND, define la frecuencia del reloj siendo esta 100Khz en velocidad estándar, y esta señal tiene la capacidad de enviar y recibir datos de los diferentes módulos que estén conectados mediante la codificación de su dirección.

1.4.3 Comunicación serial SPI

Este tipo de comunicación serial se caracteriza por ser una señal síncrona en comparación de las expuestas anteriormente, por ende, la lectura de bits se ejecuta mediante un

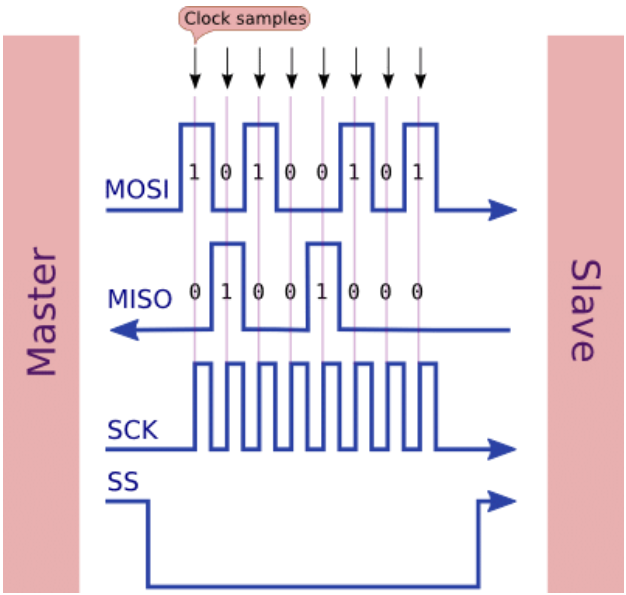
clock el cual identifica los bits, esta señal se caracteriza por tener 2 terminales llamados MOSI, MISO, SCK y SS este es un protocolo full dúplex de transmisión y cuenta con una alta velocidad de 10Mbps transferencia de datos. La comunicación de datos se ejecuta mediante el conector MOSI “Master OUT Slave IN” y MISO “Master in Slave OUT” donde respectivamente MOSI envía datos del maestro al esclavo y el otro pin envía datos del esclavo al maestro. El terminal SCK cuenta con el reloj de identificación de bits para los datos y el conector SS “Slave SELECT” es el conector que se encarga de identificar los módulos que actúan como esclavos. Su método de codificación es diferente ya que este solo envía bits de acuerdo con el reloj SCK e identifica el esclavo mediante el terminal SS “slave select”. En la figura 1-47 se puede apreciar las conexiones SS, MISO, MOSI, SCK, respectivamente.



Fuente: <https://cursos.mcielectronics.cl/2022/08/23/serial-peripheral-interface-spi/>

Figura 1-47. Conexión protocolo SPI

En la figura 1-48. Se puede observar el envío de datos desde maestro al esclavo por el canal MOSI y el envío del esclavo al maestro por MISO, mientras tanto la señal SCK es el clock identificador de bits y SS el slave select. La estandarización de estos protocolos es la SPI que designa todo lo relacionado al método de traspaso de información.

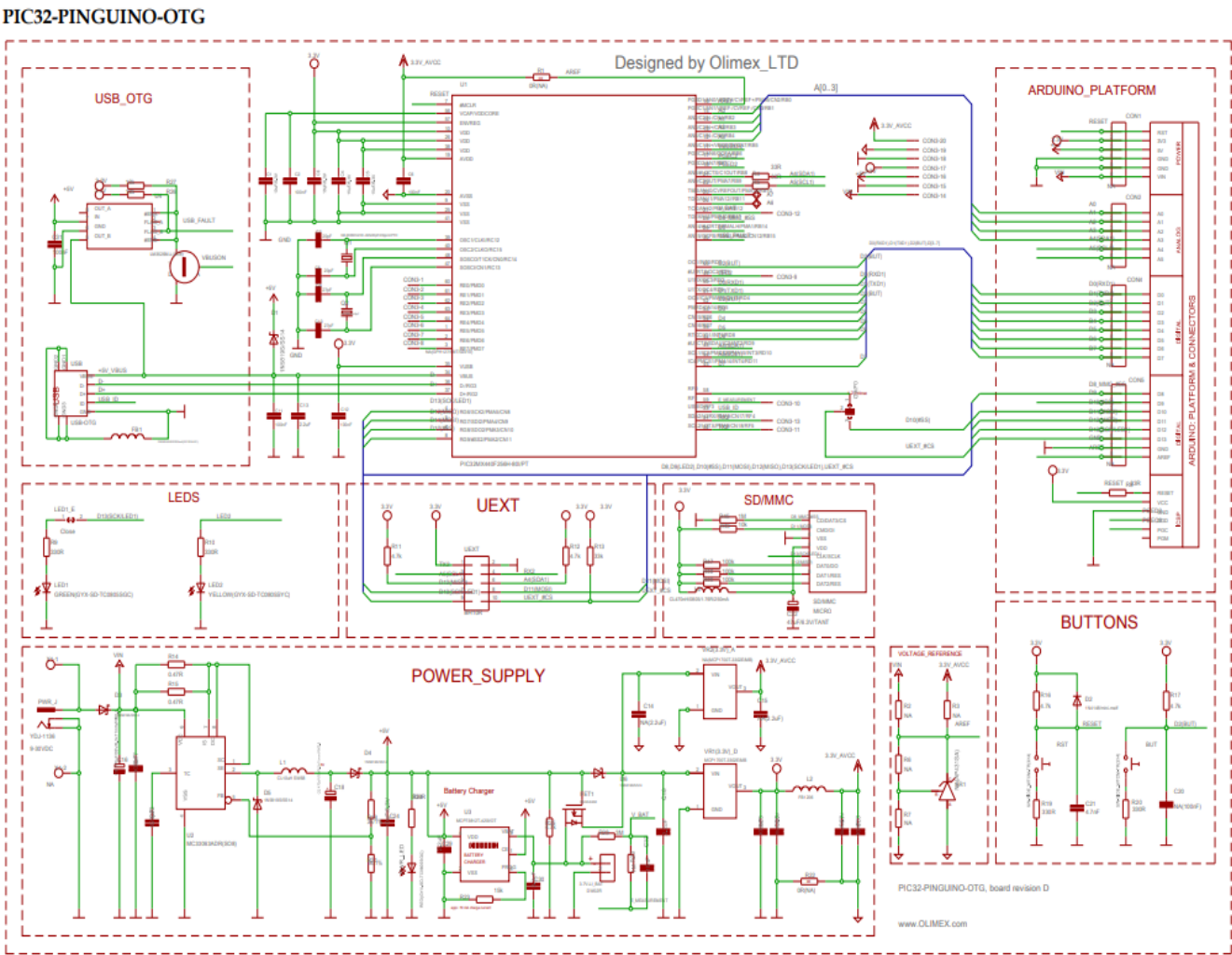


Fuente: <https://cursos.mcielectronics.cl/2022/08/23/serial-peripheral-interface-spi/>

Figura 1-48. Comunicación por terminal SPI

1.5 DIAGRAMA ELÉCTRICO

Para concluir con la revisión de la tarjeta microcontroladora se presentará un diagrama completo de la tarjeta microcontroladora Pingüino como se puede observar en la figura 1-49, este diagrama permite repasar todos sus componentes, Olimex desarrolladora de la tarjeta microcontroladora Pingüino ofrece este hardware para libre modificación y aplicación. Enfatizo energéticamente la revisión del manual online, su link está en la fuente para una mayor claridad de la imagen ya que este permite un acercamiento sin perder la calidad de la imagen.



Fuente: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pd

Figura 1-49. Diagrama completo tarjeta microcontroladora Pingüino

1.6 IMPLEMENTACIÓN EN APRENDIZAJE E INDUSTRIA

En el ámbito educacional generalmente se implementan en el estudio tarjetas de desarrollo tales como Arduino uno, ESP32 y RASPBERRY PI 3, esto se debe a que son tarjetas de bajo costo, las cuales tienen todo lo necesario para comprender de forma práctica el programa educacional. La microcontroladora PIC 32 cuenta con estas ventajas al momento del aprendizaje, es de bajo costo y contiene todas las características de lo que normalmente se enseña, esto abarca desde comunicación serial con protocolo UART con el módulo de monitores seriales, comunicación I2C para aplicar con módulos de pantallas LCD, control de señales análogas, etc. Las características que ofrece PIC32 en este caso la tarjeta Pingüino OTG con una mayor tasa de transferencia de datos que se refleja en una velocidad de reacción mayor, más un procesador de 32 bits que permite ejecutar programas más complejos deja como alternativa nada despreciable.

Otra ventaja que ofrece al momento de ejecutar la enseñanza es la capacidad de comprender que las tarjetas no están centradas en ciertas marcas específicas y tienen la capacidad de ejecutarse con alternativas más baratas o con otra arquitectura, como es en este caso con PIC 32. Generalmente la programación con microchip PIC32 se realizaba mediante ASSEMBLER lenguaje extremadamente complicado para su uso, ahora con la generación de microcontroladores PIC, en este caso Pingüino se obtiene la posibilidad de utilizar el lenguaje de programación C++, el cual mediante su ejecución cumple la función de ensamblador.

Los compiladores recomendados para la educación son:

-IDE Pingüino.

IDE Pingüino es una plataforma compiladora enfocada en Linux basada en el programa IDE ARDUINO para facilitar su uso.

-MPIDE .

MPIDE es una plataforma compiladora basada en C++ con inspiración de IDE ARDUINO enfocada en la plataforma WINDOWS

-MPLAB.

MPLAB es una plataforma compiladora basada en lenguaje C++ con su propia interfaz de programación, cuenta con características que permiten un mayor control sobre el microcontrolador a costa de una mayor dificultad de uso.

1.6.1 ¿Que son las tarjetas Arduino? Características

Arduino es una tarjeta de desarrollo de la empresa ARDUINO LLC, es una tarjeta de desarrollo enfocada en la versatilidad, aprendizaje y el desarrollo industrial. Su tarjeta microcontroladora tiene un procesador llamado ATMEGA328 siendo un procesador de 8 bits creado por la empresa ATMEL y perteneciente a la serie mega AVR.

Entre su gama de selección se puede encontrar una variedad de productos adaptados a las necesidades del circuito entre los cuales se detallan algunos a continuación.

-Arduino UNO.

Siendo la tarjeta microcontroladora más conocida, su enfoque es ser un punto intermedio entre potencia y eficiencia, presenta la capacidad de tener comunicación serial tales como UART, I2C, SPI y para su correcto funcionamiento requiere ser alimentado con un rango de 9V a 12V.

-Arduino Nano.

Esta tarjeta microcontroladora busca un enfoque más sutil ya que su objetivo es tener un tamaño diminuto, es gracias a que su microchip es el ATMEGA 32U4 el cual tiene pierde características a costa de su tamaño reducido. Este es un procesador de 8 bits y cuenta con comunicación serial tales como UART, I2C, SPI. Su rango de alimentación para el correcto funcionamiento es de 9V a 12V.

-Arduino Mega.

Arduino Mega es una de las mayores gamas de las tarjetas microcontroladoras Arduino se caracteriza por tener una mayor cantidad de pines de salida y obtener características diferentes al conocido Arduino uno, su microchip es un ATMEGA 2560 y cuenta con comunicaciones seriales tales como UART, I2C, SPI. Su rango de alimentación para el correcto funcionamiento es de 9V a 12V. En la Tabla 1-11. Se presentarán las características del Arduino UNO, la más común de estas tres tarjetas microcontroladoras

Tabla 1-11. Características de la tarjeta microcontroladora Arduino UNO

DATO	ARDUINO UNO
Bits de Transferencia	8 bits
Reloj	20 MHz
SRAM	2KB
EPROM	1 KB
USB	NO*
POTENCIA BAJA	NO
GPIO MÁX frecuencia de conmutación.	5 MHz
ADC	15 KSPS
DMA	NO
UEXT	NO
Voltaje de salida	3.3V-5V
Max. Current All Port-	200mA
Max. Current Pin	20mA
Memoria Flash	32KB
Temperatura	-40° hasta 85°
Voltaje de entrada	5v-12v
Pines digitales	12
Pines análogos	6
Microchip	ATMEGA328
Comunicación Serial	I2C, UART, SPI

Fuente: tabla creada por el alumno.

1.6.2 ¿Qué es ESP32? Características

Las tarjetas de desarrollo ESP32 de desarrollo son una gamma de tarjetas creadas por la empresa Espressif Systems, estas tarjetas tienen compatibilidad con la tecnología WI-FI y Bluetooth de modo dual integrada en el ESP32. Emplea un microchip Tensilica Xtensa LX6, este es un microchip de 2 núcleos y cuenta con 32 bits de transferencia de datos, cuenta con 4 puertos SPI, 3 puertos UART, 2 interfaces de comunicación I2C, 2 canales DAC, etc. Esta tarjeta requiere un voltaje de alimentación de 3.3V para su correcto funcionamiento. Todas sus características se pueden apreciar en la tabla 1-12.

Esta tarjeta es una tarjeta de bajo consumo con un enfoque en el tamaño. Esto le proporciona una reducción drástica de tamaño con niveles de tecnologías más nuevas en comparación de otras tarjetas microcontroladora comparadas en este trabajo.

Tabla 1-12. Características de la tarjeta microcontroladora ESP32 WROOM

DATO	ESP32 WROOM
Bits de Transferencia	32 bits
Reloj	160MHZ
SRAM	520KB
EPROM	4 KB
micro USB	Si
POTENCIA BAJA	Si
GPIO MÁX. frecuencia de conmutación	160 MHz
ADC	6 KSPS
DMA	NO
UEXT	NO
Voltaje de salida	3.0V-3.6V
Max. Current All Port-	1100mA
Max. Current Pin	40mA
Memoria Flash	16MB
Temperatura	-40°C hasta 85°C
Voltaje de entrada	3.6V
Pines digitales	36
Pines análogos	12 Max 3.3V
Microchip	Tensilica Xtensa LX6
Comunicación Serial	I2C, UART, SPI

Fuente: Tabla comparativa creada por el alumno.

1.6.3 Tabla Comparativa

En la tabla 1-13 se presenta una comparación realizada por el alumno que completa de todas las características relacionadas a las tarjetas de desarrollo Arduino UNO, Pingüino OTG, ESP32 WROOM.

Tabla 1-13. Tabla comparativa

DATO	ARDUINO UNO	PINGÜINO OTG	ESP32 WROOM
Bits de Transferencia	8 bits	32 bits	32 bits
Reloj	20 MHz	80 MHz	160 MHz
SRAM	2KB	32 KB	520 KB
EPROM	1 KB	8 KB	4 KB
USB	NO	SÍ	NO
Micro USB	NO	NO	SI
POTENCIA BAJA	NO	SÍ	SI
GPIO MÁX frecuencia de conmutación.	5 MHz	40 MHz	160 MHz
ADC	15 KSPS	1.1 MSPS	6 KSPS
DMA	NO	SÍ	SI
UEXT	NO	SÍ	NO
Voltaje de salida	5V-3.3V	5V-3.3V	3.3V
Max. Current All Port-	200mA	185mA	1100mA
Max. Current Pin	20mA	25mA	40mA
Memoria Flash	32KB	256KB	16MB
Temperatura	-40°c hasta 85°c	-25°c hasta 85°c	-40°c hasta 85°c
Voltaje de entrada	5v-12v	+5v USB, 3.7v Li BAT, 9v-30v externo	2.6V-3.3V
Pines digitales	12	14	36
Pines análogos	6	6	12
Microchip	ATMEGA328	Pic32MX440F256H	Tensilica Xtensa LX6
Comunicación Serial	SPI, UART, I2C	SPI, UART, I2C	SPI, UART, I2C, I2S
Bluetooth, Wi-Fi	NO	NO	SI

Fuente: Tabla creada por el alumno.

1.7 ¿CÓMO SE PROGRAMA?

Las tarjetas de desarrollo pingüino son de arquitectura PIC, este procesador es programado mediante ASSEMBLER un método de programación complejo y poco intuitivo. En busca de un método para poder programar los microchips PIC se desarrollaron aplicaciones llamadas compiladores, los programas compiladores se encargan de programar eficazmente estas tarjetas microcontroladoras.

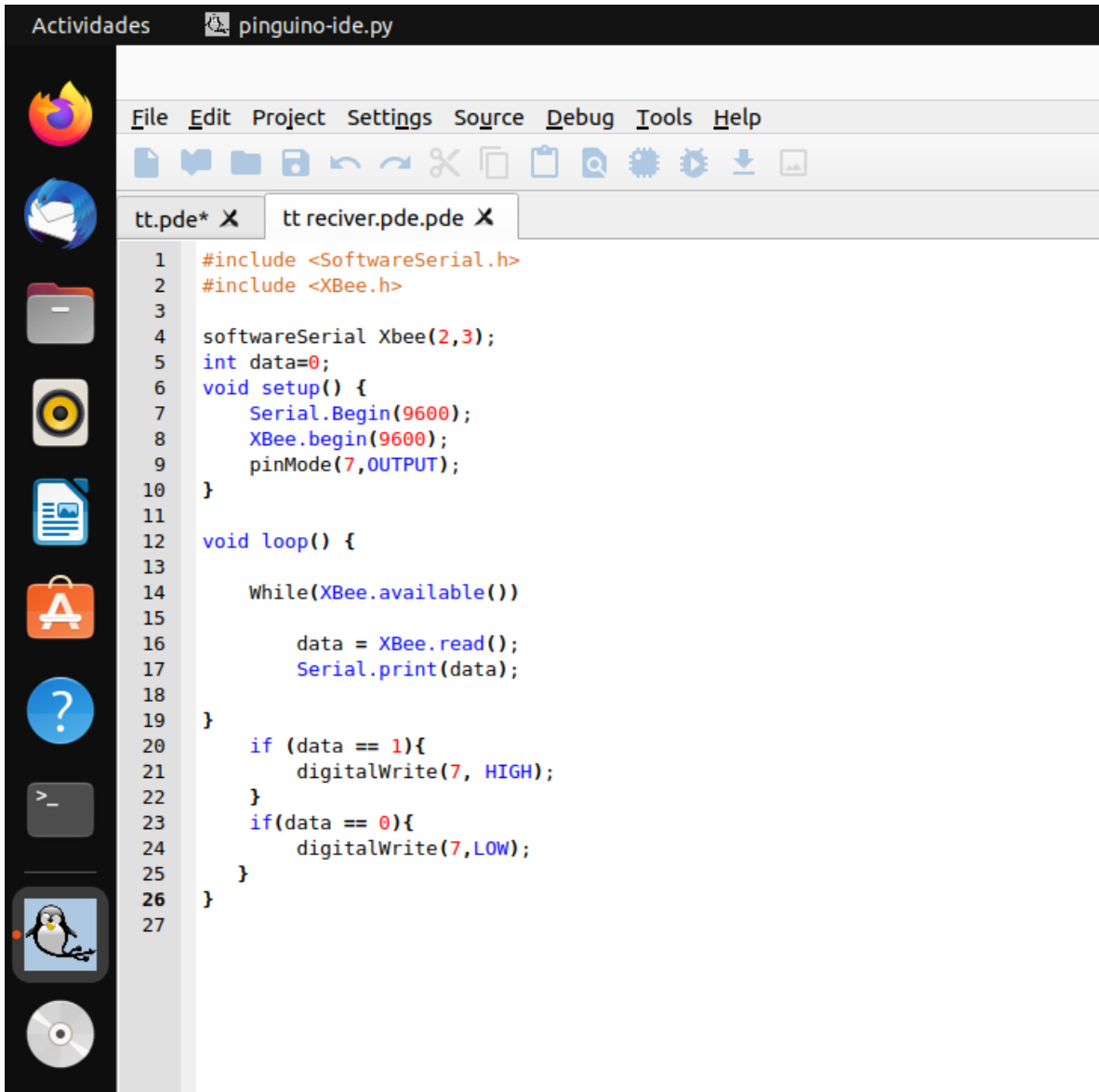
Arduino en este caso lanzó su plataforma de compilación llamada IDE Arduino, esta plataforma de compilación está basada en una versión simplificada del lenguaje C++ permitiendo crear un ambiente amigable para el aprendizaje, a diferencia de su competencia Arduino Uno, esta tarjeta microcontroladora desarrolló aparte varias plataformas de compilación para su trabajo los cuales se analizan a continuación.

1.7.1 IDE Pingüino

Esta aplicación desarrolladora es un proyecto enfocado en Linux para la compilación de programa en microchips PIC esta plataforma es de código abierto, este programa está basado en Python por ende requiere tener Python instalado, la intención de este proyecto es crear un compilador que contenga la interfaz de Arduino. En grandes rangos este compilador es poco recomendable porque su enfoque de trabajo está direccionado a la plataforma Linux.

Para probar el Ide Pingüino en la tarjeta Pingüino es recomendable descargarlo por <https://github.com/PinguinoIDE>.

En la figura 1-50. Se desarrolló en la plataforma Linux un programa con XBee para comprobar su correcto funcionamiento.



Fuente: Aplicación desarrollada por el alumno.

Figura 1-50. Desarrollo de aplicación simple en IDE pingüino con máquina virtual UBUNTU

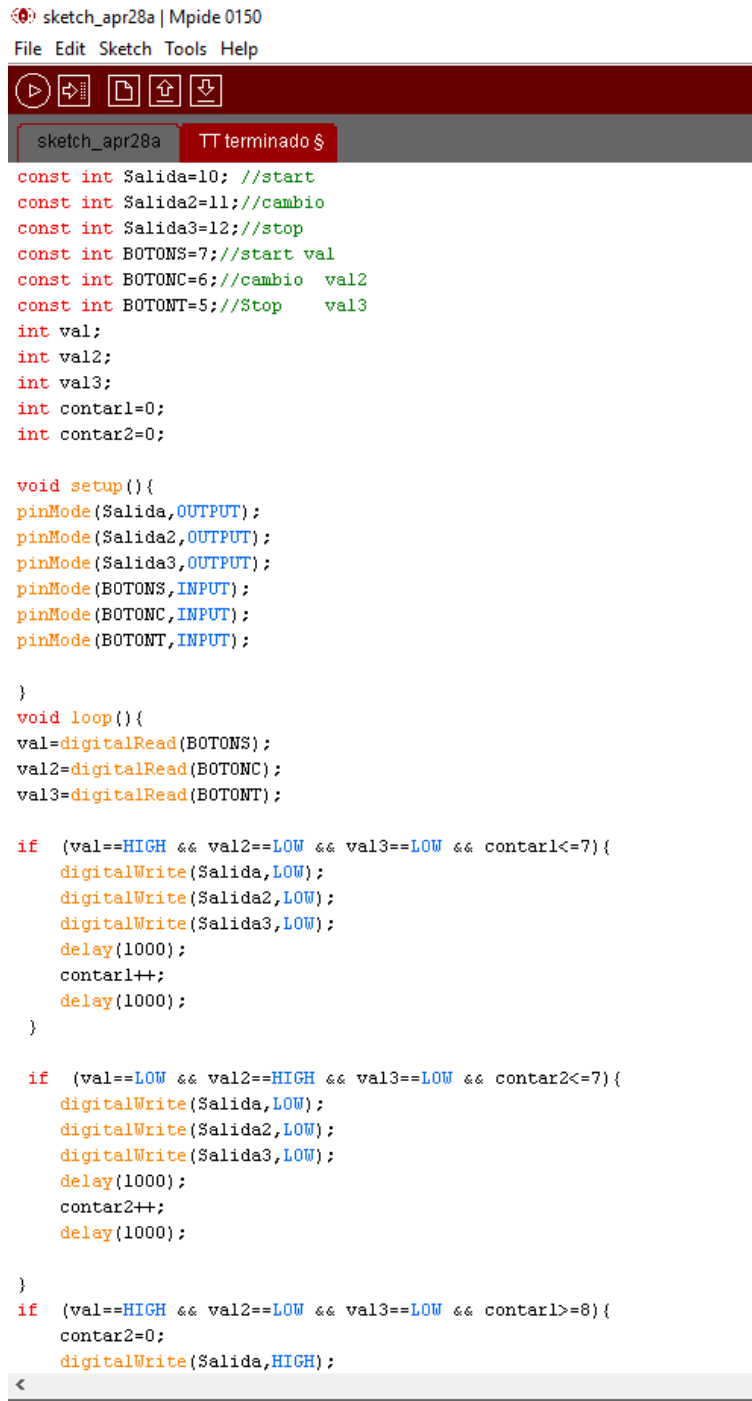
1.7.2 MPIDE

MPIDE es una plataforma compiladora la cual compila en C++ a la arquitectura de pic 32, este programa está basado en Java Script el cual intenta replicar la interfaz de IDE Arduino, este compilador toma este enfoque en busca de una mejor accesibilidad para el usuario. En la figura 1-51. Se presenta el desarrollo del proyecto como ejemplo del MPIDE.

MPIDE presenta una interfaz similar a IDE ARDUINO el cual tiene su respectivo botón de verificar y cargar, esta aplicación compiladora fue la seleccionada para desarrollar la prueba de trabajo de la tarjeta microcontroladora.

La Aplicación MPIDE se puede obtener mediante el siguiente Link.

<https://digilent.com/reference/software/mpide/start>



```

sketch_apr28a | Mptide 0150
File Edit Sketch Tools Help

sketch_apr28a TT terminado $

const int Salida=10; //start
const int Salida2=11; //cambio
const int Salida3=12; //stop
const int BOTONS=7; //start val
const int BOTONC=6; //cambio val2
const int BOTONT=5; //Stop val3
int val;
int val2;
int val3;
int contar1=0;
int contar2=0;

void setup(){
  pinMode(Salida,OUTPUT);
  pinMode(Salida2,OUTPUT);
  pinMode(Salida3,OUTPUT);
  pinMode(BOTONS,INPUT);
  pinMode(BOTONC,INPUT);
  pinMode(BOTONT,INPUT);
}

void loop(){
  val=digitalRead(BOTONS);
  val2=digitalRead(BOTONC);
  val3=digitalRead(BOTONT);

  if (val==HIGH && val2==LOW && val3==LOW && contar1<=7){
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
    contar1++;
    delay(1000);
  }

  if (val==LOW && val2==HIGH && val3==LOW && contar2<=7){
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
    contar2++;
    delay(1000);
  }

  if (val==HIGH && val2==LOW && val3==LOW && contar1>=8){
    contar2=0;
    digitalWrite(Salida,HIGH);
  }

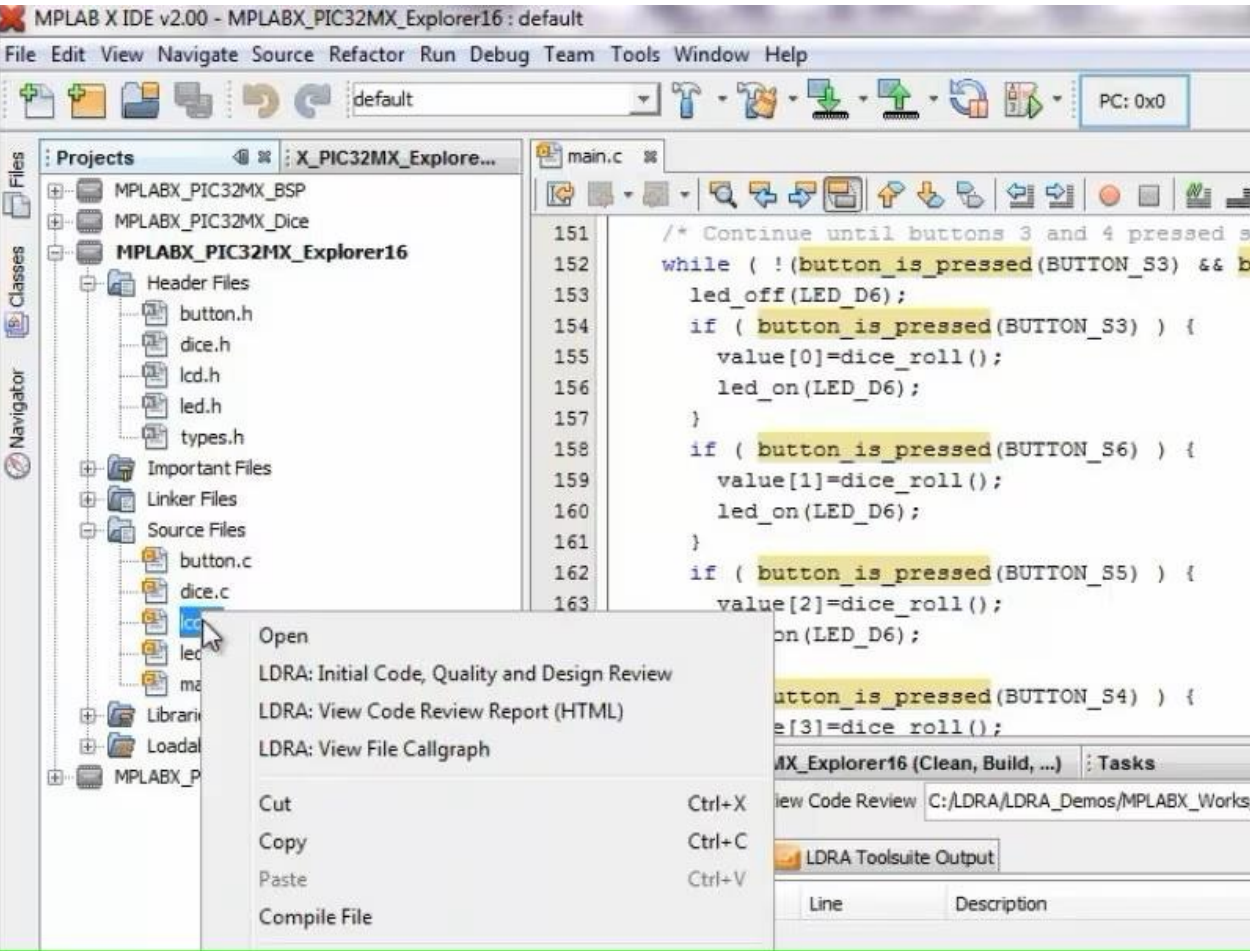
```

Fuente: Aplicación desarrollada por el alumno.

Figura 1-51. Desarrollo de aplicación de prueba de tarjeta microcontroladora con MPIDE

1.7.3 MPLAB

MPLAB es la plataforma copiladora más moderna para poder trabajar con las tarjetas microcontroladoras PIC, esta plataforma compiladora está basada en C++ y cuenta con su propia interfaz de programación, cuenta con características única que otorgan un mayor manejo del programa a ejecución y una visualización lo más cómoda posible. En la figura 1-52 se puede observar la interfaz de trabajo de MPLAB.



Fuente: <https://www.microchip.com/en-us/development-tool/sw500320>

Figura 1-52. Interfaz de programación MPLAB

CAPÍTULO 2: DESARROLLO PRACTICO DE LA TARJETA DE DESARROLLO PINGÜINO

2. DESARROLLO PRACTICO DE LA TARJETA DE DESARROLLO PINGÜINO

En el presente desarrollo del proyecto se decidió separar este mismo en 2 partes, la parte uno teórica que abarcará planteo de problemática, planteo de circuito, diseño de circuito, diagrama de flujo del circuito y componentes a usar con una breve explicación del funcionamiento de estos y la parte dos del proyecto abarcará desde el ensamblado del proyecto, puesta en ejecución y resultados del proyecto.

2.1 PLANTEO DEL PROBLEMA LA PROBLEMÁTICA

Después de analizar cómo está compuesto el microcontrolador pingüino desde sus componentes hasta su funcionamiento, se procederá a realizar en forma práctica lo aprendido hasta ahora mediante la aplicación de un circuito que sea capaz de poner en prueba el funcionamiento del microcontrolador Pingüino.

2.1.1 Planteo de problemática "Circuito"

En la selección de un circuito que permita comprender de forma práctica el funcionamiento y características únicas del microcontrolador pingüino, se seleccionó el control de un motor trifásico, el cual permita el cambio de giro. Este circuito contará con dos partes vitales, un circuito de fuerza donde se compondrá por contactores y disyuntor de seguridad, este circuito será el encargado de controlar el motor.

La segunda parte vital será el circuito de control donde se compondrá por relés, la tarjeta de desarrollo, transistores, resistencias y una sola línea de alimentación de 24V.

2.1.2 Planteo del circuito

Según lo planteado en el punto anterior se realizará el planteo teórico del circuito como estará compuesto y sus cálculos para ello.

El primer paso será el cálculo del transistor BJT utilizado, para ello se utilizó como base el TIP31 usado por su alta capacidad de corriente que posee.

Se comprende que el TIP 31 es un BJT tipo NPN y este es capaz de soportar un máximo de 3 amperios, este comprende un h_{fe} de 10 a 50, su corriente máxima de base es de 1 amperios por lo tanto se calcula I_B para comprobar el funcionamiento correcto.

$$VCC = RB \cdot IB + VBE + VE$$

Por ende, se calcula la corriente con un resistor de 1kΩ, conociendo que $VE=0$, $VBE=0,7$ y $VCC=5$ ya que este corresponde al valor máximo de la salida del microcontrolador. Se calcula.

$$5V = 1000 \cdot IB + 0.7$$

$$IB = 5(mA)$$

Como resultado se obtiene IB de 5 (ma) óptimo para el correcto funcionamiento. Se selecciona una resistencia de 1kΩ para el trabajo.

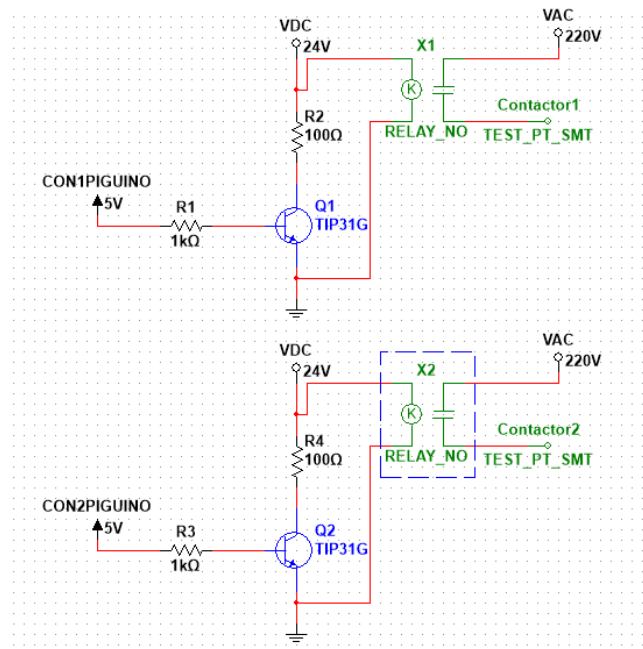
Ahora se calcula IC de 3 A para la resistencia mínima comprendiendo $VCE=0$ y $VE=0$.

$$24V = IC \cdot RC + VCE + VE$$

$$RC = \frac{24}{3}$$

$$RC = 8$$

Por ende, la resistencia mínima es de 8Ω, pero se selecciona una de 100Ω para mayor seguridad y comodidad ya que esta es fácil de conseguir. Ahora se plantea el circuito mediante la plataforma MULTISIM como se observa en la figura 2-1.

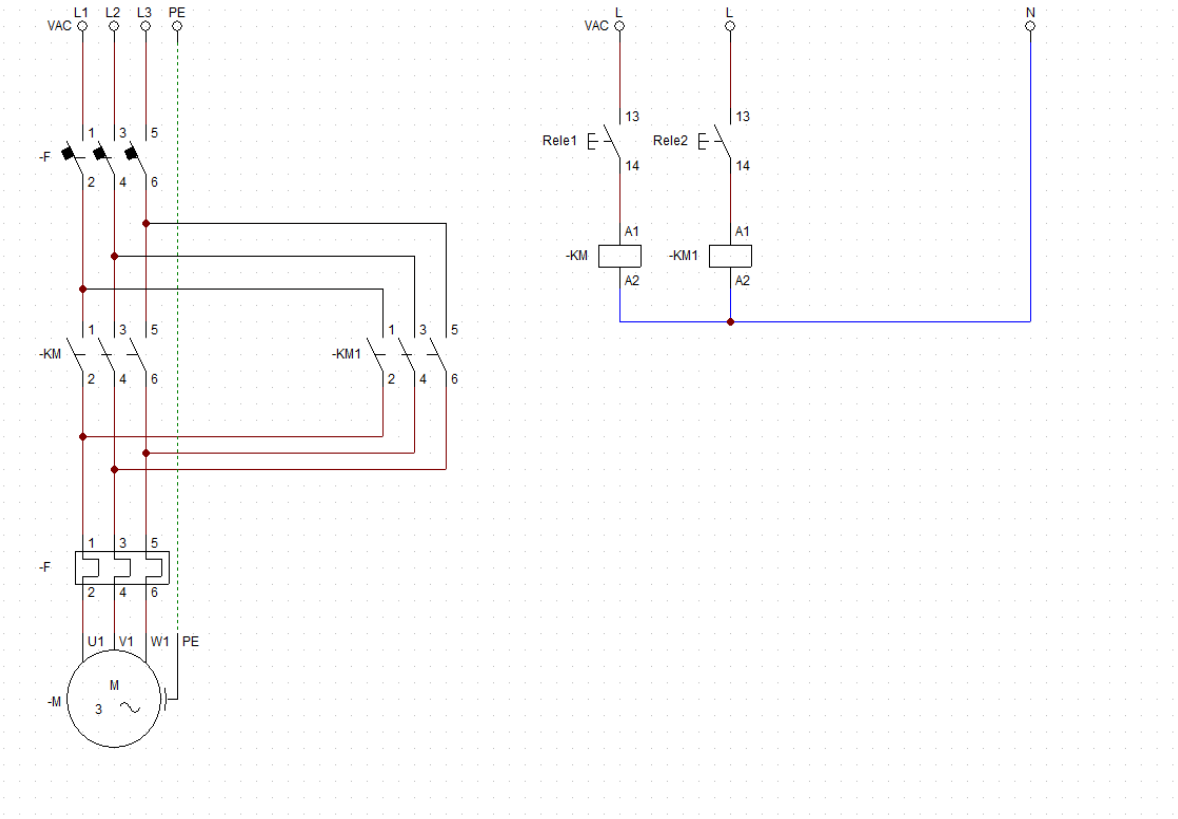


Fuente: Circuito creado por el alumno

Figura 2-1 Circuito de control

En la figura 2-1. Se puede apreciar que se agregó un relé de control, este corresponde al relé de control de 24V que será usado en el contactor para controlar una corriente de 220V, para mayor claridad se denominó el contactor como contactor 1 y 2 correspondiente al circuito de fuerza.

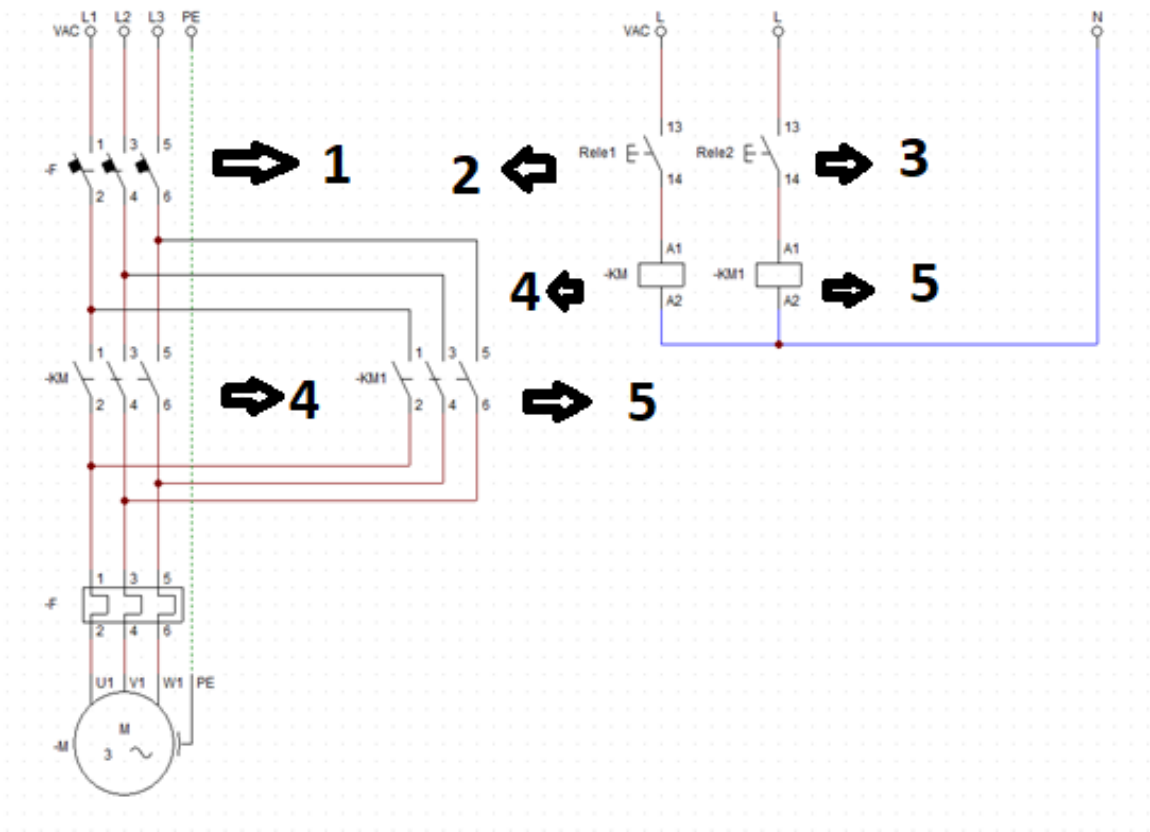
En la resolución del planteo del circuito de fuerza se analizó un control de cambio de giro el cual consta de varios componentes la línea trifásica, disyuntores, contactores, motor trifásico, relés de control. Al comprender el funcionamiento de un motor trifásico que esté siendo alimentado por tres canales siendo R, S y T con 380V y que permite el giro en un sentido u otro, lo que se puede apreciar invirtiendo dos canales de alimentación. En la figura 2-2 se puede apreciar el circuito diseñado para el circuito de fuerza, el circuito fue planteado en la aplicación de CADESIMU .



Fuente: Circuito creado por el alumno

Figura 2-2 Circuito de fuerza

Según lo presentado en la figura 2-2 se puede apreciar que el circuito de fuerza se compone por un control de contactores que son controlados por los relés de la figura 2-1 esto se puede apreciar en la figura 2-3, en la cual se muestra que se editó para una mayor comprensión, el número 1 corresponde a los disyuntores, el 2 y 3 corresponde a los relés propuestos en la figura 2-1, el número 4 y 5 corresponde a los disyuntores que se cerrarán y abrirán correspondiente al estado de los relés alimentando el motor.



Fuente: Circuito creado por el alumno

Figura 2-3 Circuito de fuerza editado

Y por último se revisa el programa Lógico encargado del control del proyecto, el programa lógico para comprobar el funcionamiento se realizó con dos versiones del programa compilador, la versión de MPIDE e ID Pingüino. La ejecución consiste en un IF encadenado el cual cuenta con un contador que inicia al presionar BOTON el cual sería el START del circuito, el cual al terminar de contar arrancará el motor, el botón de stop se encargará de detener el contador y los motores, y el botón BOTONC se encargará del cambio de giro ejecutando la otra salida, el programa imposibilita el uso de dos botones bloqueando el otro cuando uno está activo y

evitará los cambios bruscos por un cambio repentino de sentido de giro del motor para evitar daños a este. En la figura 2-4 y 2-5 se puede apreciar el programa utilizado para el microprocesador, el cual se separa en 2 imágenes para la posibilidad de observarlo completo. Este contador está programado para funcionar después de 16 segundos “Configurable”.

```
const int Salida=10; //start
const int Salida2=11;//cambio
const int Salida3=12;//stop
const int BOTONS=7;//start val
const int BOTONC=6;//cambio val2
const int BOTONT=5;//Stop val3
int val;
int val2;
int val3;
int contar1=0;
int contar2=0;

void setup(){
pinMode(Salida,OUTPUT);
pinMode(Salida2,OUTPUT);
pinMode(Salida3,OUTPUT);
pinMode(BOTONS,INPUT);
pinMode(BOTONC,INPUT);
pinMode(BOTONT,INPUT);
}

void loop(){
val=digitalRead(BOTONS);
val2=digitalRead(BOTONC);
val3=digitalRead(BOTONT);

if (val==HIGH && val2==LOW && val3==LOW && contar1<=7){
digitalWrite(Salida,LOW);
digitalWrite(Salida2,LOW);
digitalWrite(Salida3,LOW);
delay(1000);
contar1++;
delay(1000);
}

if (val==LOW && val2==HIGH && val3==LOW && contar2<=7){
digitalWrite(Salida,LOW);
digitalWrite(Salida2,LOW);
digitalWrite(Salida3,LOW);
delay(1000);
contar2++;
delay(1000);
}

if (val==HIGH && val2==LOW && val3==LOW && contar1>=8){
contar2=0;
digitalWrite(Salida,HIGH);
digitalWrite(Salida2,LOW);
digitalWrite(Salida3,LOW);
delay(1000);
}

}
```

Fuente: Circuito creado por el alumno

Figura 2-4 Programa del Proyecto parte 1

```

if (val==LOW && val2==HIGH && val3==LOW && contar2>=8){
    contar1=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,HIGH);
    digitalWrite(Salida3,LOW);
    delay(1000);
}
if (val==LOW && val2==LOW && val3==HIGH ){
    contar1=0;
    contar2=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
}
if (val==HIGH && val2==LOW && val3==HIGH ){
    contar1=0;
    contar2=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
}
if (val==HIGH && val2==HIGH && val3==HIGH ){
    contar1=0;
    contar2=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
}
if (val==LOW && val2==HIGH && val3==HIGH ){
    contar1=0;
    contar2=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
}
if (val==HIGH && val2==HIGH && val3==LOW ){
    contar1=0;
    contar2=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
}
if (val==LOW && val2==LOW && val3==LOW ){
    contar1=0;
    contar2=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
}
}
}

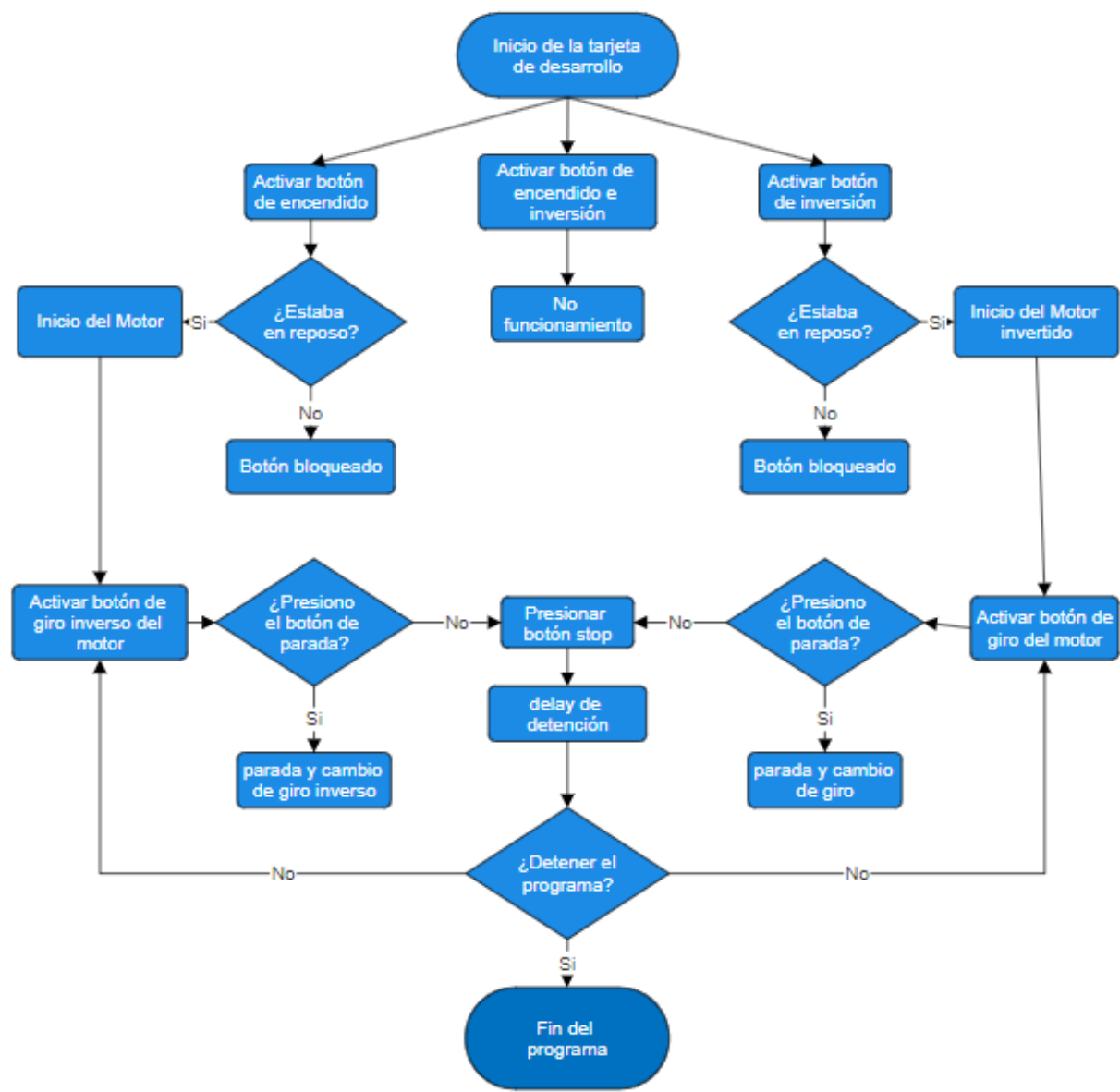
```

Fuente: Circuito creado por el alumno

Figura 2-5 Programa del Proyecto parte 2

2.1.3 Diagrama de flujo del circuito

Para mejor comprensión del funcionamiento del circuito se planteó un diagrama de Flujo que esquematizará el control del circuito completo, este diagrama esquematiza todas las acciones físicas que ejecuta el programa. Como se puede observar en la figura 2-6 se representa las acciones al momento de presionar un botón de inicio, inversión, stop.



Fuente: diagrama creado por el alumno

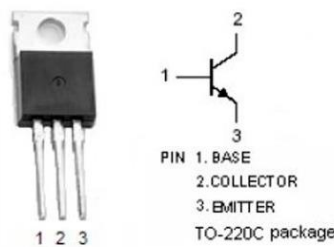
Figura 2-6 Diagrama de Flujo del proyecto

2.1.4. Componentes utilizados en el circuito.

En el circuito empleado se empleó una cantidad de componentes los cuales son vitales para el correcto funcionamiento del circuito, los componentes más caros fueron proporcionados por la universidad en un afán de disminuir costos. Los siguientes componentes se listarán a continuación.

TIP 31:

TIP 31 es un transistor BJT NPN, este componente es utilizado como amplificador de corriente o interruptor electrónico, el TIP 31 es especial ya que tiene la característica de soportar altas cargas de corriente variables entre 3A a 5^a, este transistor es vital para el control de los relés para un óptimo funcionamiento como se puede apreciar en la figura 2-7.



Fuente: <https://www.mechatronicstore.cl>

Figura 2-7. Tip 31 BJT tipo NPN

Fuente 24V:

Para el funcionamiento y puesta a prueba de la tarjeta microcontroladora, la que es capaz de soportar altos voltajes de alimentación, se puso a prueba su control con una fuente de 24 volts que alimentaba la tarjeta, el transistor y relés de control. En este caso se usó una fuente variable para el trabajo como única línea de alimentación en el circuito de control. Los relés de control trabajan con un voltaje de control de 24V y un voltaje de ejecución de 220V con una capacidad máxima de corriente de 10A como se puede apreciar en la figura 2-8.



Fuente: <https://www.finaltest.com.mx/product-p/art-6.htm>

Figura 2-8 Fuente Variable Sorensen

Resistores, relés y Contactores:

En el procedimiento de armado se calcularon los valores resistivos de las resistencias, estas variando desde 100ohm hasta 1kohm para el control del transistor, el relé es uno con activación de 24V el cual controla el paso de la energía y los contactores son de tres canales para uso trifásico. La carga del circuito de control en este caso es el Contactor el cual se representa en el circuito de fuerza visto en la figura 2-3.



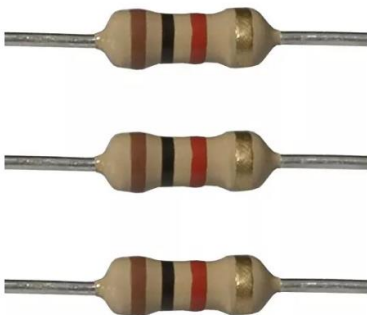
Fuente: <https://es.slideshare.net/slideshow/dimensionamiento-y-montaje-de-tablero-electrico-para-motores-de-induccion/269953330>

Figura 2-9 Contactor ECX18



Fuente: <https://www.mantenimientoelectrico.com/relays/relays-que-sirven-y-que-tipos-existen-n4226>

Figura 2-10 Relé de control de alta potencia



Fuente: https://articulo.mercadolibre.cl/MLC-1392445511-resistencias-de-1k-ohm-e-projects-100ep5121k00-12-w-10-_JM

Figura 2-11 Resistencia genéricas.

Motor de inducción Trifásica:

El Motor trifásico con los dos contactores y el disyuntor trifásico componen lo que es el circuito de fuerza, con la activación de los relés del circuito de control al presionar el Start o inversión de giro, activará los contactores representados como KM y KM1 respectivamente en la figura 2-2 y 2-3 para realizar el giro de motor en un sentido u otro. El funcionamiento de un motor de inducción trifásico se puede explicar de forma tal que este se compone de secciones tales como, el rotor, el estator, el devanado ubicado en el estator, el devanado de la armadura, un ventilador. El devanado del estator al ser alimentado genera un campo electromagnético que ejerce fuerza al rotor permitiendo el giro de este. Este campo puede invertir el giro alimentando inversamente dos canales del motor de inducción. En la figura 2-12 se puede apreciar cómo está compuesto el motor usado. Sus características cuentan con una potencia de 3 (HP), 50 (HZ) de frecuencia, un factor de potencia de 0.81(FP), un factor de servicio de 1,15(FS) proporcionando por corto periodo hasta 3.45 (HP), Voltajes de funcionamiento de 220(V) y 380(V)



Fuente: https://www.editores-srl.com.ar/sites/default/files/ricardo_berizzo_20211117_figura_1_full.jpg

Figura 2-11 Motor de inducción trifásica.

2.2 PLANTEO DEL PROBLEMA PRÁCTICO.

En el actual punto se presentará la puesta en escena del proyecto, para esto el proyecto se realizó en el laboratorio de la universidad en el cual se proporcionaron los materiales necesarios para lograr el trabajo de una forma más económica. En este punto se abarcará el proceso de montaje y resultado de este.

2.2.1. Ensamblado del proyecto.

El primer paso que se realizó fue el análisis de los terminales a usar de la tarjeta de desarrollo Pingüino, para esto se utilizó los terminales D10, D11, como salidas de control a los relés estos encargados de controlar el Start “D10” y cambio de giro “D11”. Como entradas se utilizaron D7, D6, D5. Estos corresponden respectivamente a D7 “botón de Start”, D6 “botón de cambio de giro”, D5 “botón de stop”. La alimentación se ocupó por el terminal VIN utilizando la fuente variable que se proporcionó en la universidad. En la figura 2-12 se puede apreciar la Protoboard utilizada y en la figura 2-13 el resultado del armado.



Fuente: Placa microcontroladora del alumno

Figura 2-12 Microcontrolador Pingüino OTG



Fuente: Placa microcontroladora del alumno

Figura 2-13 Microcontrolador Pingüino OTG armado

El segundo paso consistió en el conexionado desde los contactores y relés hacia el motor, la ausencia de instalación de un disyuntor se produce ya que los tableros de seguridad trifásica del motor ya vienen con un disyuntor de seguridad incorporado para la seguridad del alumno, por ende, se procede a la instalación de los relés de control hacia contactores para terminar en el motor como se observa en la figura 2-13 y figura 2-14.



Fuente: Componentes proporcionado para el alumno

Figura 2-14 Relés de alta corriente y contactores armado.

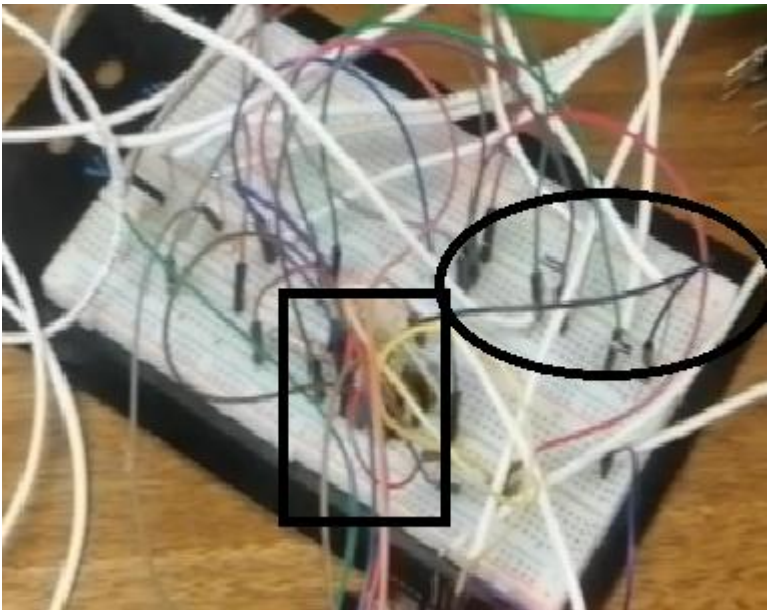


Fuente: Componentes proporcionado para el alumno

Figura 2-15 Motor de inducción trifásica armado.

Como última parte se procedió a la instalación del circuito de control para completar la instalación física como se puede observar en la figura 2-16, se recomienda enfáticamente ver el video del funcionamiento del proyecto para una vista completa de este. En la figura 2-16 se puede observar encerrados en un círculo algunos diodos, encargados de evitar corrientes inversas en los relés y en el cuadro se ubican los transistores TIP31.

<https://drive.google.com/file/d/1JrhfpavNxpxdJmV2abS70PYuoDEJd3AE/view?usp=sharing>



Fuente: Circuito del alumno

Figura 2-16 Circuito de control en Protoboard.

2.2.2. Resultados del proyecto.

Después de la ejecución del proyecto el resultado fue el control de giro de un motor con control manual, los resultados se presentaron con contratiempo por errores tales como no haber elegido el transistor correcto o errores al momento de programar, el resultado se puede apreciar en el siguiente Link que mostrará un video del funcionamiento.

<https://drive.google.com/file/d/1JrhfpavNxpxdJmV2abS70PYuoDEJd3AE/view?usp=sharing> .

Como se puede apreciar en el video, el funcionamiento lógico funciona con la razón presentada en la tabla 2-1, los pines utilizados y sus funciones.

Tabla 2-1. Control Lógico y terminales de activación

Switch	Acción	Terminal del Microcontrolador	Bloqueo "0" Activo "1"	Bloqueo "0" Activo "1"
Start	Arranque	D7	1	0
Start Invertido	Arranque invertido	D6	0	1
Stop	Parado del Motor	D5	1	1

Fuente: Tabla creada por el alumno.

CAPÍTULO 3: CONSIDERACIONES Y COSTOS DEL PROYECTO

3. CONSIDERACIONES Y COSTOS DEL PROYECTO

En el presente capitulo, después de haber realizado una proyección teórica, un desarrollo de ejecución práctica, se realizará una proyección de las consideraciones y costos del proyecto. En la proyección económica del proyecto se presentará un análisis del valor de las unidades en diferentes mercados disponibles y el costo de la mano de obra que requiera este proyecto. Mientras tanto se verá un enfoque relacionados consideraciones y recomendaciones.

3.1 CONSIDERACIONES Y RECOMENDACIONES.

En esta sección se revisará una serie de consideraciones y recomendaciones que se separan en dos ámbitos, uno en la instalación y otro en la programación del microcontrolador pingüino. Estas dos consideraciones y recomendaciones son una serie de consejos que permitan el correcto funcionamiento de la ejecución del proyecto con una tarjeta de desarrollo Pingüino OTG, si ejecuta una solución alternativa para lograr el objetivo es aceptable.

3.1.1 Consideraciones en la Instalación.

En el desarrollo del proyecto se logró presenciar una cierta cantidad de complicaciones que puede reiterarse si no se tiene una noción de estos, estos pueden abarcar desde.

Revisión de la calidad de los componentes.

Mientras se realizaba la instalación del proyecto ocasionó que se ocuparan componentes los cuales se habían estropeado confundiendo el resultado final. Otra problemática se puede ocasionar el elegir componentes sin la capacidad de soportar las corrientes del dispositivo dañándolo permanente.

Precaución de trabajar con las líneas de alimentación apagadas.

Al momento de trabajar con energías de alta potencia se requiere energéticamente trabajar con estas en forma segura y apagadas para evitar accidentes. Esto puede abarcar desde todos los rangos de voltaje, ya que independientemente si el voltaje es bajo este puede dañar componentes y resumirse en daño material.

Usar equipo de seguridad correspondiente.

En el desarrollo de este proyecto se requiere la necesidad de usar equipo de seguridad para prevenir accidentes mayores, estos equipos abarcan desde guantes eléctricos, una capa de seguridad y bototos de seguridad.

3.1.2 Consideraciones en Programación.

Al momento de programar la tarjeta microcontroladora Pingüino se tiene una serie de consideraciones que se recomiendan para el uso adecuado de este, las recomendaciones son las dictadas a continuación.

Pleno control de Linux.

En el proceso de verificación de funcionamiento de las tarjetas microcontroladoras se encontraron variadas complicaciones al momento de ejecutarse en una máquina virtual, estas complicaciones abarcaban la incompatibilidad de reconocimiento de los puertos para el control del microcontrolador, problemas de instalación de librerías. Estos problemas aparecieron cuando se desarrolló la ejecución de Pingüino IDE en una máquina virtual UBUNTU los cuales se solucionaron cuando se le dieron permisos correspondientes en el root.

Revisión de método de instalación de librerías

Cuando se ejecutó el programa MPIDE se descubrió que su método de instalación de librerías es totalmente distinto a Arduino, y para poder simular correctamente el microprocesador Pingüino es necesario instalar las librerías PIC 32 Pingüino OTG, el método utilizado se puede apreciar en el link informativo del capítulo 1 sección programación de MPIDE.

Conocimiento previo de lenguaje de programación IDE Arduino

Ide Pingüino tanto como MPIDE son plataformas que copian el estilo de programación e interfaz de Arduino, esto es tal que los programas de Pingüino e IDE se pueden usar correlativamente, esto se produce para un mejor control del microcontrolador.

3.2 **PROYECCION ECONÓMICA DEL PROYECTO DE DESARROLLO PINGÜINO “ESTUDIO ECONÓMICO”**

En la proyección económica del proyecto se revisará un análisis del valor de las unidades en diferentes mercados disponibles, también se revisará una proyección del costo de mano de obra para ejecutar el proyecto y, por último, se revisará una proyección de los costos finales vistos en dos ámbitos posibles el ámbito privado, el ámbito estudiantil.

3.2.1 **Proyección “mano de obra”**

La ejecución del proyecto al estar enfocado en el ámbito educativo sus costos con relación a mano de obra no son un costo mensual de contratación sino un costo educativo, para poder emplear correctamente el proyecto si se quiere ejecutarlo en forma real lo más recomendable es tomar un plan educativo relacionado a la electricidad, electrónica o informática, estos costos son un total mostrados en la tabla 3-1.

Tabla 3-1 Costos de profesional

Profesional requerido	Costo hora	Costo Total	Duración de instalación del Proyecto	Fuente
Técnico universitario electrónica	\$15.000/h - \$50.000/h pesos	\$120.000 Min.	1-2 Días “8-16 horas”	https://www.cronoshare.cl/cuanto-cuesta/electricista

Fuente: tabla creada por el alumno

3.2.2 Proyección “Costos Materiales”

En relación de los costos materiales se realizará una revisión de todos los costos materiales que puedan abarcar el proyecto, se revisará el costo de los componentes más significativos tales como los listados a continuación.

- Protoboard
- Motor Trifásico 3 HP
- transistor Tip 31 o 2n3055
- Resistencias 1KΩ y 100Ω
- contactores trifásicos
- disyuntor trifásico
- tarjeta de desarrollo Pingüino OTG
- Fuente de voltaje variable

Protoboard

La Protoboard es la encargada de ser la base la cual se construye el circuito de control, cuenta con precios accesibles para la fácil adquisición de particulares, en la presente tabla 3-2 se realizará una comparación económica de diferentes fuentes para racionalizar su precio.

Tabla 3-2 Mercado de Protoboard

Tienda	Link	Precio
MCI Electronics	https://mcielectronics.cl/shop/category/electronica-y-robotica/prototipo/breadboard-protoboard/	\$3.490
Mercado libre	https://listado.mercadolibre.cl/protoboard	\$4.999
Afel	https://afel.cl/producto/protoboard-mb-102-830-puntos/	\$2.500
Arduino.cl	https://arduino.cl/producto/protoboard-de-tamano-completo-830pts-mb-102/	\$3.490
Altronic Chile	https://altronics.cl/breadboard-830	\$3.353

Fuente: tabla creada por el alumno

Motor trifásico

El motor trifásico es la parte fundamental del circuito de fuerza ya que sin esta el circuito no funcionaria, su costo es elevado a estándares privado, pero puede ser sustentado en el ámbito educativo por la misma universidad. En la tabla 3-3 se realizará una revisión económica de diferentes mercados para una adquisición de privado de este producto

Tabla 3-3 Costos de motor de inducción trifásica

Tienda	Link	Precio
imatesa	https://www.imatesa.cl/	\$79.113
Rhona	https://rhona.cl/producto/10599/motor-ac-trifasico-075hp.html	\$142.923
Mercado libre	https://listado.mercadolibre.cl/motores-electricos-trifasicos	\$118.000
ndu	https://ndu.cl/61-motores-electricos	\$112.213

Fuente: tabla creada por el alumno

Transistor TIP31

El transistor TIP es un componente de bajo costo, es utilizado para controlar voltajes de mayor valor para el control de los relés de alta potencia, como se aprecia en la tabla 3-4, sus costos no representan una gran inversión.

Tabla 3-4 Costos TIP31

Tienda	Link	Precio
Global Chile	https://globalelectronica.cl/bipolares-de-potencia/536-tip31c.html	\$460
Tienda 8	https://www.tienda8.cl/reemplazos-generales/tip31c-tip31-transistor-npn-100v-3a	\$1.200
Mecatronic store	https://www.mechatronicstore.cl/transistor-tip-31c-npn-bjt-80v-40w/	\$390
Altronics	https://altronics.cl/transistor-tip31c	\$433

Fuente: Tabla creada por el alumno

Resistencias de 1kΩ y 100Ω

Las resistencias se utilizan como limitadores de corriente los cuales evitan que los componentes se quemen, su precio se puede considerar despreciable como se muestra en la tabla 3-5.

Tabla 3-5. Costos de Resistencias.

Tienda	Link	Precio
Global Electronics	https://globalelectronica.cl/650-resistencias	\$70
altronics	https://altronics.cl/pack-10-resistencias-100ohm-025watt-1porciento	\$41
Bigtrónica	https://www.bigtronica.com/componentes-pasivos/resistencias/dip/14-watt/1385-resistencia-33k-ohm-14-w-10-und-5053212013851.html	\$30

Fuente: Tabla creada por el alumno

Contactores trifásicos

Los contactores trifásicos ejecutan el control de giro del motor, sus costos son aunque un poco elevados pueden ser en el plano estudiantil proporcionados por la universidad. Como se puede apreciar en la Tabla 3-6 se realizó una comparación de diferentes precios en el mercado.

Tabla 3-6 Costos contactores.

Tienda	Link	Precio
Wia	https://cl.wiautomation.com	\$14569
FerreEléctrica	https://www.ferrelectrica.cl	\$9400
Mercado libre	https://articulo.mercadolibre.cl/MLC-622134669-contactor-12a-bobina-220v-3pna-380v-trifasico-en-chile- JM?ma	\$9848
Zona Industrial	https://www.zonaindustrial.cl	\$11500

Fuente: Tabla realizada por el alumno

Disyuntor trifásico

El disyuntor trifásico efectúa como método de prevención ante cortos circuitos, su uso es altamente necesario y este en el ámbito estudiantil puede ser proporcionado por la universidad. Los costos se pueden observar en la Tabla 3-7, la cual abarca diferentes tipos de mercados.

Tabla 3-7 Costos Disyuntor trifásico

Tienda	Link	Precio
Diartek	https://diartek.cl	\$10.983
FerreElectrica	https://www.ferrelectrica.cl	\$6.490
Mercado Libre	https://www.mercadolibre.cl	\$9.250
Sodimac	https://sodimac.falabella.com/sodimac-cl/product/110228905/Interruptor-automatico-trifasico-32-A-curva-C-6KA/110228929	\$38.990

Fuente: Tabla creada por el alumno

Tarjeta de desarrollo Pingüino OTG

La tarjeta de desarrollo es el componente lógico del proyecto, sus costos son accesibles para el estudiante. En la tabla 3-8 se puede hacer una comparación de precios en diferentes mercados.

Tabla 3-8 Costos Pingüino OTG

Tienda	Link	Precio
Mouser electronic	https://www.mouser.cl/ProductDetail/Olimex-Ltd/PIC32-PINGUINO-OTG?qs=zi4u%2Fr02%252BjH%2F78wc5iPdRA%3D%3D	17.255
Olimex	https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/open-source-hardware	14.275
Aliexpress	https://es.aliexpress.com/i/1005005191022941.html	30.945

Fuente: tabla creada por el alumno

Fuente de voltaje variable.

Este es el encargado de suministrar la energía en el circuito de control, sus costos son altos, pero pueden ser proporcionado al estudiante por la universidad. Sus costos se pueden apreciar en la tabla 3-9.

Tabla 3-9 Costos Fuente de poder variable.

Tienda	Link	Precio
Mercado libre	https://www.mercadolibre.cl	\$69.990
MCI electronic	https://mcielectronics.cl/shop/product/fuente-de-poder-digital-regulable-de-0-a-30v-5amp-ele-tech-22626/	\$109.990
Mechatronic chile	https://www.mechatronicstore.cl/fuente-regulable-0-30vdc-0-10a-ms3010d/	\$119.900
Global Electronics	https://globalelectronica.cl/fuentes-de-alimentacion-para-laboratorio/4512-m10-qd305e.html	\$119.900

Fuente: Tabla creada por el alumno

3.2.3 Costos finales del proyecto con vista privada y estudiantil

Vista privada

Cuando es necesario realizar un estudio en el ámbito privado se tomará el promedio de todos los elementos vistos anteriormente y se realizará una tabla con estos, como se puede observar en la tabla 3-10. Es posible apreciar en la tabla que el costo es bastante elevado ya que se tiene que adquirir todos los componentes.

Tabla 3-10 Costos totales privado

Objeto	Costo
Protoboard	\$3.566
Motor	\$113.062
Transistor TIP31	\$620
Resistencias	\$47
Disyuntor trifásico	\$16.428
Fuente de poder	\$99.960
Contactores trifásicos	\$45.317
Tarjeta de desarrollo	\$20.825
Total	\$299.825

Fuente: Tabla creada por el alumno

Vista estudiantil

Realizar el proyecto siendo un estudiante en el caso de la Universidad Federico-Santa María, es posible apreciar una sustancial reducción de costos en el valor del proyecto, esto se realiza ya que los componentes más caros los proporciona la universidad. En la tabla 3-11 se puede apreciar los costos totales que paga el alumno, en esta tabla no ampara los costos de la carrera ,solo los costos materiales.

Tabla 3-11 Costos totales estudiantil

Objeto	Costo
Protoboard	\$3.566
Transistor TIP31	\$620
Resistencias	\$47
Tarjeta de desarrollo	\$20.825
Total	\$25.058

Fuente: Tabla creada por el alumno

CONCLUSIONES.

La Investigación realizada y ejecución de un proyecto permite comprender el funcionamiento y aplicación de la tarjeta de desarrollo Pingüino OTG, en la cual el microcontrolador presenta una serie de ventajas en comparación de otros microcontroladores tales como una mayor capacidad de procesamiento o la capacidad de ser alimentado con una señal de voltaje mayor, esto es gracias a los dos reguladores de voltaje que tiene “conmutado y lineal”.

El método de programación de la tarjeta microcontroladora es similar a Arduino IDE permitiendo una facilidad al momento de trabajar con esta tarjeta.

La desventaja que se le puede atribuirse es la escasa retroalimentación que presenta esta tarjeta microcontroladora, el encontrar información en momentos de dificultad se vuelve difícil, esto se debe ya que el mayor soporte se realiza con enfoque laboral y no de aprendizaje, dificultando la comprensión del funcionamiento a gente que no está debidamente preparada.

La tarjeta de desarrollo Pingüino se recomienda que sea utilizada para enseñanza teórica ya que en la práctica puede generar confusión a alumnos que recién empiezan la enseñanza en microcontroladores, en casos más avanzados o proyectos, el alumno tenga la opción de utilizar esta tarjeta de desarrollo teniendo presente sus respectivas ventajas y desventajas.

La recomendación propia al momento de trabajar con esta tarjeta microcontroladora es utilizar la aplicación MPIDE para iniciar en la programación de esta tarjeta y una vez adquiera experiencia ocupar la aplicación MPLAB ya que esta es una plataforma más avanzada con un lenguaje C++ especializado para microchip PIC.

BIBLIOGRAFIA

Manual de Pingüino OTG. Características y funcionalidades. [En línea]

<https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/resources/PIC32-PINGUINO_AND_OTG.pdf> [consulta: 11 julio 2024]

Reguladores DC A DC en modo conmutado. [En línea]

<<https://es.scribd.com/document/367145791/Reguladores-de-CD-a-CD-en-Modo-Conmutado>> [consulta: 11 julio 2024]

Reguladores DC A DC en modo conmutado y Convertidores. [En línea]

<<https://unicrom.com/reguladores-de-conmutacion-convertidores-dc-d/>> [consulta: 11 julio 2024]

Understanding the Advantages and Disadvantages of Linear Regulators. [En línea]

<<https://www.digikey.com.br/pt/articles/understanding-the-advantages-and-disadvantages-of-linear-regulators#:~:text=Los%20reguladores%20lineales%20tienen%20la,entrada%20como%20sus%20homólogos%20conmutados>> [consulta: 11 julio 2024]

¿De cuánto es el regulador de voltaje que incluye Arduino uno? [En línea]

<<https://forum.arduino.cc/t/de-cuanto-es-el-regulador-de-voltaje-que-incluye-arduino-uno/67970/3>> [consulta: 11 julio 2024]

Manual de esquemático Arduino Uno. [En línea]

<<https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf>> [consulta: 11 julio 2024]

Regulador de tensión lineal LM317. [En línea]

<<https://www.hwlibre.com/lm317/>> [consulta: 11 julio 2024]

Características del microchip ATMEGA. [En línea]

<<https://www.mouser.cl/new/microchip/atmelatmega328/>> [consulta: 11 julio 2024]

Regulación lineal. [En línea]

<[https://espanol.libretexts.org/Ingenieria/Amplificadores_Operacionales_y_Circuitos_Integrados_Lineales_-_Teoría_y_Aplicación_\(Fiore\)/08%3A_Regulación_de_voltaje/8.03%3A_Reguladores_Lineales](https://espanol.libretexts.org/Ingenieria/Amplificadores_Operacionales_y_Circuitos_Integrados_Lineales_-_Teoría_y_Aplicación_(Fiore)/08%3A_Regulación_de_voltaje/8.03%3A_Reguladores_Lineales)>
[consulta: 11 julio 2024]

Memoria EEPROM [En línea]

<<https://html.alldatasheet.com/html-pdf/1179101/ESPRESSIF/ESP-WROOM-32/4617/8/ESP-WROOM-32.html>> [consulta: 11 julio 2024]

Funcionamiento de una memoria. [En línea]

<https://www.it.uc3m.es/pbasanta/asng/course_notes/ch02s03.html> [consulta: 11 julio 2024]

ESP32 DevKitC. [En línea]

< <https://www.infootec.net/esp32-devkitc-v4> > [consulta: 11 julio 2024]

ESP32 Esquema. [En línea]

< https://dl.espressif.com/dl/schematics/esp32_devkitc_v4-sch.pdf > [consulta: 11 julio 2024]

Pinout y detalles del hardware del ESP32. [En línea]

< <https://www.luisllamas.es/esp32-detalles-hardware-pinout/> > [consulta: 11 julio 2024]

Procesador ESP32-D2WD. [En línea]

<<https://www.mouser.cl/ProductDetail/Esspressif-Systems/ESP32-D2WD?qs=chTDxNqvsynYfhF3um46Mw%3D%3D>>[consulta: 11 julio 2024]

Tipos de conectores USB. [En línea]

<<https://hardzone.es/tutoriales/rendimiento/conector-usb-tipos/>>[consulta: 11 julio 2024]

Bus de comunicación I2C. [En línea]

<<https://biblus.us.es/bibing/proyectos/abreproy/70099/fichero/Trabajo+Fin+de+Master+-+Vicente+Madero%252F14+-+Apéndice+H+-+I2C.pdf>>[consulta: 11 julio 2024]

¿Qué es UART? [En línea]

<[https://www.rohde-schwarz.com/es/productos/test-y-medida/essentials-test-equipment/digital-oscilloscopes/que-es-uart_254524.html#:~:text=UART%20\(universal%20asynchronous%20receiver%20%2F%20transmitter,y%20recibir%20en%20ambas%20direcciones](https://www.rohde-schwarz.com/es/productos/test-y-medida/essentials-test-equipment/digital-oscilloscopes/que-es-uart_254524.html#:~:text=UART%20(universal%20asynchronous%20receiver%20%2F%20transmitter,y%20recibir%20en%20ambas%20direcciones) >[consulta: 11 julio 2024]

Comunicación SPI con PIC. [En línea]

<<https://controlautomaticoeducacion.com/microcontroladores-pic/comunicacion-spi/> >[consulta: 11 julio 2024]

Protocolo SPI. [En línea]

< <https://cursos.mcielectronics.cl/2022/08/23/serial-peripheral-interface-spi/> >[consulta: 11 julio 2024]

Aprendiendo Arduino. [En línea]

<<https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/> >[consulta: 11 julio 2024]

INTRODUCCIÓN A LOS MICROCONTROLADORES. [En línea]

<<https://docplayer.es/15506145-Departamento-electronica-pic-tema-1-introduccion-a-los-microcontroladores.html> >[consulta: 11 julio 2024]

Qué Es La Memoria PROM. [En línea]

<<https://es.scribd.com/document/590479932/Que-es-la-memoria-PROM> >[consulta: 11 julio 2024]

Protocolo serial y su teoría. [En línea]

<<https://hetpro-store.com/TUTORIALES/puerto-serial/> >[consulta: 11 julio 2024]

Why is a lower switching frequency more efficient? [En línea]

<<https://electronics.stackexchange.com/questions/38189/why-is-a-lower-switching-frequency-more-efficient> >[consulta: 11 julio 2024]

Página de Olimex. [En línea]

<<https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO-OTG/> >[consulta: 11 julio 2024]

TIP31 Datasheet. [En línea]
<<https://www.alldatasheet.com/datasheet-pdf/pdf/2776/MOSPEC/TIP31.html>>[consulta: julio 2024] 11

ANEXOProgramasIde Pingüino reciver test

```

#include <SoftwareSerial.h>           // Librería Software serial
#include <XBee.h>                     //Librería compatibilidad modulo XBEE
SoftwareSerial Xbee                  //Se declara los pines del XBEE
Int data=0                           //Se declara 0 como entero

void setuo() {
    Serial.Begin(9600);               //velocidad en baudios de comunicación serie
    Xbee.begin(9600);                 //comunicación serie en baudios del XBEE
    pinMode(7, OUTPUT);              //Se declara el pin 7 Salida
}

Void loop () {
    While (Xbee.available())          //acción que se tomara al conectar XBEE
        data= Xbee.read()             //Data es el valor que envió el XBEE
        Serial.print(data)            //Imprimirá el valor de data
    }

    If (data == 1)                    // si data es igual a 1 acción
        digitalWrite(7, HIGH);        //salida 7 activada
    }

    if(data ==0 )                     //si data es igual a 0 acción
        digitalWrite(7, LOW);         //salida 7 apagada
    }
}

```

Proyecto control de motor MPIDE e IDPINGUINO

```

const int Salida=10; //start           //se define pin 10 como salida de Start
const int Salida2=11;//cambio          //se define pin 11 como salida cambio de giro
const int Salida3=12;//stop            //se define pin 12 inútil "se me olvido eliminarlo"
const int BOTONS=7;//start val         //se define pin 7 como entrada botón de inicio
const int BOTONC=6;//cambio val2       //se define pin 6 como botón cambio de giro
const int BOTONT=5;//Stop val3         //se define pin 5 como parada del motor
int val;                               //val como entero
int val2;                              //val2 como entero
int val3;                              //val 3 como entero
int contar1=0;                         //contar1 es un entero y tiene valor 0
int contar2=0;                         //contar2 es un entero y tiene valor 0

void setup(){
  pinMode(Salida,OUTPUT);               //salida definida como salida
  pinMode(Salida2,OUTPUT);              //salida2 definida como salida
  pinMode(Salida3,OUTPUT);              //salida3 definida como salida
  pinMode(BOTONS,INPUT);                 //BOTONS definida como entrada
  pinMode(BOTONC,INPUT);                 //BOTONC definida como entrada
  pinMode(BOTONT,INPUT);                 //BOTONT definida como entrada
}

void loop(){
  val=digitalRead(BOTONS);               //val es igual a la lectura de BOTONS
  val2=digitalRead(BOTONC);              //val2 es igual a la lectura de BOTONC
  val3=digitalRead(BOTONT);              //val3 es igual a la lectura de BOTONT

  if (val==HIGH && val2==LOW && val3==LOW && contar1<=7){
    digitalWrite(Salida,LOW);            //salida apagada
    digitalWrite(Salida2,LOW);           //salida apagada
    digitalWrite(Salida3,LOW);           //salida apagada
    delay(1000);                          //refresco cada 1 segundo
    contar1++;                             //contar1 cuenta hasta el condicionante de ser menor que 7
  }
}

```

```

    delay(1000);
}          //if condicional que inicia el temporizador de inicio

if (val==LOW && val2==HIGH && val3==LOW && contar2<=7){
    digitalWrite(Salida,LOW);      //salida apagada
    digitalWrite(Salida2,LOW);     //salida apagada
    digitalWrite(Salida3,LOW);     //salida apagada
    delay(1000);                  //delay de 1 segundo
    contar2++;                    //contador 2 inicia hasta ser menor que 7
    delay(1000);

                                //if condicional para cambio de giro igual que el anterior
}

if (val==HIGH && val2==LOW && val3==LOW && contar1>=8){
    contar2=0;                    //al iniciar el motor loquea el contador 2 en 0
    digitalWrite(Salida,HIGH);     //salida del motor encendida
    digitalWrite(Salida2,LOW);     //salida del motor invertido apagada
    digitalWrite(Salida3,LOW);     //Salida apagada
    delay(1000);

                                //if condicional si contador1 es mayor a 7 inicia el motor
}

if (val==LOW && val2==HIGH && val3==LOW && contar2>=8){
    contar1=0;                    //contador1 bloqueado a 0
    digitalWrite(Salida,LOW);     //salida motor apagada
    digitalWrite(Salida2,HIGH);   //salida motor invertido encendida
    digitalWrite(Salida3,LOW);    //salida apagada
    delay(1000);
}          //if condicional para el arranque inverso después de 8s contador2

if (val==LOW && val2==LOW && val3==HIGH ){
    contar1=0;                    // bloqueo de contador 1 en 0
    contar2=0;                    //bloqueo de contador 2 en 0
    digitalWrite(Salida,LOW);     //motor apagado
    digitalWrite(Salida2,LOW);    //motor apagado
    digitalWrite(Salida3,LOW);    //salida apagada
    delay(1000);
}

```

```

}                                     //if condicional para stop del motor

if (val==HIGH && val2==LOW && val3==HIGH ){
    contar1=0;                       //contador 1 es 0
    contar2=0;                       //contador 2 es 0
    digitalWrite(Salida,LOW);        //apagado motor
    digitalWrite(Salida2,LOW);       //apagado motor inverso
    digitalWrite(Salida3,LOW);       //apagada salida
    delay(1000);
}                                     //if condicionante stop y arranque encendido
if (val==HIGH && val2==HIGH && val3==HIGH ){
    contar1=0;                       // contadores en 0 y salidas apagadas
    contar2=0;
    digitalWrite(Salida,LOW);
    digitalWrite(Salida2,LOW);
    digitalWrite(Salida3,LOW);
    delay(1000);
}                                     //if condicionante de todos los switches encendidos
if (val==LOW && val2==HIGH && val3==HIGH ){
    contar1=0;                       //contar 1 bloqueado en 0
    contar2=0;                       //contador 2 bloqueado en 0
    digitalWrite(Salida,LOW);        //salida motor apagada
    digitalWrite(Salida2,LOW);       //salida motor inverso apagada
    digitalWrite(Salida3,LOW);       //salida apagada
    delay(1000);
}                                     //if condicionante de segundo y tercer switch encendidos
if (val==HIGH && val2==HIGH && val3==LOW ){
    contar1=0;                       //contar 1 bloqueado en 0
    contar2=0;                       //contar 2 bloqueado en 0
    digitalWrite(Salida,LOW);        //salida apagada
    digitalWrite(Salida2,LOW);       //salida motor inversión apagada
    digitalWrite(Salida3,LOW);       //salida apagada
    delay(1000);
}                                     //if condicional si todos los switches están activados

```

```
if (val==LOW && val2==LOW && val3==LOW ){  
    contar1=0;                // apagado contador 1 en 0  
    contar2=0;                //apagado contador 2 en 0  
    digitalWrite(Salida,LOW); //salida motor apagada  
    digitalWrite(Salida2,LOW); //salida motor inversión apagada  
    digitalWrite(Salida3,LOW); //salida apagada  
    delay(1000);  
}                               //if condicional todo apagado  
  
}
```