PORTADA:

S k y 1 F



Integrantes

- Esteban Lautaro.
- Rubio Santiago Gabriel.
- Meabrio Lucas David.
- Flores Leandro.
- Brizuela Agustín.
- Leiva Santiago.
- Romocordoba Emiliano.
- Godoy Baldovino Jofiel

Contacto

Lautaro Esteban:

Mail: Lautyesteban14@gmail.com

Linkedin: Lautaro Sebastian Esteban

Instagram: @lauty.esteban

Santiago Rubio:

Mail: Santy201205@gmail.com

Linkedin: Santiago Rubio

Instagram: @santy_201205

Lucas Meabrio:

Mail: Meabriolucas@gmail.com

Linkedin: Lucas Meabrio

Instagram: @lucass_meab

Leandro Flores:

Mail: leandro200flores@gmail.com

Linkedin: Leandro Flores

Instagram: @lean.floresss

Agustin Brizuela:

Mail: brizuu750@gmail.com

Linkedin: Agustin Lionel Brizuela

Instagram: @agustiin.brizuela

Santiago Leiva:

Mail: Santiagoleiva745@gmail.com

Linkedin: Santiago Leiva

Instagram: @leivva.s

Emiliano Romocordoba:

Mail: Romoemiliano324@gmail.com

Linkedin:

Instagram:

Jofiel Godoy Baldovino:

Mail: Markbaldj@gmail.com

Linkedin: Marco Baldovino

Instagram: @jofiel_godoy

ÍNDICE

Descripción del Proyecto:

Variadores de Frecuencia:

Resumen del objetivo.

Lista de materiales usados.

Paneles de Control:

Descripción del Proyecto

Nuestro proyecto se enfocará en el armado y reacondicionamiento de un simulador de vuelo con el fin de recrear una experiencia inmersiva y entretenida para el que se meta dentro de la cabina.

Variadores de Frecuencia

Resumen del objetivo

El objetivo principal con el uso del variador de frecuencia es lograr una comunicación eficiente tanto con el programa de control como con los motores del simulador. Este tipo de comunicación es crucial para garantizar un funcionamiento sincronizado y preciso entre el software y el hardware. En nuestro caso, la comunicación se establece utilizando el protocolo **Modbus**, un estándar de comunicación ampliamente utilizado en sistemas industriales debido a su fiabilidad y flexibilidad.

El protocolo Modbus permite el intercambio de datos entre el variador de frecuencia y el microcontrolador **ESP32**, facilitando el control de parámetros clave como la velocidad y la dirección de los motores. Esta programación la estamos llevando a cabo mediante la plataforma **Arduino IDE**, donde se desarrollan los códigos que gestionan la interacción entre los distintos componentes del sistema. A través de Arduino IDE, se ha configurado el protocolo Modbus para que las señales enviadas desde el programa de simulación sean interpretadas correctamente por el variador de frecuencia, y a su vez, estas se traduzcan en comandos que controlan el movimiento de los motores.

La integración exitosa de este sistema es fundamental para garantizar que la simulación de vuelo sea lo más realista posible, permitiendo que los motores respondan con precisión a las condiciones simuladas, como cambios de altitud, aceleración y maniobras.

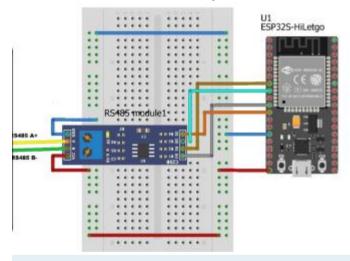
Lista de materiales usados

Usamos los siguientes materiales para este proceso actualmente:

- Módulo de RS485
- ESP32 de 30 pines
- MPU6050
- Fuente calibrada a 5V
- 8 cables macho-macho
- 2 protoboards
- Cable de datos

Comunicaciones

Comunicación entre el ESP32 y el RS485:

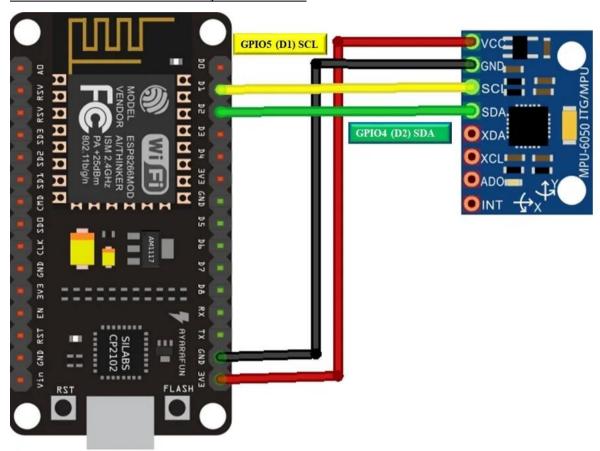


- MAX485 TTL a RS485 >> VCC >> +5V de ESP32
- MAX485 TTL A RS485 >> GND >> GND de ESP32
- MAX485 TTL a RS485 >> RO >> GPIO26 de ESP32 (SoftwareSerial RX)
- MAX485 TTL a RS485 >> DI >> GPIO27 de ESP32 (SoftwareSerial TX)
- 2. GPI026 del ESP32 se conecta al pin R0 (Recepción) del módulo receptor RS485.
- 3. GPIO27 del ESP32 se conecta al pin DI (Transmisión) del módulo transmisor RS485.

Comunicación entre el RS485 y el Variador de frecuencia (GK500):

- o Pin A (A+) del módulo RS485: Conéctalo al terminal U (fase U) del variador de frecuencia.
- o Pin B (B-) del módulo RS485: Conéctalo al terminal V (fase V) del variador de frecuencia.
 - Pin A va conectado al terminal "485+"
- Pin B va conectado al terminal "485-"

Comunicación entre el ESP32 y el MPU6050:



- o Conecta el pin SCL del MPU6050 al pin D1 (GPI08) del ESP32.
- o Conecta el pin SDA del MPU6050 al pin D2 (GPI09) del ESP32.
- Conecta el pin VCC del MPU6050 a 3.3V del ESP32.
- Conecta el pin GND del MPU6050 a GND del ESP32.

ESP32

De 38 pines:

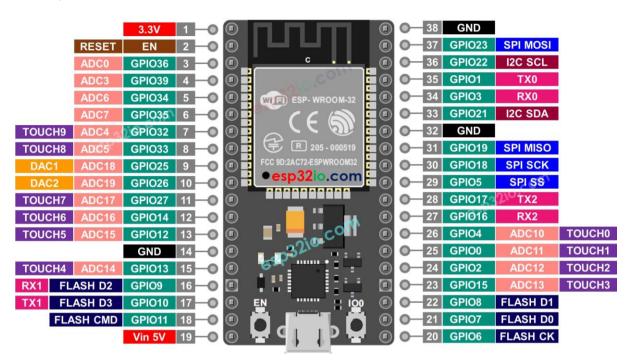


Table 2-1. Pin Overview

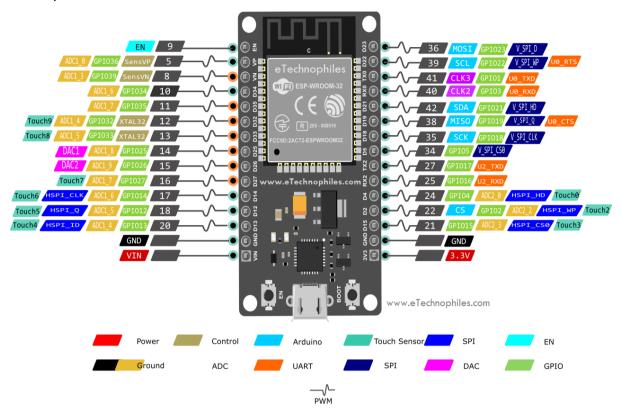
Name	No.	Туре	Function				
						Analog	
VDDA	1	Р	Analog po	ower supply (2	.3 V ~ 3.6 V)		
LNA_IN	2	1/0	RF input a	and output			
VDD3P3	3	Р	Analog po	ower supply (2	.3 V ~ 3.6 V)		
VDD3P3	4	Р	Analog po	ower supply (2	.3 V ~ 3.6 V)		
					VD	D3P3_RTC	
SENSOR_VP	5	- 1	GPI036,	ADC1_CHO,	RTC_GPI00		
SENSOR_CAPP	6	- 1	GPIO37,	ADC1_CH1,	RTC_GPI01		
SENSOR_CAPN	7	- 1	GPI038,	ADC1_CH2,	RTC_GPI02		
SENSOR_VN	8	- 1	GPI039,	ADC1_CH3,	RTC_GPI03		
			High: On;	enables the c	hip		
CHIP_PU	9	1	Low: Off;	the chip shuts	down		
			Note: Do	not leave the	CHIP_PU pin floa	ating.	
VDET_1	10	- 1	GPI034,	ADC1_CH6,	RTC_GPI04		
VDET_2	11	- 1	GPI035,	ADC1_CH7,	RTC_GPI05		
32K_XP	12	1/0	GPI032,	ADC1_CH4,	RTC_GPI09,	TOUCH9,	32K_XP (32.768 kHz crystal oscillator input)
32K_XN	13	1/0	GPIO33,	ADC1_CH5,	RTC_GPIO8,	TOUCH8,	32K_XN (32.768 kHz crystal oscillator output)
GPIO25	14	1/0	GPI025,	ADC2_CH8,	RTC_GPI06,	DAC_1,	EMAC_RXDO
GPI026	15	1/0	GPI026,	ADC2_CH9,	RTC_GPIO7,	DAC_2,	EMAC_RXD1
GPI027	16	1/0	GPI027,	ADC2_CH7,	RTC_GPIO17,	TOUCH7,	EMAC_RX_DV
MTMS	17	1/0	GPIO14.	ADC2 CH6.	RTC GPIO16.	TOUCH6.	EMAC TXD2. HSPICLK, HS2 CLK, SD CLK, MTMS

İ	MTDI	18	1/0	GPI012,	ADC2_CH5,	RTC_GPI015,	TOUCH5,	EMAC_TXD3,	HSPIQ,	HS2_DATA2,	SD_DATA2,	MTDI
	VDD3P3_RTC	19	Р	Input pov	wer supply for R	TC IO (2.3 V \sim 3	3.6 V)					
	MTCK	20	1/0	GPI013,	ADC2_CH4,	RTC_GPI014,	TOUCH4,	EMAC_RX_ER,	HSPID,	HS2_DATA3,	SD_DATA3,	MTCK
Γ	MTDO	21	1/0	GPI015,	ADC2_CH3,	RTC_GPI013,	TOUCH3,	EMAC_RXD3,	HSPICSO,	HS2_CMD,	SD_CMD,	MTDO

Mama	Nie	T	Franchis :							
Name	No.	Туре	Function							
GPI02	22	1/0	GPIO2,	ADC2_CH2,	RTC_GPI012,	TOUCH2,		HSPIWP,	HS2_DATAO,	SD_DATA0
GPI00	23	1/0	GPIOO,	ADC2_CH1,	RTC_GPIO11,	TOUCH1,	EMAC_TX_CLK,			
GPIO4	24	1/0	GPIO4,	ADC2_CHO,		TOUCHO,	EMAC_TX_ER,	HSPIHD,	HS2_DATA1,	SD_DATA1
VDD_SDIO										
GPIO16	25	1/0	GPI016,	HS1_DATA4,	U2RXD,	EMAC_CLK_				
VDD_SDIO	26	Р		ower supply: 1.8						
GPIO17	27	1/0	GPIO17,	HS1_DATA5,	U2TXD,	EMAC_CLK_	OUT_180			
SD_DATA_2	28	1/0	GPIO9,	HS1_DATA2,	U1RXD,	SD_DATA2,	SPIHD			
SD_DATA_3	29	1/0	GPI010,	HS1_DATA3,	U1TXD,	SD_DATA3,	SPIWP			
SD_CMD	30	1/0	GPIO11,	HS1_CMD,	U1RTS,	SD_CMD,	SPICS0			
SD_CLK	31	1/0	GPI06,	HS1_CLK,	U1CTS,	SD_CLK,	SPICLK			
SD_DATA_0	32	1/0	GPIO7,	HS1_DATAO,	U2RTS,	SD_DATAO,	SPIQ			
SD_DATA_1	33	1/0	GPIO8,	HS1_DATA1,	U2CTS,	SD_DATA1,	SPID			
					VD	D3P3_CPU				
GPI05	34	1/0	GPIO5,	HS1_DATA6,	VSPICSO,	EMAC_RX_C	CLK			
GPIO18	35	1/0	GPI018,	HS1_DATA7,	VSPICLK					
GPIO23	36	1/0	GPI023,	HS1_STROBE,	VSPID					
VDD3P3_CPU	37	Р	Input pov	ver supply for C	PU IO (1.8 V ~ 3	3.6 V)				
GPIO19	38	1/0	GPI019,	UOCTS,	VSPIQ,	EMAC_TXDC)			
GPI022	39	1/0	GPI022,	UORTS,	VSPIWP,	EMAC_TXD1				
UORXD	40	1/0	GPIO3,	UORXD,	CLK_OUT2					
UOTXD	41	1/0	GPIO1,	UOTXD,	CLK_OUT3,	EMAC_RXD2	2			
GPIO21	42	1/0	GPI021,		VSPIHD,	EMAC_TX_E	N			
Analog										
VDDA	43	Р	Analog po	ower supply (2.3	3 V ~ 3.6 V)					
XTAL_N	44	0	External of	crystal output						
XTAL_P	45	1		crystal input						
VDDA	46	Р	Analog po	ower supply (2.3	3 V ~ 3.6 V)					
			U 1							

Name	No.	Туре	Function
CAP1	48	- 1	Connects to a 10 nF series capacitor to ground
GND	49	Р	Ground

De 30 pines:



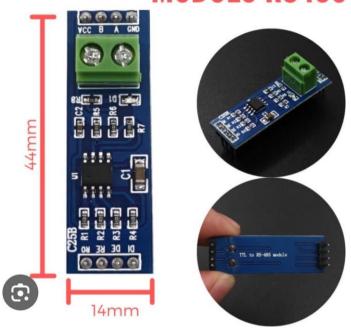
Precio:



\$ 9.990 en 6 cuotas de \$ 2.370¹³ Ver los medios de pago

RS485

MODULO RS485



conexión en modo emisor es la siguiente



conexión en modo receptor es la siguiente



Si queremos que durante la conexión el conversor RS485 pueda cambiar su papel de emisor a receptor (conexión half duplex) simplemente tenemos que conectar los pines RE y DE a una salida digital para poder cambiar su tensión de Gnd a Vcc.





MPU6050

El MPU6050 es una unidad de medición inercial o IMU (Inertial Measurment Units) de 6 grados de libertad, pues combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Este sensor es muy utilizado en navegación, geometría, estabilización, etc. Este componente nos sirve para inclinar la cabina en sus 3 ejes, basándose en las mediciones de los SimVars.



Acelerometro 3 Ejes Mpu6050 Giroscopio Para Arduino Emakers



\$ 3.730

Ver los medios de pago

Listado de parámetros que pusieron en el variador

Parámetros	Designación	Rango	Rango puesto
b0-01	Fuente de Mando de Frecuencia Maestra	0: Configuración Digital (b0-02) + ^ /v Ajuste en Panel de Control 1: Configuración Digital (b0-02) + Ajuste por Bornera UP/DOWN 2: Entrada Analógica AI 3: Potenciómetro 6: Salida de PID 8: Comando Multivelocidad 9: Comunicación	9 9: Comunicación
b0-02	Configuración Digital de Frecuencia Maestra	Límite Inferior ~ Límite Superior de Frecuencia	50,00 Hz
b1-00	Comando RUN	0: Control por Panel de Control 1: Control por Bornera 2: Control por Comunicación	2 2: Control por Comunicación
b1-01	Enlace de comando run y configuración frecuencia	Unidades: Fuente de configuración frecuencia agrupada bajo control panel de control:	AAA A: Entrada de Comunicación

Parámetros	Designación	Rango	Rango puesto
		0: Sin enlace 1: Configuración Digital (b0-02) +	
H0-01	Configuración del Puerto de Comunicación RS-485	Unidades: Velocidad en Baudios 0: 4800 bps 1: 9600 bps 2: 19200 bps 3: 38400 bps 4: 57600 bps Decenas: Formato de datos 0: Formato 1-8-2-N, RTU 1: Formato 1-8-1-E, RTU 2: Formato 1-8-1-O, RTU 3: Formato 1-7-1-E, ASCII 4: Formato 1-7-1-E, ASCII 5: Formato 1-7-1-O, ASCII Centenas: Tipo de conexión 0: Conexión cable directo (232/485) 1: MODEM (232) Unidades de Mil: Almacenamiento 0: Sin almacenar ante Pérdida de Energía 1: Almacenado ante Pérdida de Energía	0001 1: 9600bps
H0-05	Opción Maestro/Esclavo	Se usa independientemente Como Maestro Como Esclavo	2 2: Como Esclavo

Paneles de Control

El simulador posee dos paneles que cumplen su respectiva función; el panel de control (mediante palancas y botones interactuando con el entorno virtual) y el panel de instrumentos (estos se van a mostrar mediante una tele que transmite los datos reales de vuelo en los instrumentos)

El panel de control cuenta con todo el Hardware que se comunicará con el software a utilizar, (que en nuestro caso sería el Flight Simulator 2020).

Panel de Control

En el panel de control hay 7 interruptores de palanca, 8 pulsadores, una llave selectora rotativa de 6 estados y 4 interruptores divididos en 2 componentes que tienen 2 interruptores cada uno.

Para que estos componentes se accionen en simultáneo al vuelo que se haga en el Flight Simulator 2020, a través de una herramienta llamada "MobiFlight" y dos Arduino UNO. Con el MobiFlight asignamos a cada componente a que realice determinada acción dentro del FS2020. Y con la Arduino UNO, en conjunto al MobiFlight, ejecuta esa acción a realizar A continuación, explicaremos cuáles son estos componentes a utilizar y cuál es su función dentro del entorno virtual del FS2020:

<u>Interruptores de palanca:</u>

• Bomba de combustible:



Activa la bomba de combustible eléctrica, para cebar el motor antes de arrancar. Esto también proporciona un sustituto para la bomba mecánica en caso de avería.

• Luces:



- BCN: Enciende y apaga la luz intermitente roja que se encuentra en la parte superior o inferior del fuselaje. Se usa para indicar que el avión está en movimiento o que los motores están en marcha. Es buena práctica encenderla antes de arrancar el motor y dejarla encendida.
- LAND: Controla la luz de aterrizaje, que es una luz blanca muy brillante ubicada en el tren de aterrizaje o el ala. Se utiliza durante el despegue y aterrizaje para iluminar la pista y aumentar la visibilidad del avión. También puede utilizarse en vuelo bajo cuando se vuela cerca de aeropuertos o en áreas congestionadas, para aumentar la visibilidad del avión frente a otros pilotos.

- TAXI: Controla la luz de rodaje, una luz menos intensa que la luz de aterrizaje, diseñada para iluminar el área frente al avión durante el taxi en el suelo. Controla la luz de rodaje, una luz menos intensa que la luz de aterrizaje, diseñada para iluminar el área frente al avión durante el taxi en el suelo.
- NAV: Controla las luces de navegación, que son las luces rojas, verdes y blancas colocadas en las alas y la cola. Controla las luces de navegación, que son las luces rojas, verdes y blancas colocadas en las alas y la cola.
- STRB: Controla las luces estroboscópicas, que son luces blancas intermitentes de alta intensidad en las puntas de las alas. Controla las luces estroboscópicas, que son luces blancas intermitentes de alta intensidad en las puntas de las alas.
- Tubo Pitot:



Al activarse este interruptor, se habilita el calentador del tubo Pitot.

Interruptores:

Interruptor Maestro del Alternador y Batería:



Si se activa el interruptor del alternador, el alternador suministrará energía a los sistemas eléctricos cuando el motor esté en funcionamiento y carga la batería.

Si el interruptor de BAT está en ON, la batería proporcionará energía a los sistemas eléctricos cuando el motor no esté en marcha.

Hay que recalcar que estos dos interruptores suelen funcionar juntos.

• Interruptores de Bus Aviónico:



Estos interruptores accionan los buses eléctricos que alimentan la aviónica.

- BUS1 activa: Comm Panel, G530 GPS, Transponder, Autopilot, VOR1 / ILS y BUS2
- BUS2 activa: G430 GPS, VOR2, ADF y VOR2

Magneto (con llave selectora rotativa de 6 estados):



Al accionar el magneto, este proveerá corriente eléctrica a las bujías, las que a su vez producen la chispa necesaria para encender la mezcla de combustible y aire en los cilindros del motor.

Estructura:

Estos componentes irán colocados en una madera rígida pero liviana, la cuál ha sido medido con el fin de mejorar la estética, y esta medida sería de 500mmx140/120mm. Ya cortadas las medidas se colocarán los interruptores y demás componentes. Luego en la parte final se pintará la madera con un aerosol negro y se pondrán letras en blanco que formen los nombres de cada interruptor, tal como en una cabina real del CESSNA 172 para dar más realismo.

Panel de Visualización de Instrumentos

Con el fin de generar una experiencia realista de vuelo, implementamos un Panel de Instrumentos virtual, el cuál por medio de una pantalla de 18 pulgadas transmitiremos los datos reales de cada instrumento de vuelo. Por lo que para transmitir esos datos o también llamados "Variables", se encargan los programadores del equipo: Santiago Rubio y Lucas Meabrio.

Para darle una mejor estética al panel de instrumentos y que no sea solo una televisión mostrando los instrumentos, decidimos usar una placa de plástico con los agujeros de los instrumentos del avión usada por el grupo anterior del simulador (AVIS), la cuál para la fecha de la exposición será forrada para mejorar la estética del panel en conjunto a el otro panel.

Una parte importante del Panel de Instrumentos es toda la labor de *Personalización de la ubicación de los instrumentos en la pantalla*, ya que mediante herramientas como MobiFlight o Air Manager se puede llevar a cabo la personalización. Nosotros elegimos usar MobiFlight y un complemento suyo llamado "FSUIPC7" para FS2020, ya que AirManager cuesta 30 dólares actualmente.

MATERIALES:

- 1 Madera para el panel de control
- 1 Placa de plástico con los agujeros para cada instrumento
- 2 Arduino UNO
- 40 cables macho-macho aprox.
- 1 llave selectora rotativa de 6 estados
- 7 interruptores de palanca
- 8 pulsadores PUL-ST030-N
- 4 interruptores/switches
- 1 Televisión de 18 pulgadas
- 1 Aerosol de color negro/balde de pintura color negro
- 1 Pack de letras transferibles color blanco
- 2 SIMPLE FILA CONTACTO DOBLE PASO PB20S

Gastos:

- 40 cables macho-macho aprox: 5000\$
- 1 llave selectora rotativa de 6 estados: 3800\$
- 8 interruptores de palanca: 9714\$ (1214\$ c/u)
- 4 interruptores/switches: 4000\$
- 1 Pack de letras transferibles color blanco: 11400\$
- 2 filas de contacto doble paso: 1435\$ (717\$ c/u)

TOTAL: 35.340\$

