

Universidad Católica de Santa María

Facultad de Ciencias e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería Electrónica



“DISEÑO Y CONSTRUCCIÓN DE UN CUADRICOPTERO PARA EL MONITOREO DE NIVELES DE CONTAMINACIÓN ACÚSTICA Y PARÁMETROS FÍSICOS AMBIENTALES EN LA CIUDAD DE AREQUIPA”

Tesis presentada por el Bachiller:

Gil Salas Eduardo Moisés

Para optar el Título Profesional de:

Ingeniero Electrónico

Asesor: **MSc. Ing. Sulla Torres, Raúl**

AREQUIPA-PERU

2018

UNIVERSIDAD CATOLICA DE SANTA MARIA
FACULTAD DE INGENIERIA ELECTRONICA



ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

DICTAMEN DE BORRADOR DE TESIS

Los que suscriben:

Iny. Raul Salla Torres
Iny. Juan Quiroz Janga

Jurado Dictaminador del Borrador de Tesis titulado:

Diseño y construcción de un computador para el
monitoreo de niveles de contaminación acústica y
parámetros físicos ambientales en la ciudad de Arequipa

Nombrados con Decreto N°: 015-DEPIT-2017, de fecha: 18-12-2017

Presentado por el (la) los Bachiller (es):

1. Carl Salas Eduardo Moser
2. _____

cumple con emitir su dictamen:

- No registra observaciones, es **procedente**
- Registra observaciones que deben ser subsanadas, según hoja adjunta

Arequipa, 2017 Diciembre 23

Iny. Raul Salla Torres
Código Nro.: 1764

Iny. Juan Quiroz Janga
Código Nro.: 1691

DEDICATORIA

A Dios

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A la memoria de mi madre Benigna Begazo y mi tío Manuel Begazo.

A mis padres Luis Y Rosa por todo el amor y el cariño que me dan, por el apoyo y la motivación de seguir adelante y no rendirme ante cualquier adversidad.

A mi hermana Patricia por estar conmigo, apoyarme siempre y darme los ánimos para culminar este proyecto.

AGRADECIMIENTOS

A mi hermana Patricia que siempre me apoyo y me acompaño en todo momento, animándome siempre para cumplir todos mis objetivos.



ÍNDICE

RESUMEN	XVIII
ABSTRACT	XIX
NOMENCLATURA	XX
CAPITULO I	
PALNETAMIENTO DEL PROBLEMA	01
1.1.- IDENTIFICACION DEL PROBLEMA	01
1.2.- DESCRIPCION DEL PROBLEMA	01
1.3.- JUSTIFICACION	02
1.4.- OBJETIVOS.....	03
1.4.1.-GENERAL	03
1.4.2.-ESPECIFICOS.....	03
1.5.- APORTACIONES	03
1.6.- ESTRUCTURA DE LA TESIS.....	04
CAPITULO II	
MARCO TEORICO.....	06
2.1.- VEHICULO AEREO NO TRIPULADO (UAV).....	06
2.2.- RESEÑA HISTORICA.....	07
2.3.- CLASIFICACION DE VEHICULOS AEREOS NO TRIPULADOS.....	08
2.4.- VEHICULO AEREO NO TRIPULADO MULTIROTOR.....	11
2.5.- COMPONENTES QUE INTEGRAN UN UAV	12
2.5.1.-MOTOR BRUSHLESS.....	12
2.5.2.-CONTROLADOR ELECTRONICO DE VELOCIDAD (ESC).....	14
2.5.3.-UNIDAD DE MEDICION INERCIAL (IMU).....	15
2.5.4.-CONTROLADORES DE VUELO	15
2.5.4.1PIXHAWK	16
2.5.4.2.-DJI-NAZA M-LITE.....	18
2.5.4.3.-ARDUPILOT MEGA (APM)	18
2.5.4.4.-SPARKY 2.0	19
2.5.5.-SISTEMAS DE CONTROL POR RADIOFRECUENCIA.....	20
2.5.6. SISTEMAS DE POSICIONAMIENTO GLOBAL (GPS).....	22

2.5.7.-ESTACION DE CONTROL EN TIERRA (GCS).....	23
2.5.8.-BATERIAS DE POLIMERO DE LITIO	24
2.6.- INTERNET DE LAS COSAS.....	26
2.7.- PROTOCOLOS DE COMUNICACIÓN PARA INTERNET DE LAS COSAS	27
2.7.1.-HTTP	27
2.7.2.- MQTT	28
2.7.3.-CoAP	30
2.8.- PLATAFORMAS DE DESARROLLO PARA EL INTERNET DE LAS COSAS	30
2.8.1.-NODE RED.....	30
2.8.2.- ECLIPSE IoT	31
2.8.3.-THINGSPEAK.....	32
2.8.4.-THINGERIO.....	33
2.9.- TARJETAS DE ADQUISICION DE DATOS	34
2.10.- BASES DE DATOS	35
2.11.- COMUNICACIONES.....	36
2.12.- CONTAMINACION ACUSTICA	37
2.12.1.-PRESION SONORA.....	37
2.12.2.-CONTAMINACION ACUSTICA EN AREQUIPA.....	38
2.13.- PARAMETROS FISICOS AMBIENTALES	40
2.13.1.-TEMPERATURA.....	40
2.13.2.-HUMEDAD	41
2.13.3.-PRESION	42
2.13.4.-INTENSIDAD LUMINOSIDAD.....	43
2.13.5.- RADIACION ULTRAVIOLETA.....	43
CAPITULO III	
DISEÑO DEL SISTEMA DE MONITOREO.....	45
3.1.-DIAGRAMA DE BLOQUES GENERAL.....	46
3.2 DESCRIPCION DE LOS DISPOSITIVOS UTILIZADOS.....	47
3.2.1.-SENSOR DE TEMPERATURA DHT22	47
3.2.2.-SENSOR DE PRESION BMP180.....	47
3.2.3.-SENSOR DE LUMINOSIDAD BH1750.....	48

3.2.4.-SENSOR DE RADIACION UV	49
3.2.5.-SENSOR DE DIOXIDO DE CARBONO MQ-135.....	50
3.2.6.-SENSOR DE MONOXIDO DE CARBONO MQ-7	51
3.2.7.-SENSOR DE SONIDO	52
3.2.7.-MODULO EXPANSOR ADS1115.....	52
3.3.-NODEMCU ESP8266.....	54
3.3.1.-CARACTERISTICAS	54
3.3.2.-DISTRIBUCION DE PINES	55
3.4.-NODE JS	55
3.5.-NODE-RED	56
3.6.-FIREBASE.....	57
3.7.-INSTALACION DE NODE RED	59
3.8.-INSTALACION DE MOSCA MQTT.....	61
3.9.-INSTALACION DE BASE DE DATOS FIREBASE	63
3.10.-DIAGRAMA DE BLOQUES DEL SISTEMA DE MONITOREO.....	64
3.11.-DESCRIPCION DEL DIAGRAMA DE BLOQUES.....	64
3.11.1.-Sensores	64
3.11.2.-NodeMCU.....	64
3.11.3.-MQTT Mosca.....	64
3.11.4.-Node Red	64
3.11.5.-Base de datos.....	64
3.12.-PRINCIPIO DE FUNCIONAMIENTO SENSOR DHT22	65
3.13.-DIAGRAMA DE FLUJOS TEMPERATURA Y HUMEDAD	66
3.14.-RUTINA DE MEDICION DE TEMPERATURA Y HUMEDAD	67
3.15.-MUESTRA DE DATOS POR MONITOR SERIE.....	68
3.16.-DIAGRAMA DE FLUJOS PRESION BAROMETRICA	70
3.17. PRINCIPIO DE FUNCIONAMIENTO	71
3.17.1.-COEFICIENTES DE CALIBRACION	71
3.17.2.-CALCULO DE ALTITUD ABSOLUTA.....	71
3.17.3.-CALCULO DE PRESION A NIVEL DE MAR	72
3.18.-RUTINA DE MEDICION DE PRESION Y ALTITUD	73
3.19.- MUESTRA DE DATOS POR MONITOR SERIE.....	74

3.20.-DIAGRAMA DE FLUJOS MEDICION DE LUMINOSIDAD.....	75
3.21.- RUTINA DE ADQUISION DE LUMINOSIDAD.....	76
3.22.- MUESTRA DE DATOS POR MONITOR SERIAL.....	76
3.23.- PRINCIPIO DE FUNCIONAMIENTO SENSOR UV	77
3.24.-DIAGRAMA DE FLUJO MEDICION DE RADIACION UV	80
3.25.- RUTINA DE MEDICION DE RADIACION UV	81
3.26.-DIAGRAMA DE FLUJOS MEDICION DE NIVEL DE PRESION SONORA EN DB	83
3.27.- LIMITES PERMISIBLES DE CONTAMINACION ACUSTICA.....	84
3.28.- RUTINA DE MEDICION.....	85
3.29.- VISUALIZACION DE DATOS POR MONITOR SERIE	86
3.30.- GRAFICA DE DB CON RESPECTO AL TIEMPO	86
3.31.-DIAGRAMA DE FLUJO MEDICION DE MONOXIDO DE CARBONO	87
3.32.- PRINCIPIO DE FUNCIONAMIENTO.....	88
3.33.-LIMITES PERMISIBLES DE CO EN EL AIRE	89
3.34.- RUTINA DE MEDICION.....	90
3.35.- VISUALIZACION DE DATOS POR MONITOR SERIE	91
3.36.-DIAGRAMA DE FLUJO MEDICION DE DIOXIDO DE CARBONO	92
3.37.- CURVA CARACTERISTICA DE SENSOR MQ135	93
3.38.- RUTINA DE MEDICION DE CO2	94
3.39.-DATOS OBTENIDOS DE LA RUTINA DE MEDICION.....	95
3.40.-DIAGRAMA DE FLUJOS PROGRAMA GENERAL	96
3.41.- DESARROLLO DE DASHBOARD PARA VISUALIZACION EN TIEMPO REAL EN NODE RED.....	99
3.41.1 CONFIGURACION PARA LA COMUNICACIÓN ENTRE NODE RED Y NODEMCU.....	99
3.41.2.-INTEGRACION DE BASE DE DATOS EN NODE RED	108
3.41.2.1 CREACION DE BASE DE DATOS	108
3.41.2.2.-RUTINA PARA HACER PÚBLICA BASE DE DATOS	111
3.41.3.-ELABORACION DE PLACA PARA EL SISTEMA DE MONITOREO.....	113

CAPITULO IV

MODELAMIENTO Y CONSTRUCCION.....	116
4.1 MODELAMIENTO MATEMATICO DEL SISTEMA	116
4.1.1.-MODELO DEL UAV.....	116
4.1.2.-OBTENCION DE ECUACIONES DE MOVIMIENTO ROTACIONAL (FORMULACION NEWTON –EULER)	122
4.2.ESTRUCTURA DE CUADRICOPTERO	124
4.2.1CHASIS DJI F450 MM.....	124
4.2.1.1.-CARACTERISTICAS	124
4.2.2 PLACA DE DISTRIBUCION DE VOLTAJE.....	125
4.2.2.1.-HELICES	126
4.2.2.2.-BATERIA LIPO C30 4000 mAh HOBBYLINE	127
4.2.2.3.-BATERIA LIPO C45 5400 mAh TIGER POWER.....	128
4.2.2.4.-MOTOR BRUSHLESS A2212/13T 1000KV	129
4.2.2.5.-OBTENCION DE GRAFICAS DE DESEMPEÑO	130
4.2.2.6.-GRAFICAS OBTENIDAS PARA SELECCION DE HELICES ...	131
4.2.2.7.-CALCULO DE DURACION DE BATERIA HOBBYLINE C30 4000 mAh	133
4.2.2.8.-CALCULO DE DURACION DE BATERIA TIGER POWER C45 5400 mAh	134
4.2.3.-PROCESO DE EMSAMBLADO (PARTE HARDWARE)	135
4.2.3.1.-ARMADO DE CHASIS.....	135
4.2.3.2.-CONEXIÓN DE VARIADORES A PLACA DISTRIBUIDORA DE VOLTAJE.....	135
4.2.3.3.-MONTAJE DE LOS MOTORES A2212/13T	136
4.2.3.4.-MONTAJE DE VARIADORES ESC 30 AMPERIOS	137
4.2.3.5.-MONTAJE DE BRAZOS A PLACA DISTRIBUIDORA DE VOLTAJE.....	137
4.2.3.6.-CONEXIONADO DE LOS MOTORES A LA TARJETA	138
4.2.3.7.-CONEXIONADO DE RECEPTOR FLYSKY A LA TARJETA....	139
4.2.3.8.-MONTAJE DE LA PLACA APM 2.8 Y RECEPTOR Y MODULO GPS	140

CAPITULO V

CONTROL DE CUADRICOPTERO	141
--------------------------------	-----

5.1. TARJETA APM 2.8	141
5.1.1. CARACTERISTICAS	141
5.2. DIAGRAMA DE FLUJO DE PROGRAMACION DE APM.....	142
5.3 CONFIGURACION DE TARJETA APM2.8 EN MISSION PLANNER	143
5.4. CALIBRACION DEL ACELEROMETRO.....	145
5.5. CALIBRACION DE COMPASS.....	146
5.6. CONFIGURACION DE MODOS DE VUELO APM2.8	147
5.6.1. CONFIGURACION DE MODOS DE VUELO	148
5.7. CONFIGURACION DE EMISORA FLYSKY 6 CANALES.....	150
5.8. MAPEO DE CANALES	151
5.9. CONFIGURACION DEL GEOFENCE DE ARDUPILOT	153
5.10. VERIFICACION DE CONFIURACION DE APM 2.8.....	154
5.11. DIAGRAMA DE FLUJOS CALIBRACION DE VARIADORES ESC.....	155
5.12. VERIFICACION DE IMU DE APM2.8.....	156
5.13. CONFIGURACION DE MONITOREO DE BATERIA EN ESTACION BASE	157
5.14. PID MISSION PLANNER	158
CAPITULO VI	
PRUEBAS Y RESULTADOS.....	160
6.1. RED DE NODOS FINALIZADO.....	161
6.2. SISTEMA DE MONITOREO EN NODE-RED FINALIZADO.....	162
6.3. MUESTRA DE FUNCIONAMIENTO DE BASE DE DATOS CON FIREBASE.....	163
6.4. PRIMER PROTOTIPO DE VUELO	164
6.5. PROTOTIPO DE VUELO	165
6.6. UNIDAD DE MONITOREO.....	165
6.7. PRUEBAS DE VUELO	166
6.8. PRUEBAS DE MONITOREO	172
CAPITULO VII	
CONCLUSIONES Y RECOMENDACIONES	181
CONCLUSIONES.....	182
RECOMENDACIONES	184

BIBLIOGRAFIA	185
ANEXOS	188



ÍNDICE DE FIGURAS

Figura 2.1: RQ-4 Global Hawk	07
Figura 2.2: UAV de ala fija	10
Figura 2.3: Uav multirotor	10
Figura 2.4: Movimiento de un cuadricoptero	12
Figura 2.5: Esquema de motor brushless	13
Figura 2.6: Motor Brushless a detalle	14
Figura 2.7: ESC	14
Figura 2.8: IMU.....	15
Figura 2.9: Controladores de vuelo comerciales	16
Figura 2.10: Controlador Pixhawk	17
Figura2.11: Controladora de vuelo dji-naza M-lite.....	18
Figura2.12: Controladora de vuelo ardupilot	19
Figura2.13: Controladora de vuelo sparky 2.0.....	20
Figura2.14: emisora flysky.....	21
Figura2.15: receptor flysky	21
Figura2.16: Sistema de posicionamiento global misión planner.....	24
Figura2.17: Estación de control en tierra.....	22
Figura2.18: Baterías LiPo.....	25
Figura2.19: Internet of things.....	27
Figura2.20: Protocolo http	28
Figura2.21: MQTT bróker.....	29
Figura2.22: Dashboard Node-Red	31
Figura2.23: Eclipse IoT	32
Figura2.24:Dashboard en thingspeak	33
Figura2.25: Dashboard en thingerio	34
Figura2.26: Termómetro digital con sonda	41
Figura2.27: higrómetro	41
Figura2.28: Barómetro analógico	42
Figura2.29: Luxómetro digital	43
Figura2.30: Pirómetro	44
Figura3.1: Sensor DHT22.....	47

Figura3.2: Sensor bmp180	48
Figura3.3: Sensor bh1750	49
Figura3.4: Sensor GYML8511	50
Figura3.5: Sensor de CO2	51
Figura3.6: Sensor de CO	52
Figura3.7: Sensor de sonido	52
Figura3.8: Modulo expansor adc 16 bits	53
Figura3.9: NodeMCU ESP8266	55
Figura3.10: Distribución de pines nodeMcu	55
Figura3.11: Node red	56
Figura3.12: Logo Firebase	57
Figura3.13: Firebase applications	58
Figura3.14: Descarga de node.js	59
Figura3.15: Cod. instalacion node-red	59
Figura3.16: Instalacion de node-red	60
Figura3.17: Ejecucion de node-red	60
Figura3.18: Entorno de trabajo Node-red	61
Figura3.19: Código de instalación mosca	61
Figura3.20: Instalación de mosca	62
Figura3.21: Ejecución de mosca	62
Figura3.22: manage palette.....	63
Figura3.23: Datos obtenidos de dht22	68
Figura3.24: Grafica de humedad/temperatura/temperatura en grados F	68
Figura3.25: Coeficientes de calibración bmp180.....	71
Figura3.26: Calculo de altitud absoluta	71
Figura3.27: Variación de altitud con respecto a la presión	72
Figura3.28: Calculo de presión a nivel del mar	72
Figura3.29: Datos obtenidos por bmp180	74
Figura3.30: Muestra de datos obtenidos por sensor bh1750	76
Figura3. 31: Grafica de medición de luminosidad con respecto al tiempo	77
Figura3. 32: Limites de radiación UV	77
Figura3. 33: Diagrama de bloques funcional del sensor ML8511.....	78
Figura3. 34: Relación de volt out con intensidad UV.....	78

Figura3.35: Respuesta espectral del sensor ml8511	79
Figura3.36: Muestra de datos obtenidos por el sensor ML8511.....	82
Figura3.37: Variacion de muestras de radiación uv con respecto al tiempo	82
Figura3.38: Obtención de datos de sensor de sonido	86
Figura3.39: Variación de db con respecto al tiempo	86
Figura3.40: Curva característica de sensor mq7	88
Figura3.41: Dependencia de Co con respecto a temperatura y humedad.....	89
Figura3.42: Niveles de CO mostrados en monitor serie	91
Figura3.43: Curva característica de sensor mq125	93
Figura3.44: datos obtenidos por sensor de co2	95
Figura3.45: Verificación de server (ip).....	99
Figura3.46: Asignación de server y mqtt	99
Figura3.47: Comunicación exitosa con MQTT	100
Figura3.48: Conexión con mqtt y muestra de datos por monitor serie	100
Figura3.49: Debug de temperatura,humedad,luminosidad	101
Figura3.50: Muestra de datos recibidos en node red	102
Figura3.51: elaboración de dashboard con nodos	103
Figura3.52: Depuración de datos enviados por nodeMcu	103
Figura3.53: Configuración de parámetro para gauge temperatura.....	104
Figura3.54: Configuración de parámetros grafica chart	105
Figura3.55: nodos mqtt con nodos dashboard	106
Figura3.56: parámetros listos para ser graficados	107
Figura3.57: Dashboard elaborado	107
Figura3.58: Creación de proyecto firebase	108
Figura3.59: Autenticación de cuenta	109
Figura3.52: Confirmación exitosa.....	109
Figura3.61: database firebase.....	110
Figura3.62: Código de autenticación	110
Figura3.63: Creación de variables a almacenar	111
Figura3.64: Nodos firebase integrados	112
Figura3.65: Almacenamiento en tiempo real	113

Figura3.66: Elaboración de placa de control	113
Figura3.67: Placa de control con tarjetas incluidas	114
Figura3.68: Sensores y tarjeta NodeMCU.....	114
Figura3.69: Sensores conectados NODEMCU	115
Figura4.1: Yaw, pitch, roll	116
Figura4.2: sistema en coordenadas traslacionales y rotacionales.	117
Figura4.3: Placa de distribución de voltaje	125
Figura4.4: Hélices 10x45.....	126
Figura4.5: Batería 30C ChinaHobbyLine.....	127
Figura4.6: Batería 45C Tiger Power	128
Figura4.7: motor A2212/13T.....	129
Figura4.8: Grafica de velocidad,potencia y eficiencia del motor A2212/13T.....	130
Figura4.9: Grafica de empuje y potencia del motor A2212/13T	131
Figura4.10: Configuración de perfil CLARK-Y	132
Figura4.11:Grafica de potencia ,empuje y eficiencia	132
Figura4.12: Variadores soldados a tarjeta distribuidora de voltaje	135
Figura 4.13:Montaje y sujeción de motor a brazo de cuadricoptero	136
Figura 4.14: conexionado de variadores a motores y sujeción a brazo de cuadricoptero.....	137
Figura4.15: Montaje de brazos a tarjeta distribuidora de voltaje	137
Figura4.16: sentido de conexión de variadores a tarjeta apm	138
Figura4.17: Diagrama de conexión de motores, variadores a tarjeta apm	139
Figura4.18: conexión de receptor a tarjeta apm	140
Figura4.19: Incorporación de tarjeta y modulo GPS.....	140
Figura5.1:apm2.8	141
Figura5.2: Conexión de apm 2.8 con misión planner	143
Figura5.3: Descarga del firmware mission planner	143
Figura5.4: Wizard mission planner	144
Figura5.5: Selección de frame.....	144
Figura5.6: Cargando firmware a apm 2.8.....	145
Figura5.7: Verificación de firmware instalado.....	145
Figura5.8: Procedimiento de calibración de acelerómetro	146

Figura5.9 Secuencia de calibración	146
Figura5.10: Calibración de Compas	147
Figura5.11: Compensaciones obtenidas después de la calibración.....	147
Figura5.12: Setup de canales auxiliares	149
Figura5.13: Cambio de canales auxiliares swC=canal5 swD=canal6	149
Figura5.14: Verificación de modos de vuelo.....	149
Figura5.15: Calibración de emisora flysky	150
Figura5.16: Valores Max y min de emisora	151
Figura5.17: Mapeo de canales emisora	152
Figura5.18: Valores obtenidos después de la calibración	152
Figura5.19: Configuración del Geofence.....	153
Figura5.20: Verificación de proceso de instalación culminado.....	154
Figura5.21: Verificación de pitch	156
Figura5.22: Verificación de roll	156
Figura5.23: tuning pitch and roll simultaneamente	156
Figura5.24: Configuración de monitoreo de batería	157
Figura5.25: configuración de GPS	158
Figura5.26: Setup de PID	159
Figura6.1: Red de nodos para dashboard.....	161
Figura6.2: Dashboard en node red.....	162
Figura6.3: base de datos elaborada en firebase	163
Figura6.4: primer prototipo elaborado	164
Figura6.5: Unidad de vuelo	165
Figura6.6: Unidad de monitoreo	165
Figura6.7: Equipo listo para ser probado	166
Figura6.8: Despegue del equipo	166
Figura6.9: Estabilización del equipo.....	167
Figura6.10: Equipo estabilizado	168
Figura6.11: modo de vuelo: position hold.....	169
Figura6.12: Position hold.....	169
Figura6.13: equipo estabilizado.....	170
Figura6.14: Decremento en el throttle	171

Figura6.15: Alrededores de la clínica Arequipa	172
Figura6.16: Alrededores de la Avenida Ejercito	173
Figura6.17: Puente Grau	174
Figura6.18: Alrededores del mercado el Avelino	175
Figura6.19: Mercado el Avelino	175
Figura6.20: Seguro social	177
Figura6.21: Cambios en el dashboard	179
Figura6.22: Base de datos con valores modificados	179
Figura6.23: Variación de datos en Dashboard	180

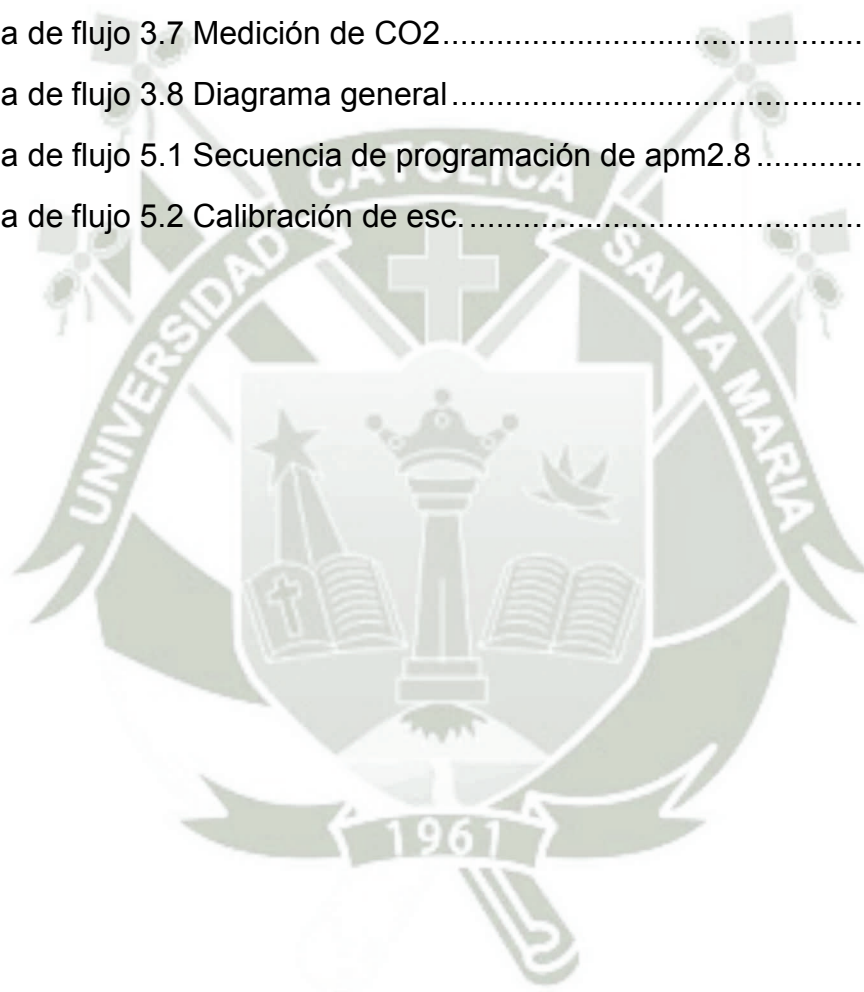


ÍNDICE DE TABLAS

Tabla 3.1: Características Sensor DHT22.....	47
Tabla 3.2: Características sensor bmp180.....	48
Tabla3.3: Características sensor bh1750.....	49
Tabla3.4: Características sensor GYML8511.....	49
Tabla3.5: Características sensor de CO2	50
Tabla3.6: Características sensor de CO	51
Tabla3.7: Características de módulo expansor ads1115	53
Tabla3.8: Características NodeMCU	54
Tabla3.9: Límites permisibles de ruido.....	84
Tabla3.10: Límites permisibles de CO	89
Tabla4.1: Características chasis dji f450.....	124
Tabla4.2: Características de hélices	126
Tabla4.3: Características batería lipo hobbyline.....	127
Tabla4.4: Características batería tiger power.....	128
Tabla4.5: Características motor brushless A2212.....	129
Tabla5.1: Características apm2.8	141
Tabla6.1: Datos obtenidos en la clínica Arequipa	172
Tabla6.2: Datos obtenidos por la avenida ejercito.....	173
Tabla6.3: Datos obtenidos por el puente Grau.....	174
Tabla6.4: Datos obtenidos por el Mercado el Avelino	176
Tabla6.5: Datos obtenidos por el seguro social.....	178

INDICE DE DIAGRAMAS DE FLUJO

Diagrama de flujo 3.1 Humedad y temperatura	66
Diagrama de flujo 3.2 Presión	70
Diagrama de flujo 3.3 Luminosidad	75
Diagrama de flujo 3.4 Radiación UV	80
Diagrama de flujo 3.5 Medición de dB.....	83
Diagrama de flujo 3.6 Medición de CO.....	87
Diagrama de flujo 3.7 Medición de CO2.....	92
Diagrama de flujo 3.8 Diagrama general.....	96
Diagrama de flujo 5.1 Secuencia de programación de apm2.8	142
Diagrama de flujo 5.2 Calibración de esc.....	135



RESUMEN

El presente trabajo de tesis tuvo como objetivo el diseño de un cuadricoptero para un sistema de monitoreo de niveles de contaminación acústica, parámetros físicos ambientales (temperatura, humedad, nivel de radiación UV, presión barométrica, niveles de luminosidad), niveles de monóxido de carbono y nivel de dióxido de carbono en la ciudad de Arequipa.

Primeramente se realizó una revisión de la bibliografía y el estado del arte de los vehículos aéreos no tripulados. Seguidamente, se procedió a detallar los requisitos esenciales para el diseño y construcción de la unidad. En la parte de control de la unidad, se realizó un estudio de los diferentes sistemas comerciales existentes en el mercado y que sea el adecuado para nuestra unidad de vuelo, llegando a la conclusión que la tarjeta Ardupilot Mega cumple con los objetivos trazados en el presente trabajo de tesis.

Se explica a detalle de la parte de toma de datos de todos los sensores utilizados, se hará uso de la tarjeta nodeMCU ESP8266, debido a que es una plataforma open-source, lo cual hace que sea muy flexible a la hora de programar esta se encargara de conectarse a una red wifi, procesara, almacenara y visualizara los valores obtenidos por los sensores.

Se explica el desarrollo de la interfaz gráfica hecha usando node-red y el protocolo MQTT para la comunicación entre la tarjeta y node-red, esta permitirá visualizar y graficar los valores obtenidos por la tarjeta nodeMCU.

Finalmente, se realizan pruebas del sistema integrado, obteniéndose un cuadricoptero con vuelo radio controlado, Ardupilot Mega cumple con las necesidades de la unidad de vuelo, consecuentemente de la parte de vuelo y monitoreo, se busca que el equipo pueda llegar a partes de la ciudad Arequipa en las que el incremento del parque automotor ha hecho que aumente indiscriminadamente la contaminación acústica y la variación exagerada de las condiciones ambientales.

Palabras claves: Nodered, internet of things, cuadricoptero, MQTT, monitoreo, sensores.

ABSTRACT

The objective of this thesis was to design a quadriopter for a system for monitoring levels of noise pollution, physical environmental parameters (temperature, humidity, UV radiation level, barometric pressure, light levels), carbon monoxide levels and carbon dioxide level in the city of Arequipa.

First, a review of the literature and the state of the art of unmanned aerial vehicles was carried out. Next, we proceeded to detail the essential requirements for the design and construction of the unit. In the control part of the unit, a study was made of the different commercial systems existing in the market and that is appropriate for our flight unit, reaching the conclusion that the Ardupilot Mega card meets the objectives set out in this thesis work.

It is explained in detail of the data collection part of all the sensors used, it will use the nodeMCU card ESP8266, because it is an open-source platform, which makes it very flexible when programming this will charge to connect to a wifi network, process, store and visualize the values obtained by the sensors.

The development of the graphic interface made using node-network and the MQTT protocol for communication between the card and node-network is explained, this will allow to visualize and graph the values obtained by the nodeMCU card.

Finally, tests of the integrated system are performed, obtaining a quadricopter with controlled radio flight, Ardupilot Mega meets the needs of the flight unit, consequently of the flight and monitoring part, it is sought that the team can reach parts of the city Arequipa, in which the increase in the number of vehicles has led to an indiscriminate increase in noise pollution and an exaggerated variation in environmental conditions.

Keywords: Node-red, internet of, things, quadcopter, MQTT, monitoring, sensors.

NOMENCLATURA

MIPS: Millones de instrucciones por segundo

IMU: Unidad de medición inercial

UAV :Unmanned Aerial Vehicle (Vehículo aéreo no tripulado)

PPM: Partes por millón

PWM: pulse-width modulation

ESC: Electronic Speed Controller

RSSI: Received Signal Strength Indicator

I2C: Inter-Integrated Circuit

SPI: Serial Peripheral Interface

LiPo: batería de polímero de iones de litio

mAh: miliamperio-hora

APM: Ardupilot mega

SRAM: Static Random Access Memory

UART: Universal Asynchronous Receiver-Transmitter

DAC: Digital-to-Analog Conversion

ADC: Analog-to-Digital Conversion

EEPROM: Erasable Programmable Read Only Memory

$R_{z,\psi}$: matriz de rotación en z

$R_{y,\theta}$: matriz de rotación en y

$R_{x,\phi}$: matriz de rotación en x

PID: Proporcional, integrativo, derivativo

ASK: Amplitude-shift keying (modulación por desplazamiento de amplitud)

OOK: On-Off Keying

TTL: transistor-transistor logic

Npm: package manager for JavaScript

ICSP: In-Circuit Serial Programming

TX: Transmisión

RX: Recepción

E_{CT} : energía cinética de traslación (x, y, z)

m = masa del cuadricoptero

ε = es la velocidad lineal en cada uno de los ejes(x, y, z)

E_{CR} : energía cinética rotacional

I = momento de inercia

η = es la velocidad angular en cada uno de los ejes

GPS: Sistema de posicionamiento global

GCS: estación de control en tierra

CO₂: Dioxido de carbono

CO: Monoxido de carbono

UV: Ultraviolet index

IoT: Internet of things

HTTP: Hypertext Transfer Protocol

HTML: HyperText Markup Language (lenguaje de marcas de hipertexto)

URL: Uniform Resource Locator (identificador de recursos uniforme)

MQTT: Message Queue Telemetry Transport

M2M: Machine to machine

CPU: central processing unit

CoAp: Constrained application protocol

IBM: International Business Machines

API: Application Programming Interface (interfaz de programación de aplicaciones)

WLAN: wireless local area network

SIL: nivel de interferencia con el lenguaje

dB: Decibelios

mPa: margen de presión audible

mb: Milibares

hPa: Hectopascales

Pa: Pascales

RH: Relative Humidity

mW: milivatios

Bit: Binary digit

Json: JavaScript Object Notation

E_p : energía potencial

ϵ : gravedad

z : viene a ser la altura que se encuentra el cuadricoptero en el espacio

F_{ε} = Fuerzas generalizadas aplicadas sobre las articulaciones(motores)

∂q_i = coordenadas generalizadas(posicion de las articulaciones)

$\partial \dot{q}_i$ = velocidades de las articulaciones

∂L = derivada parcial del lagraniano

$I_{3 \times 3}$ = matriz de inercia tiene una estructura simetrica con respecto X, Y, Z

$\dot{\omega}$ = velocidad angular

τ = torque o momentos generalizados

L = lagraniano

K = energia cinetica(tanto rotacional como traslacional)

U = energia potencial

CN: noise categorization

Admob: advertising on mobile

CDN: content delivery network

GDL: Grados de libertad

OEFA: Orgnismo de evaluación y fiscalización ambiental

ECA: Estándares de calidad ambiental

FPV: First person view

Lmp :limites maximos permisibles

ECA: Estandares de calidad ambiental

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

En este capítulo se expondrá la identificación del problema, descripción del problema, justificación, objetivos, aportaciones y la estructura de la tesis.

Se da a conocer lo empleado en la solución del problema como también los métodos y herramientas que se usaron, se indican los aportes de este trabajo y se detalla la estructura de la tesis por capítulos.

1.1.- IDENTIFICACION DEL PROBLEMA

El vertiginoso crecimiento coloca a Arequipa como la segunda ciudad después de Lima con la mayor cantidad de vehículos que ingresaron al parque automotor.

Por años viene siendo un factor determinante la salud pública de los ciudadanos, a consecuencia de los gases contaminantes, cambios bruscos en la temperatura, aumento de los niveles de radiación UV y la contaminación acústica que está presente en el día a día de los ciudadanos.

En la actualidad, encontramos que la explotación de los recursos naturales a modernizado sus técnicas, que conlleva una mayor cantidad de desechos que el medio ambiente no puede procesar, con la misma velocidad con la que se producen, ni que el hombre puede tolerar, ocasionando la contaminación del aire, agua, suelo y perturbando el bienestar del ser humano y los demás seres vivos que habitan en los ecosistemas. Esto lleva consigo la destrucción de muchos ecosistemas y por tanto una pérdida importante de la biodiversidad. También se ve amenazado el futuro de las poblaciones debido a las consecuencias que produce la contaminación.

1.2.- DESCRIPCION DEL PROBLEMA

Actualmente en la ciudad de Arequipa el parque automotor se ha incrementado exponencialmente y debido a esto, no solo los niveles de contaminación atmosférica se incrementaron sino también incremento otro

tipo de contaminación, la contaminación acústica ,está en esencia perjudica nuestro día a día ocasionando severos daños gradualmente de forma indiscriminada.

Hay diversos puntos en la ciudad en los que cuesta sensibilizar a la población, porque desafortunadamente se han acostumbrado a vivir rodeados de esta contaminación.

Decimos que el ruido y la emanación de gases provocada por los vehículos son los más perjudiciales actualmente, ya que estos causan trastornos en la salud de las personas y el medio que las rodea.

1.3.- JUSTIFICACION

La obtención de parámetros físicos ambientales es muy importante en el monitoreo ambiental, mayormente este proceso se lleva a cabo con plataformas instrumentadas denominadas estaciones meteorológicas. Actualmente en el mercado se encuentra una variedad de estaciones meteorológicas que permiten leer una serie de parámetros, sin embargo, resulta en muchos casos complicado el uso de estos dispositivos presentando otra desventaja, son estaciones fijas, eso quiere decir estarán adquiriendo datos en determinado punto.

Otro punto que juega en contra es que están limitados a monitorizar ciertos parámetros sin opción a añadir otros componentes.

El presente trabajo de tesis describe el diseño y construcción de un cuadricoptero para el monitoreo de niveles de contaminación acústica y parámetros físicos ambientales de altas prestaciones y con un reducido costo.

Lo que quiere decir es que nuestra unidad podrá transportarse a diferentes puntos de la ciudad, poder adquirir los datos censados, almacenarlos, y poder visualizarlos en una interfaz gráfica en tiempo real.

1.4.- OBJETIVOS

1.4.1.-GENERAL

- Diseñar y construir un cuadricoptero para transportar un sistema de monitoreo de niveles de contaminación acústica y parámetros físicos ambientales en la ciudad de Arequipa.

1.4.2.-ESPECIFICOS

- Aplicar conocimientos en comunicaciones, aerodinámica, medición remota e interfaz gráfica y monitoreo.
- Diseñar y construir un vehículo aéreo no tripulado multirrotor capaz de llevar un equipo de monitoreo, considerando las principales fuerzas aerodinámicas y utilizando materiales de fácil reparación, reemplazo y bajo costo.
- Aplicar fundamentos electrónicos basados en el diseño y construcción inalámbrica de un sistema de comunicación, transmisión y análisis de señales.
- Especificar detalladamente los requerimientos técnicos para un vehículo aéreo no tripulado de tipo multirrotor.
- Empleo de plataformas de código abierto (open source) basada en hardware y software flexibles, tanto para la implementación del equipo de monitoreo en tiempo real como para el control de vuelo de nuestra unidad.

1.5.- APORTACIONES

Las aportaciones que brinda este trabajo de tesis son las siguientes:

- Un prototipo de vuelo y de bajo costo en comparación con los existentes en el mercado actual.
- Una metodología que servirá como guía para la realización de un sistema de monitoreo de parámetros físicos ambientales y contaminación acústica.
- El uso de tecnologías que trabajen tanto con software y hardware open source debido a que existe una comunidad muy activa en el desarrollo de actualizaciones, plugins y extensiones.

- Un prototipo que este interrelacionado activamente con el internet de las cosas ya que en la actualidad se trata de una revolución en las relaciones entre objetos y personas, incluso entre objetos directamente, que se conectaran entre ellos y con la red y ofrecerán datos en tiempo real.

1.6.- ESTRUCTURA DE LA TESIS

Esta tesis está conformada por 7 capítulos los cuales se dispondrán de la siguiente manera:

1.6.1.-CAPITULO I. En este capítulo se da una introducción del problema a tratar dando a conocer la identificación del problema, descripción del problema, justificación, objetivos, aportaciones y estructura de la tesis.

1.6.2.-CAPITULO II. Este capítulo se inicia presentando la definición de un vehículo aéreo, reseña y sus aplicaciones, enfoque en UAV multirotores.

Seguidamente se hace una descripción de los sistemas de monitoreo y adquisición de datos, así como también las herramientas para la elaboración de una interfaz gráfica.

Y finalmente se habla de la contaminación acústica y los parámetros físicos ambientales.

1.6.3.-CAPITULO III. En este capítulo primeramente se definen los pasos a seguir para la construcción de nuestra unidad de monitoreo, detallando los elementos que integraran este prototipo, como lo sensores que se encargaran de adquirir los datos, así como la tarjeta que se comunicara con la interfaz que se diseñó.

También se hace hincapié en las herramientas que se usaron para el desarrollo de la interfaz como son node.js, node-red, base de datos firebase y el protocolo mqtt como principales componentes para la realización de este trabajo.

1.6.4.-CAPITULO IV. En este capítulo se desarrolla el modelo basado en leyes físicas que describan la posición y orientación de nuestro

cuadricoptero. El modelo dinámico del cuadricoptero se presenta bajo dos formulaciones matemáticas Newton-Euler y LaGrange-Euler.

El control del movimiento se logra variando la velocidad angular de cada uno de los motores, la razón principal de este análisis es que un cuadricoptero posee tres tipos de movimientos: alabeo (roll), cabeceo (pitch) guiñada (yaw).

También se procede a detallar el desarrollo de construcción de nuestra unidad, indicando los pasos a seguir y especificando cada uno de los componentes que integran nuestra unidad, empezando desde el frame, motores, variadores, y componentes varios.

1.6.5.-CAPITULO V. En este capítulo describiremos el uso de la tarjeta ardupilot apm2.8 y el uso asociado con el software mission planner.

Se hace una descripción acerca de los pasos a seguir adecuadamente para poder instalar el firmware de trabajo en la tarjeta ardupilot, como la calibración del acelerómetro, compás magnético, calibración de emisora flysky, configuración de modos de vuelo, configuración del sistema de telemetría y diversos parametros, calibración de los variadores, y el ajuste del pid siguiendo las instrucciones de la guía de instalación de ardupilot.

1.6.6.-CAPITULO VI. En este capítulo se muestra los resultados obtenidos en el desarrollo del presente trabajo de tesis, como son el prototipo de vuelo completamente terminado y funcional, así como también la interfaz desarrollada en node-red en completa interacción con la base de datos elaborada en firebase y finalmente presentando las especificaciones técnicas del prototipo de vuelo y de monitoreo así como las diversas pruebas realizadas en algunos puntos de la ciudad.

1.6.7.-CAPITULO VII. En este capítulo se detallan las conclusiones con respecto al trabajo elaborado y las recomendaciones para el desarrollo de trabajos futuros.

CAPITULO II

MARCO TEORICO

En el presente capítulo se hace una definición de un vehículo aéreo no tripulado (UAV o VANT), seguidamente de una reseña acerca del desarrollo de los VANT. Luego se hace una clasificación según su tipo de formación. Se hace enfoque en los VANT multirrotores presentando sus ventajas frente a modelos convencionales, principio de funcionamiento y tipo de control y dando a conocer los diferentes componentes que lo integran.

Seguidamente se hace una descripción de los sistemas de monitoreo y adquisición de datos, así como también las herramientas para la elaboración de una interfaz gráfica.

Y finalmente se habla de la contaminación acústica y los parámetros físicos ambientales.

2.1.- VEHICULO AEREO NO TRIPULADO (UAV)

El desarrollo tecnológico alrededor de los UAV (Unmanned air vehicle), es uno de los de mayor evolución en los últimos años. Un VANT es una unidad que no cuenta con algún tripulante a bordo y puede ser totalmente autónomo u operado mediante control remoto.

Estos vehículos han sido utilizados en aplicaciones militares (Figura 2.1) tales como reconocimiento de terreno y ataque. En la actualidad investigadores en el área de la robótica e inteligencia artificial aeronáutica y control continúan enfocando sus esfuerzos en aplicaciones en área civil.

Nos podemos encontrar actualmente con situaciones en la que la visión aérea es de muchísima ayuda, por ejemplo en la detección de incendios,

control policial en situaciones de riesgo y reconocimiento de desastres naturales.

Es importante recordar que cuando hacemos referencia a un UAV no estamos hablando solo del vehículo aéreo sino también del conjunto de sensores que ayudan a controlar su vuelo, sistemas que permitan determinar tanto su rumbo como trayectorias y equipos que permitan recolectar información para ser transmitidas a centros de control.¹



Figura 2.1: RQ-4 Global Hawk

2.2.- RESEÑA HISTORICA

En 1917 Charles Kettering de General Motors, desarrollo un biplano no tripulado pre-programado conocido como torpedo aéreo Kettering. De acuerdo a su idea, este vehículo accionado por un mecanismo de relojería debería plegar las alas en un lugar programado y caer sobre un enemigo como una bomba. Con la financiación del Ejército de EE.UU. fueron construidos unos cuantos aparatos, pero no fueron utilizados en combates.

¹ https://es.wikipedia.org/wiki/Veh%C3%ADculo_a%C3%A9reo_no_tripulado#Historia

Sin embargo, el primer drone producido en serie a gran escala fue el estadounidense Radioplane OQ-2, que sirvió como blanco volante para la formación de pilotos.

Investigaciones similares se llevaban a cabo en la Unión Soviética. En 1930-1940 el diseñador de aviones Nikitin desarrolló un drone planeador armado con torpedo PSN-1 y 2 y tipo de ala volante en dos modalidades: una como blanco aéreo para entrenamiento de pilotos y otra con automatización completa. A principios de la década de los 40 fue diseñado un torpedero con alcance de 100 kilómetros y una velocidad de 700 km/h. Sin embargo, los drones de este proyecto no fueron lanzados en serie.

Los VANT han demostrado sus capacidades en diferentes escenarios, especialmente en la guerra del Golfo y en la guerra de Bosnia el inmenso potencial con el que cuentan. En cuanto a la obtención, manejo y transmisión de información gracias a la aplicación de nuevas técnicas de protección de la misma resulta posible conseguir comunicaciones más seguras, más difíciles de detectar e interferir.²

2.3.- CLASIFICACION DE VEHICULOS AEREOS NO TRIPULADOS

Existen UAV de diferentes tipos y cada uno tiene cualidades que les hacen más aptos para un tipo de trabajo u otro. Si comparamos los UAV de ala fija (FIGURA 2.2) y los multirrotor (FIGURA 2.3), como los cuadricopteros, se parte de una diferencia básica, su diseño.

Los multirrotoros pueden tener cuatro (cuadricópteros), seis (hexacópteros) u ocho rotores (octocópteros). Es la forma en que se sostienen en el aire y

² <https://actualidad.rt.com/actualidad/view/80396-vehiculos-aereos-tripulados-hitos-historicos>

obtienen la propulsión para desplazarse. Pueden identificarse con el sistema de propulsión de un helicóptero. Por su parte, los drones de ala fija son más aerodinámicos y suelen tener una forma y desplazamiento similares a una aeronave convencional.

En el caso de los multicopteros, su sistema de propulsión les proporciona la capacidad de realizar vuelos en estatico. Todo lo contrario que los dispositivos de ala fija ya que sus características hacen que necesiten estar en constante movimiento. Aunque al tener un peso más ligero que un multirrotor les dota de una mayor autonomía.

Teniendo en cuenta estas diferencias, podemos distinguir diversas aplicaciones en las que se suele encuadrar cada modelo. Gracias a su mayor autonomía y aerodinámica que le otorga una mayor eficiencia y velocidad de vuelo, los UAV de ala fija suelen destinarse a labores en las que se necesite cubrir áreas extensas, como ocurre en trabajos de cartografía y teledetección. También se usan para labores de vigilancia, aprovechando que su impacto sonoro es menor que el de los multicópteros y esto dificulta su detección.

La capacidad de vuelo en una posición estática hace que los cuadricópteros y otros drones multirrotor resulten más útiles en labores de mayor precisión, como pueden ser trabajos de inspección en diferentes sectores, como el industrial.

Otra clasificación es que les puede asignar es de acuerdo a su altitud/rango como se muestra a continuación:

- Handheld : 600m de altitud en un rango de 2 Km

- Close: 1500m de altitud en un rango superior a 10Km
- Tactical: 5500m de altitud en un rango superior a los 50Km
- Male: 9000m de altitud en un rango de hasta 200Km
- Hale: 99Km de altitud con rango indefinido
- Orbital: En orbitas bajas terrestres (Mach 25)

El UAV que se desarrollara en este proyecto de tesis que se encargara de transportar nuestros sensores puede ubicarse en la categoría Handheld ya que el equipo será probado a nivel urbano abarcando áreas pequeñas.



Figura 2.2: UAV de ala fija



Figura 2.3: Uav multirotor

2.4.- VEHICULO AEREO NO TRIPULADO MULTIROTOR

Los cuadricopteros representan una plataforma muy versátil y de creciente popularidad que se encuentra dentro de la categoría UAV. Una de las principales características a destacar es la gran maniobrabilidad que posee este tipo de vehículo. Al tener cuatro motores, dos que giran en sentido horario y los otros dos giran en sentido anti-horario de tal manera que pueda contrarrestar el torque. Con ello los efectos giroscópicos y los momentos aerodinámicos tienden a cancelarse en el vuelo estacionario.

Para lograr el movimiento hacia adelante la velocidad del rotor trasero debe ser aumentada y, simultáneamente, la velocidad del rotor delantero debe ser disminuida. El movimiento lateral se ejecuta con el mismo procedimiento, pero usando los rotores de la izquierda y derecha. El movimiento de rotación sobre el mismo se obtiene con cada par de rotores es decir, se acelera los dos rotores con sentido horario mientras se desacelera los rotores con sentido anti-horario y viceversa (Figura2.4).

Dada su versatilidad frente a otros vehículos es su confiabilidad y portabilidad. Debido que al usar hélices fijas y generalmente motores eléctricos estos tienden a fallar menos mecánicamente, su mantenimiento es más sencillo y económico, lo cual es suma importancia ya que no existe la necesidad de usar hélices muy grandes para producir el mismo empuje, unas hélices más pequeñas aumentan la velocidad de reacción ya que existe menos inercia de parte de estas.³

³ Fuente: <http://www.copters.com/>

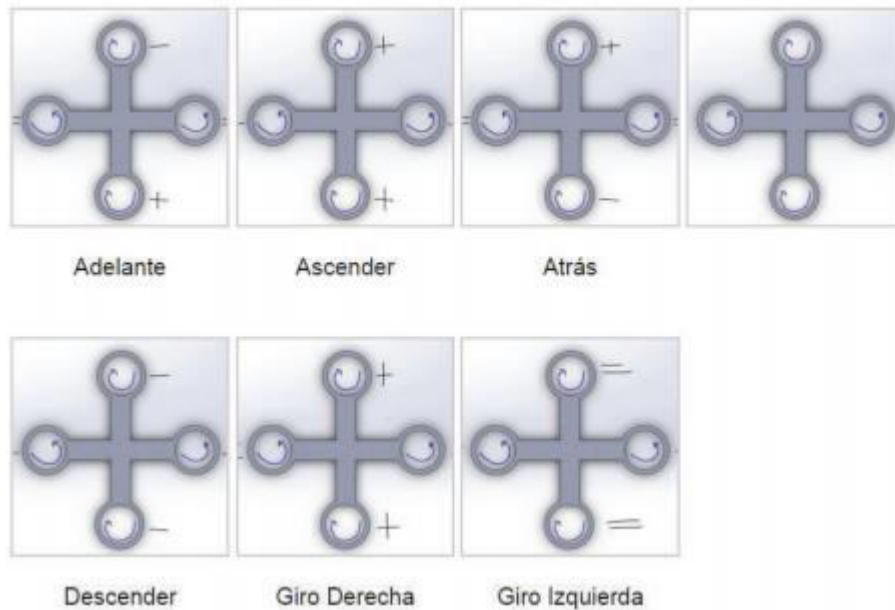


Figura 2.4: Movimiento de un cuadricoptero

2.5.- COMPONENTES QUE INTEGRAN UN UAV

A continuación se describen los principales elementos que componen un UAV:

2.5.1.-MOTOR BRUSHLESS

Es un motor síncrono trifásico que posee imanes permanentes en el rotor. Los devanados del estator son alimentados con tensiones de manera que el rotor sigue los campos magnéticos creados por el estator. Son motores que debido a su funcionamiento no poseen escobillas, lo cual los hace más eficientes.⁴

⁴ Fuente: <http://www.dronetrest.com/t/brushless-motors-how-they-work-and-what-the-numbers-mean/564>

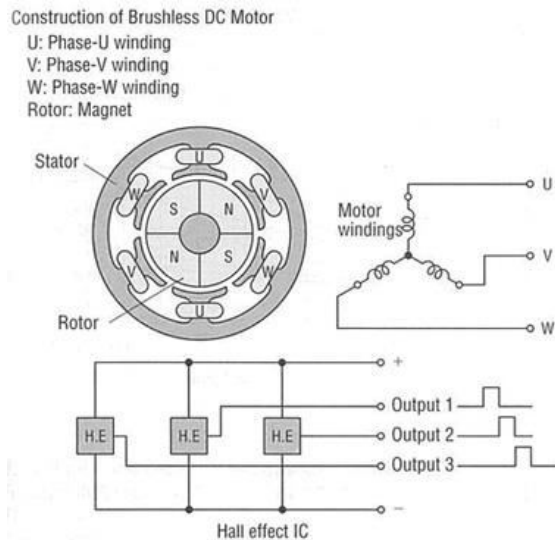


Figura 2.5: Esquema de motor brushless

Entre las principales ventajas que presenta frente a un motor de corriente continua se presentan las siguientes:

- Mejor respuesta dinámica
- Mayor eficiencia
- Mejor relación velocidad-torque
- Mayor rango de velocidades
- Inercia baja

Pero así como presenta ventajas también tiene inconvenientes, como su elevado costo de construcción y mantenimiento y necesita de un control electrónico óptimo para su buen funcionamiento.⁵

⁵ Fuente: <http://rctimer.com/>

OUTRUNNER COMPONENTS

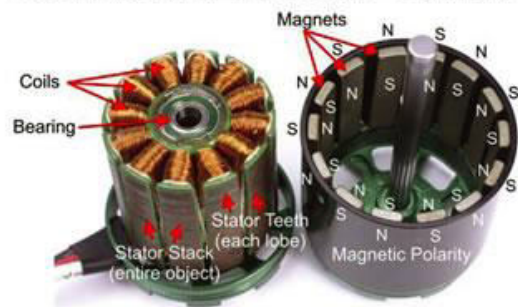


Figura 2.6: Motor Brushless a detalle

2.5.2.-CONTROLADOR ELECTRONICO DE VELOCIDAD (ESC)

Los ESC o controles de velocidad electrónicos son circuitos que tienen como propósito variar la velocidad de un motor eléctrico.

Los controles para motores sin escobillas tienen un sistema de control de velocidad es diferente al de motores con escobillas. Básicamente consiste en un circuito trifásico que envía una secuencia de señales para llevar a cabo la rotación y la dirección del motor.⁶

El ESC se encarga de convertir la corriente continua entregada por la batería en tres señales alternas desfasadas 120° . La velocidad varía según la frecuencia de la onda.

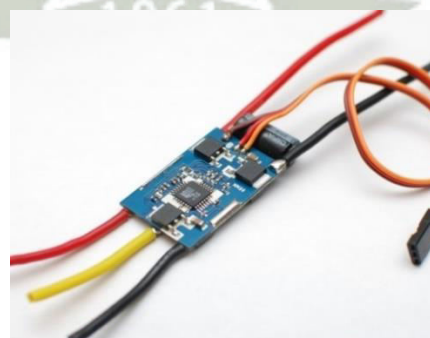


Figura 2.7: ESC

⁶ Fuente: <http://www.sarkanyellato.hu/wp-content/uploads/2011/10/RC-Timer-10.18.30.40A-ESC-Instruction.pdf>

2.5.3.-UNIDAD DE MEDICION INERCIAL (IMU)

Una unidad de medición inercial es un dispositivo electrónico cuya finalidad es la de medir velocidad, rotación y fuerzas gravitacionales de forma autónoma. Estos dispositivos son usados como componentes en sistemas de navegación de barcos, aviones, helicópteros, misiles o cualquier unidad en la que sea necesario poder estimar estas mediciones. Generalmente una IMU está compuesta por un conjunto de acelerómetros y giroscopios⁷.

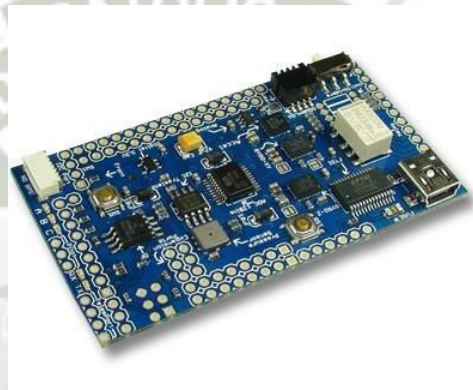


Figura 2.8: IMU

Fuente: <https://www.buildyourowndrone.co.uk/ardupilot-mega-imu-shield-oilpan-rev-h-v1-0-no-pin-headers>

2.5.4.-CONTROLADORES DE VUELO

Uno de los principales factores que aseguran la estabilidad de vuelo son los controladores de vuelo. Una persona común no estaría en la capacidad de poder controlar o estabilizar el movimiento de los motores de la unidad. Los controladores de vuelo (Figura 2.9) trabajan con una unidad de vuelo inercial la cual como ya se mencionó posee varios dispositivos integrados, un procesador el cual contiene el algoritmo de control el cual está pensado

⁷ Fuente: https://es.wikipedia.org/wiki/Unidad_de_medici%C3%B3n_inercial

para el control y estabilidad de los motores para el funcionamiento dinámico de la unidad.

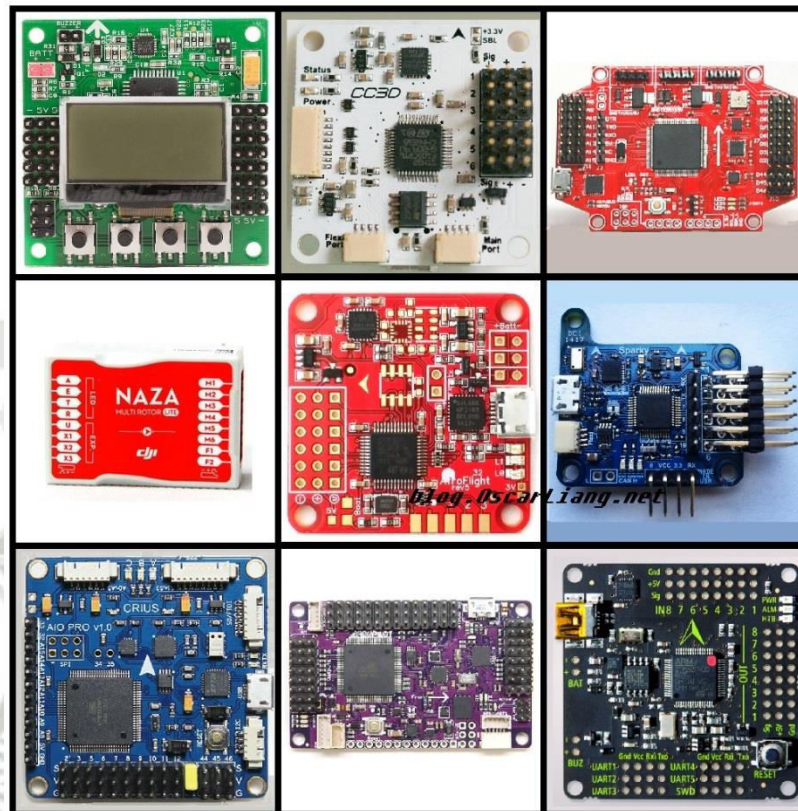


Figura2.9: Controladores de vuelo comerciales

Fuente: <https://oscarliang.com/best-flight-controller-quad-hex-copter/>

Con la combinación de estos componentes, se llega a conseguir una vasta información del medio para la toma de decisiones correctas sobre los actuadores los cuales son los encargados de realizar el vuelo. A continuación se detallaran algunos controladores de vuelo comerciales más usados:⁸

2.5.4.1 PIXHAWK

Es un controlador de vuelo que trae incorporado un procesador de 32 bits ARM CórteX M4F que puede realizar unas 252 MIPS a 168 MHz (Figura

⁸ Fuente: <https://oscarliang.com/best-flight-controller-quad-hex-copter/>

2.10). Adicionalmente, contiene un coprocesador de 32 bits en casos de emergencia.



Figura 2.10: Controlador Pixhawk

Fuente: <https://pixhawk.org/>

En cuanto a los sensores que este controlador de vuelo tiene, están:

Acelerómetro de seis ejes, con 14 bits de resolución⁹.

- Giroscopio con 16 bits de resolución.
- Barómetro con precisión mayor a 50 cm

En cuanto a comunicaciones y puertos de entrada y salida tenemos:

- 5 puertos serie
- 2 puertos CAN bus
- Puerto Spektrum DSM Satellite
- 1 puerto I2C
- 1 puerto SPI
- Entradas analógicas
- 8 salidas de motor

⁹ Fuente: <https://pixhawk.org/>

- 6 Entradas /salidas auxiliares digitales

2.5.4.2.-DJI-NAZA M-LITE

Es un controlador de vuelo sofisticado con diversas funciones para la unidad de vuelo, brinda una estabilidad muy confiable, adicionalmente cuenta con sistema GPS, cuenta también con funciones inteligentes de orientación, regreso a casa, seguro de altitud, seguro de bajo voltaje(Figura 2.11)¹⁰.



Figura2.11: Controladora de vuelo dji-naza M-lite

Fuente: <https://www.dji.com/naza-m>

2.5.4.3.-ARDUPILOT MEGA (APM)

Es un piloto automático profesional que se basa en la plataforma de arduino mega .Este piloto está en la capacidad de controlar aviones, helicópteros, multirrotores, rovers.Es un sistema completo, capaz de generar una estabilización autónoma, navegación basada en puntos (Waypoints) (Figura2.12).

Presenta las siguientes características:

¹⁰ Fuente: <https://www.dji.com/naza-m>

- Es open source
- Compatibilidad con los sistemas APM, Mission Planner y GCS.
- Modulo GPS para generación de trayectorias y regreso a casa
- Fácil configuración y uso de puerto USB
- Puerto de telemetría para transmisión de comandos.¹¹



Figura 2.12: Controladora de vuelo ardupilot¹²

Fuente: <http://ardupilot.org/copter/docs/common-ardupilot25-and-26-overview.html>

2.5.4.4.-SPARKY 2.0

Cuenta con un procesador STM32F4 funcionando a 168MHz con una mayor capacidad de memoria .Es ideal para el almacenamiento de datos y vuelos más largos ,secuencias de waypoints.Actualmente cuenta con una mejor conectividad, que es muy útil al momento de ejecutar magnetómetros externos,GPS,receptores de espectro(Figura 2.13).¹³

¹¹ Fuente:<http://ardupilot.org/>

¹² Fuente: <http://ardupilot.org/copter/docs/common-ardupilot25-and-26-overview.html>

¹³ Fuente: https://hobbyking.com/es_es/taulabs-sparky-2-0-32bit-flight-controller.html?__store=es_es

Cuenta con las siguientes características:

- Se ejecuta en código abierto Openpilot.
- Procesador STM32F4 funcionando a 168MHz.
- Más conectividad (USB, RCVR, FLEXI).
- Radio RFM22b utilizado en tele medida.
- Entradas analógicas para el seguimiento de voltaje y corriente.
- Conjunto de Sensores (barómetro, giroscopio acelerómetro).

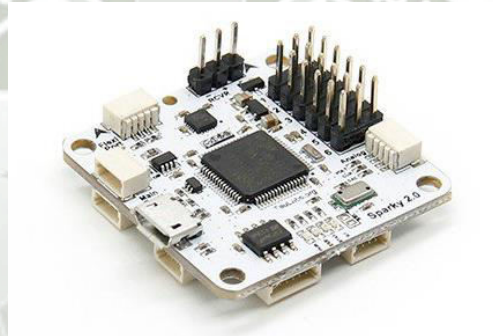


Figura2.13: Controladora de vuelo sparky 2.0

Fuente: <https://www.dronematters.com/sparky-2-0.html>

2.5.5.-SISTEMAS DE CONTROL POR RADIOFRECUENCIA

Generalmente el sistema de control por radiofrecuencia consta de dos partes: El control emisor (Figura 2.14), que es el que maneja el usuario.¹⁴

¹⁴ Fuente: <http://www.hobbypartz.com/79p-t6-6ch-radio-lcd.html>



Figura2.14: emisora flysky

Fuente: <http://www.ifuturetech.org/product/flysky-fs-t6-fs-t6-6ch-2-4g-w-lcd-screen-transmitter-fs-r6b-receiver/>

El receptor (Figura 2.15), este es conectado a nuestro controlador de vuelo.



Figura2.15: receptor flysky

Fuente: <http://www.ifuturetech.org/product/flysky-fs-t6-fs-t6-6ch-2-4g-w-lcd-screen-transmitter-fs-r6b-receiver/>

Este sistema de control trabaja en la banda de los 2.4GHz, frecuencia designada para este tipo de control de manera que no haya ningún problema ni tampoco haya interferencias con otras señales de radio. El rango de la frecuencia de 2.4GHz es mucho mayor con la capacidad de

llegar mucho más lejos, esto se debe a que las ondas se atenúan mucho más rápido a frecuencias más altas.

Para el control de equipos multirrotor, lo usual es trabajar con cuatro canales, pero también tengamos en cuenta las funciones adicionales que le queramos añadir, para esos casos es preferible usar de 5 a 8 cañales.¹⁵

2.5.6. SISTEMAS DE POSICIONAMIENTO GLOBAL (GPS)

El sistema de posicionamiento global (GPS) es un servicio de propiedad de los Estados Unidos de América, que proporciona a los usuarios datos de posicionamiento, navegación y cronometría; este sistema está constituido por tres segmentos: el segmento espacial, el segmento de control, y el segmento de usuario.

Los satélites del GPS proporcionan servicios a usuarios civiles y militares. El servicio a civiles es gratuito, está a disposición de todos los usuarios de manera permanente y global.¹⁶

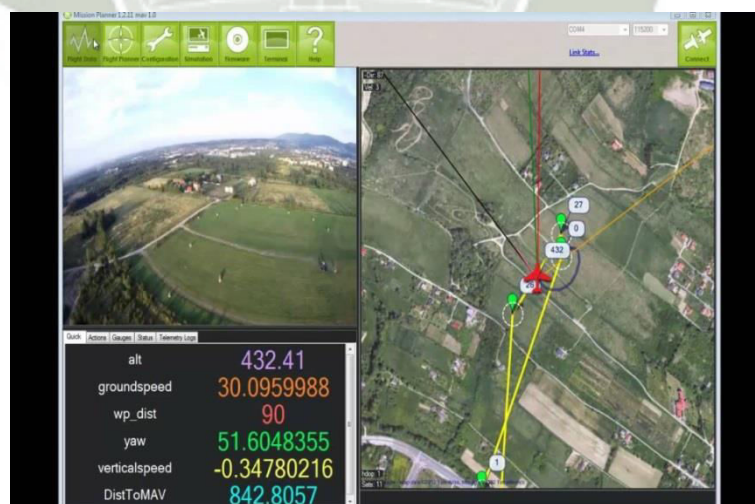


Figura2.16: Sistema de posicionamiento global misión planner

Fuente: <http://ardupilot.org/planner/docs/mission-planner-overview.html>

¹⁵ Fuente: <https://m.rcmoment.com/es/p-rm2326.html>

¹⁶ Fuente: <https://www.gps.gov/spanish.php>

2.5.7.-ESTACION DE CONTROL EN TIERRA (GCS)

Toda unidad de vuelo con sistema de autonomía cuenta con principalmente con un equipo de comunicación, en este caso una estación de control en tierra (Figura 2.17). Está integrada por un conjunto de equipos y sistemas que tienen como misión, cumplir las tareas de comunicación, planificación y control de la misión así como también la distribución de información a usuarios exteriores o finales.

Estas estaciones pueden estar situadas en tierra, mar o aire en un emplazamiento fijo o en una unidad móvil.

Como principales funciones tiene:

- Planificación de la misión: Esta se constituye fundamentalmente por la definición del plan de vuelo a seguir y por las acciones que tiene que ejecutar.
- Comunicación con el UAV: Para la transmisión de órdenes y recibir la telemetría, con datos de velocidad, altura y posición, se usan antenas direccionales y omnidireccionales con radio módems.
- Carga útil: Está compuesta por cámaras, ya sean infrarrojas, ópticas o térmicas; incluye también sensores, humedad, CO₂; estos con la finalidad de capturar una gran cantidad de información y enviarla en tiempo real a la estación en tierra.¹⁷

¹⁷ <http://flighttechspanish.weebly.com/control.html>



Figura2.17: Estación de control en tierra

Fuente: <http://www.dronetechnology.eu/ground-control-stations/>

2.5.8.-BATERIAS DE POLIMERO DE LITIO

La batería es uno de los elementos más importantes y en lo que más cuidado se debe de tener (Figura 2.18). Principalmente se utilizan baterías de LiPo y las tensiones que ofrecen junto con su capacidad y peso es la elección que cada cual debe hacer.

Por norma general, para multirrotores es mejor decantarse por unas batería de 3S, 4S, 5S o 6S. La letra S indica las celdas en serie, teniendo en cuenta que cada celda tiene una tensión nominal de 3.7 voltios, obteniendo tensiones nominales de 11.1, 14.8, 18.5 y 22.2 voltios respectivamente. Para multicopteros no profesionales, lo normal es poder trabajar en el rango de 3S y 4S donde la electrónica y los motores que componen a la unidad de vuelo tienen precios razonables. En caso de aviones, lo adecuado es que se trabaje con 3S o 2S.

Tengamos en cuenta también la capacidad o mAh, resulta que en aeronaves, el peso, la eficiencia de los motores, hélices, ESC y muchas

otras más variables que integran nuestra unidad de vuelo hacen que la elección de nuestra elección de la capacidad de la batería deba ser algo importante.

Con esto decimos, que si por ejemplo tenemos una batería 3S de 4000mAh en un cuadricoptero, este tendrá una autonomía de vuelo de aproximadamente de 10 minutos. Ahora si nosotros añadimos una batería igual en paralelo en tendríamos en total una batería de 3S 2P 8000mAh que al instalarla en nuestro cuadricoptero este tendría una autonomía de vuelo de 20 minutos, todo esto dicho en teoría. Pero lo normal es que no sobrepase de los 15 minutos.

La explicación a este suceso es el peso, añadir más capacidad no siempre resulta productivo debido a que si mantenemos constante el funcionamiento el resto de partes del equipo. A mayor peso, mayor consumo de energía y por lo tanto, llegar a aumentar la cantidad de baterías resultaría contraproducente.¹⁸



Figura2.18: Baterías LiPo

Fuente: <https://rightbattery.com/tag/lipo-batteries/>

¹⁸ https://es.wikipedia.org/wiki/Bater%C3%ADa_de_pol%C3%ADmero_de_litio

2.6.- INTERNET DE LAS COSAS

El Internet of Things, Internet de las Cosas o IoT es un concepto que se refiere a la interconexión digital de los objetos cotidianos con Internet, convirtiéndose así en objetos inteligentes. Este término fue propuesto por Kevin Ashton en 1999.

El internet de las cosas se refiere a una red de dispositivos físicos, la cual puede incluir hasta vehículos y edificios enteros, conectados entre sí y a internet y dotados de una serie de características tales como sistemas electrónicos, sensores, actuadores, software computacional, entre otras que les permite el manejo de una enorme cantidad de información la cual es almacenada, procesada y luego transmitida según sea la aplicación para la que han sido diseñados.

De esto se derivan términos como casa y edificios inteligentes, medios de transporte inteligentes y hasta ciudades inteligentes. Basta con imaginar algunas posibles aplicaciones para entender todo el potencial que hay detrás del internet de las cosas.

El gran valor detrás del Internet de las Cosas, no es tener un dispositivo u otro midiendo una gran cantidad de variables, sino las enormes oportunidades que se nos abren ante nosotros; fruto de la conexión de todos estos sensores y el manejo autónomo e inteligente de la información capturada.¹⁹

¹⁹ <http://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-iot>

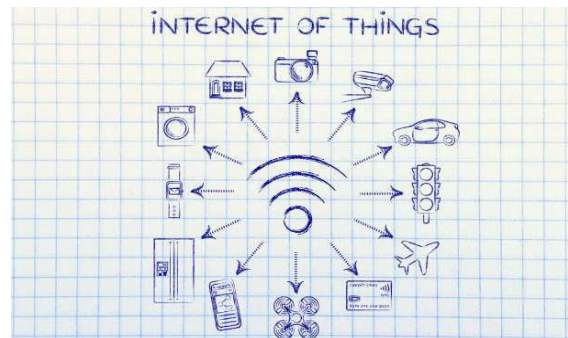


Figura2.19: Internet of things

Fuente: <https://innovateuk.blog.gov.uk/tag/internet-of-things/>

2.7.- PROTOCOLOS DE COMUNICACIÓN PARA INTERNET DE LAS COSAS

2.7.1.-HTTP

El protocolo HTTP (Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet.

El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML). Entre un navegador (el cliente) y un servidor web (denominado, entre otros, HTTP en equipos UNIX) localizado mediante una cadena de caracteres denominada dirección URL.²⁰

La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

²⁰ <https://www.arrow.com/es-mx/research-and-events/articles/protocols-for-the-internet-of-things>

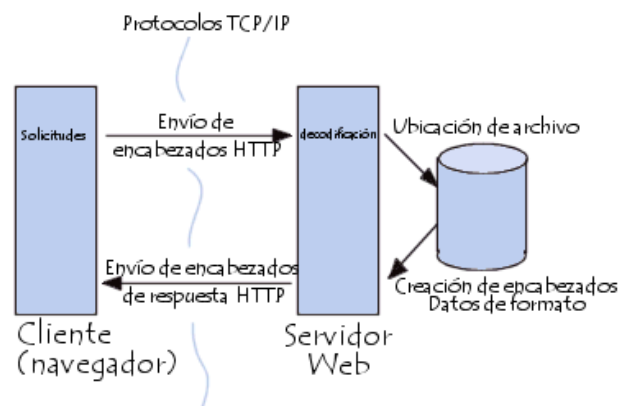


Figura2.20: Protocolo http

El navegador realiza una solicitud HTTP y el servidor procesa la solicitud y después envía una respuesta HTTP. En realidad, la comunicación se realiza en más etapas si se considera el procesamiento de la solicitud en el servidor.

2.7.2.- MQTT

MQTT (Message Queue Telemetry Transport), un protocolo usado para la comunicación machine-to-machine (M2M) en el "Internet of Things". Este protocolo está orientado a la comunicación de sensores, debido a que consume muy poco ancho de banda y puede ser utilizado en la mayoría de los dispositivos empotrados con pocos recursos (CPU, RAM). Un ejemplo de uso de este protocolo es la aplicación de Facebook Messenger tanto para android y iPhone. La arquitectura de MQTT sigue una topología de estrella, con un nodo central que hace de servidor o "broker" con una capacidad de hasta 10000 clientes. El broker es el encargado de gestionar la red y de transmitir los mensajes, para mantener activo el canal, los clientes mandan periódicamente un paquete (PINGREQ) y esperan la

respuesta del broker (PINGRESP). La comunicación puede ser cifrada entre otras muchas opciones.²¹

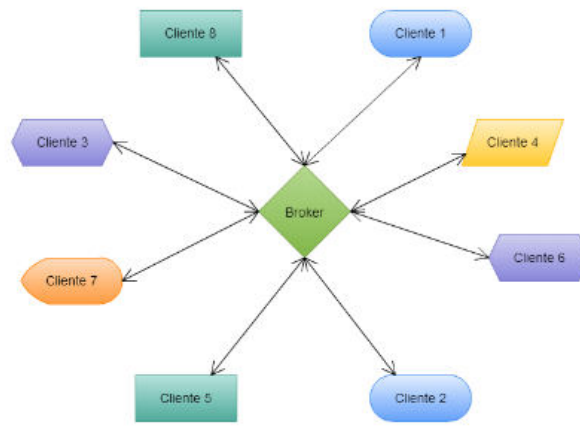


Figura2.21: MQTT bróker

MQTT es un protocolo abierto, sencillo, ligero y fácil de implantar. Es ideal para responder a las siguientes necesidades:

- Está especialmente adaptado para utilizar un ancho de banda mínimo
- Es ideal para utilizar redes inalámbricas
- Consume muy poca energía
- Es muy rápido y posibilita un tiempo de respuesta superior al resto de protocolos web actuales
- Permite una gran fiabilidad si es necesario
- Requiere pocos recursos procesadores y memorias

²¹ <https://www.arrow.com/es-mx/research-and-events/articles/protocols-for-the-internet-of-things>

2.7.3.-CoAP

CoAP (Constrained Application Protocol) es un protocolo software a nivel de aplicación pensado para ser usado en dispositivos electrónicos simples permitiendo que pueda comunicarse sobre Internet.

Está pensado especialmente para sensores de baja potencia, se ha diseñado para trasladar el modelo HTTP pero incluyendo otros requisitos como multicast, bajo overhead y simplicidad, que son muy importantes para el Internet de las cosas (IoT) y Machine-to-Machine (M2M).

CoAP implementa el modelo REST de HTTP (con las primitivas GET, POST, PUT y DELETE), usa cabeceras reducidas, y limita el intercambio de mensajes, añadiendo soporte UDP y otras modificaciones como mecanismos de seguridad específicos.²²

2.8.- PLATAFORMAS DE DESARROLLO PARA EL INTERNET DE LAS COSAS

2.8.1.-NODE RED

Node-RED es una herramienta de visualización sencilla y de código abierto para la conexión de dispositivos para el Internet de las Cosas. La herramienta, desarrollada por el departamento Emerging Technology de IBM, permite conectar, un hardware, una API o un servicio online. Node-RED proporciona una interfaz de usuario amigable para que los desarrolladores conecten dispositivos de forma sencilla y rápida. Node-RED está desarrollada en node.js, una plataforma JavaScript del lado del servidor muy utilizada en proyectos de IoT, y se puede ejecutar en la nube.

²² <https://www.arrow.com/es-mx/research-and-events/articles/protocols-for-the-internet-of-things>

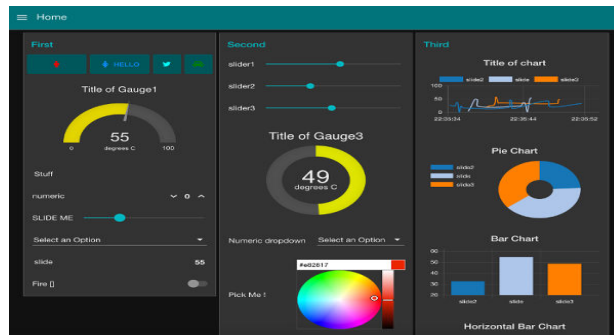


Figura2.22: Dashboard Node-Red

Fuente: <https://github.com/node-red/node-red-dashboard>

2.8.2.- ECLIPSE IoT

Eclipse IoT es una plataforma de código abierto que permite el desarrollo en Java de aplicaciones para el Internet de las Cosas. Proporciona un grupo de tecnologías open source para conectar y gestionar varios dispositivos en un entorno IoT.

Además, da soporte para algunos de los estándares abiertos fundamentales para cualquier solución del Internet de las Cosas: MQTT (un protocolo de conexión machine-to-machine, CoAP (un protocolo para la conexión sencilla de los dispositivos a Internet) y Lightweight M2M (un protocolo de comunicación entre servidor y cliente que permite la transmisión de datos o la administración de sensores o móviles).²³

Eclipse IoT ofrece servicios de puerta de enlace al Internet de las Cosas para ayudar los desarrolladores a manejar tanto dispositivos como aplicaciones IoT. Dentro de esta plataforma, los programadores contarán

²³ <https://iot.eclipse.org/>

con el marco de desarrollo Kura, basado en Java y OSGi, que implementa servicios tan importantes como:

- Administración de la conectividad en la nube.
- Apoyo a los protocolos de conexión entre dispositivos y servidores.
- Configuración de red WiFi.
- Aplicación remota y configuración y administración de dispositivos.

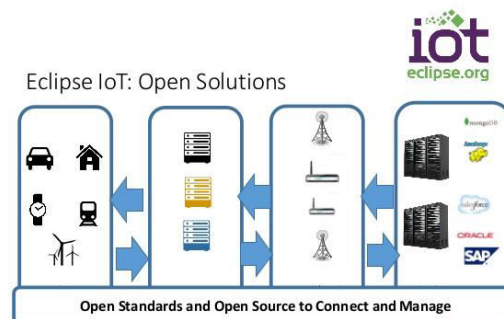


Figura2.23: Eclipse IoT

Fuente: <https://www.slideshare.net/AGILEIoT/eclipse-foundation-contribution-to-the-agileiot-project>

2.8.3.-THINGSPEAK

Es una API abierta para el Internet de las Cosas que permite recopilar, almacenar, analizar, visualizar y actuar sobre la información recogida en sensores y dispositivos como aplicaciones web y móviles, redes sociales como Twitter, hardware de código abierto como Arduino, Raspberry Pi o BeagleBone o con lenguajes de cálculo computacional como MATLAB.

ThingSpeak funciona siempre con canales, los cuales contienen los campos de datos, ubicación y estado. Para empezar a trabajar con esta interfaz es necesario crear un canal, donde se recopilará la información de dispositivos y aplicaciones, datos que posteriormente se pueden analizar y

visualizar en gráficos y el paso final es operar sobre esa documentación. El proceso con la API siempre es el mismo.²⁴

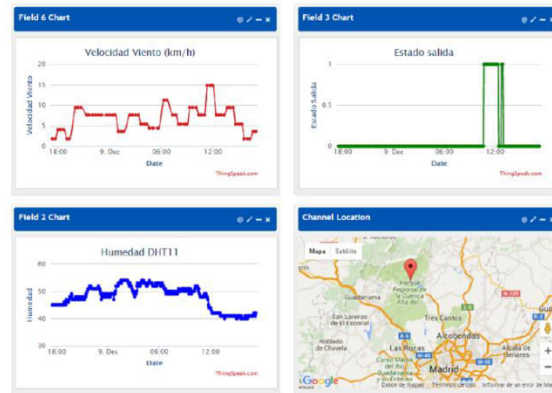


Figura2.24: Dashboard en thingspeak

Fuente: <https://community.particle.io/t/control-monitor-and-log-a-spark-core-with-thingspeak/10543>

2.8.4.-THINGERIO

Thinger.io es una plataforma española de código abierto que podemos utilizarla gracias a sus propios servidores como en GitHub para instalarla en una máquina propia y ofrecen una cuenta gratuita para utilizar su infraestructura en la nube.

Dispone de una consola de administración bastante completa donde podemos gestionar y geolocalizar nuestros objetos conectados. Además, la documentación es extensa y libre con una comunidad de desarrolladores muy activa.²⁵

²⁴ <https://bbvaopen4u.com/es/actualidad/apis-para-el-internet-de-las-cosas-thingspeak-pachube-y-fitbit>

²⁵ <https://thinger.io/>

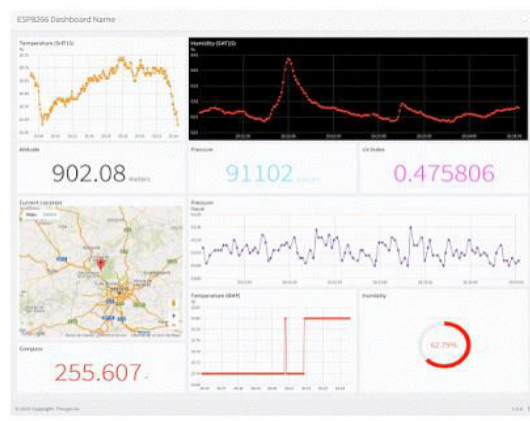


Figura2.25: Dashboard en thingerio

Fuente: <https://community.thinger.io/t/arduino-mkr1000-thinger-io/117>

2.9.- TARJETAS DE ADQUISICION DE DATOS

Estos equipos como su nombre lo indica, son los productos y/o procesos utilizados para recopilar información para documentar o analizar algún fenómeno. Esta tecnología ha ido avanzando con el pasar de los años, y estos procesos se han ido simplificando, haciéndose más precisos, versátiles, y fiables a través de equipos electrónicos.

Estos forman parte de un sistema focal, uniendo una gama de productos, tales como sensores que indiquen temperatura, presión, caudal o nivel.²⁶

Estos equipos están conformados por:

- Convertidor análogo-digital.
- Entradas/ salidas digitales.
- Entradas simples
- Bus de interfaz de propósitos generales(GPIB)

²⁶ <https://www.jmi.com.mx/tarjetas-de-adquisicion-de-datos>

2.10.- BASES DE DATOS

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales han sido recolectados y explotados por los sistemas de información de una empresa o negocio particular.

Entre sus características principales tenemos:

- Independencia lógica y física de los datos.
- Integridad de los datos.
- Consultas complejas optimizadas
- Seguridad de acceso y auditoria
- Respaldo y recuperación
- Acceso a través de lenguajes de programación.
- Acceso concurrente por parte de múltiples usuarios.

Cada sistema de base de datos posee tipos que pueden ser similares o diferentes; entre los más comunes podemos nombrar: numérico, booleanos, fechas, alfanuméricos y auto incrementables.

En la actualidad existen diferentes tipos de base de datos, entre las principales tenemos:

- ACCESS: Base de datos desarrollada por Microsoft
- Microsoft SQL server: Es una base de datos mucho más potente que Access con la capacidad de manejar grandes volúmenes de información, también es desarrollada por Microsoft.
- MySql: Base de datos con licencia GLP basada en un servidor, su principal característica es su rapidez.

Hoy en día el lenguaje SQL es el más universal en los sistemas de base de datos. Este lenguaje nos permite realizar consultas a nuestras bases de datos para mostrar, insertar, actualizar y borrar datos.²⁷

2.11.- COMUNICACIONES

Una de las ventajas de las tarjetas comerciales que existen hoy en día es que podemos dotarlos de comunicación de una forma sencilla, añadiendo shields o breakout boards y así disponer de casi cualquier tipo de comunicación, tanto de acceso a internet como de comunicar estas tarjetas entre sí o con otros dispositivos de una red privada.

A continuación se enumera las principales comunicaciones que utilizan:

- Wifi
- Ethernet
- Bluetooth
- Zigbee/Xbee
- Xbee
- LoRaWAN es una especificación de una red LPWAN(Low Power Wide Area Network)
- Sigfox:solución de conectividad celular mundial para el internet de las cosas
- Radiofrecuencia

²⁷ <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
<https://aprendiendoarduino.wordpress.com/2016/12/18/comunicaciones-arduino/>

2.12.- CONTAMINACION ACUSTICA

El efecto del ruido en las emociones humanas van de insignificantes, pasando por molesto y fastidioso, hasta psicológicamente perturbado. Fisiológicamente el ruido puede ser inocuo o hasta doloroso y físicamente dañino. Incluso también puede llegar a tener efectos económicos al decrecer la eficiencia de un trabajador, afectando así los márgenes de ganancia.

Lo primero que se debe realizar para un control del ruido es comparar el ruido existente o potencial con criterios de categorización adecuados.

Los medios ruidosos más fáciles de catalogar son aquellos que varían lentamente tanto en el nivel como en el contenido espectral. Como por ejemplo los ruidos producidos por maquinarias, y el ruido ambiental que varía lentamente entre el día y la noche. Los niveles de interferencia con el lenguaje (SIL) y las curvas de categorización del ruido (NC) son ejemplos de procedimiento de categorización.²⁸

2.12.1.PRESION SONORA

El sonido es una sensación auditiva provocada por las vibraciones y ondas acústicas de frecuencia entre 20 Hz y 20.000 Hz., que se propagan en un medio sólido, líquido o gaseoso.

El sonido para propagarse tiene necesidad de un medio, no se propaga en el vacío. Cuando las frecuencias de excitación del fenómeno acústico son inferiores a 20 Hz. se habla de infrasonidos. Cuando son superiores a 20.000 Hz., de ultrasonidos.

²⁸ Fuente: <http://www.acusticaintegral.com/3303/presion-sonora-oscilogramas/>

Cabe recordar lo siguiente:

- Margen de frecuencias audibles de 20 Hz a 20000 Hz
- Aplicación de una ponderación logarítmica
- Margen muy amplio de valores numéricos
- Margen de presión audible 20 mPa(umbral de audición) a 200Pa (límite del dolor).²⁹

2.12.2.-CONTAMINACION ACUSTICA EN AREQUIPA

Dentro de la jurisdicción de la Provincia de Arequipa la producción de ruidos nocivos y Molestos, cualquiera fuera el origen y lugar en que estos se producen, tales como el uso de bocinas o claxon de cualquier vehículo motorizado en general, los escapes libres de cualquier tipo de vehículo motorizado, los altoparlantes y megáfonos de emisión de sonidos tanto estacionarios como móviles (equipos de sonido, sirenas, silbatos, cohetes, petardos o cualquier otro medio que por su intensidad, tipo duración o persistencia ocasionen molestias al vecindario). Es decir, la cada vez insoportable, pero más aún, dañina y atentatoria contaminación sonora producida por los vehículos, debería tener en este argumento normativo un marco legal que le debería ser imposible sortear. No obstante, ello no sucede en nuestra realidad; cada vez nos estamos acostumbrando, a pesar de nuestro continuo Rechazo y ante nuestra frustrante impotencia - aunque en no pocos casos, a causa. También de nuestro complaciente silencio a que los ruidos vehiculares formen una parte no deseada y más bien desagradable de nuestra vida cotidiana.

²⁹ Fuente:<https://www.acusticaintegral.com/3305/potencia-sonora-equivalencias/>

Los niveles de contaminación sonora han incrementado en los últimos diez años y sobrepasan ampliamente los estándares máximos permitidos. El 65% del sonido ambiental es provocado por emisiones de vehículos. De esta manera, los ruidos del claxon y de los motores de los bolidos se convierten en los principales agentes de contaminación sonora.

Así por ejemplo, el primer elemento llega a alcanzar los 90 a 95 decibeles cuando los estándares máximos apuntan a cifras máximas de 50 decibeles (DB) en zonas próximas a hospitales, centros educativos, orfanatos, asilos, albergues, entre otros.

Los estándares de calidad ambiental para ruido no se respetan en Arequipa, donde el Centro Histórico es uno de los sectores de mayor contaminación sonora.

Especialistas indican que la regulación evidencia cinco tipos de mediciones sonoras para igual número de zonas y en diferente horario. La primera, denominada zona de protección especial (hospitales, colegios, asilos, etc.), considera valores máximos de 50 decibeles entre las 7:00 horas a las 22:00 horas (horario diurno). Fuera de ese horario (nocturno) la exigencia es de 40 decibeles. Sin embargo, ello no se cumple. En el hospital Honorio Delgado, los valores llegan a los 90 a 95, debido al tráfico vehicular .

La segunda es la zona residencial, que involucra a viviendas o residencias. En estos lugares en horario diurno, los decibeles del ruido deben llegar como máximo a 60 y 50 en la noche, empero ello no ocurre. En las avenidas principales, los valores sobre pasan los 70 a 80 decibeles.

En zonas comerciales, los estándares máximos deben llegar a 70 decibeles durante el día y 60 de noche. En zona industrial, las cifras son de 80 DB en la mañana y 70 para noche, mientras que para zonas mixtas (residencial- comercial o industrial) los valores no deben excederse de los 60 a 80 DB, pero en la práctica ello no ocurre³⁰.

2.13.- PARAMETROS FISICOS AMBIENTALES

Los parámetros ambientales en general, son aquellos que nos permiten conocer las condiciones del agua, aire, suelo y de los ecosistemas.

Estos valores pueden ser resumidos en indicadores que nos permitan alcanzar un mejor entendimiento de ambiente que nos rodea.

Estos elementos, son aquellos fenómenos que se emplean para definir un lugar o una región de la superficie de la tierra. Básicamente si son: temperatura, humedad relativa, presión barométrica, luminosidad, radiación ultravioleta.³¹

2.13.1.TEMPERATURA

Es la cantidad de energía solar retenida por el aire en un momento dado. El termómetro (Figura 2.26) es el instrumento de fiabilidad que se utiliza para medir esa cantidad de energía. Esta medición se debe realizar a 1,5 metros del suelo, siendo un lugar ventilado y protegido de la influencia directa de los rayos del sol. El resultado se expresa en una escala de grados centígrados (Celsius) o en la escala de grados Fahrenheit. Se puede afirmar que la temperatura depende ante todo de la radiación solar.

³⁰Fuente: <http://larepublica.pe/archivo/715600-contaminacion-sonora-en-arequipa-subio-20-ocasionando-males>

³¹ Fuente:<http://www.lab-ferrer.com/sensores/instrumentacion-y-sensores/parametros-ambientales.html>

Es debido a esto que las temperaturas son más elevadas en horas del día que en el transcurso de la noche.³²

No obstante, son tres los factores del clima que hacen variar funcionalmente a la temperatura. Estos son:

- Altitud
- Latitud
- Proximidad al mar



Figura2.26: Termómetro digital con sonda

2.13.2.-HUMEDAD

Es la cantidad de vapor de agua que contiene la atmosfera. La misma que proviene de una parte de la evaporación del agua de la superficie de los océanos, lagos, ríos y la otra que es suministrada por la tierra y la vegetación.

Sin embargo, de un lugar a otro de la superficie terrestre, la humedad varía considerablemente y esto depende justamente de la temperatura del aire, ya que cuando es mayor la temperatura, mayor es la evaporación. Se expresa utilizando el término humedad relativa, es decir el cociente entre la

³² Fuente: <http://conceptodefinicion.de/temperatura/>

cantidad de vapor de agua contenido en la atmosfera y la máxima que puede contener, expresándose así en porcentaje.³³

Este proceso es realizado por el instrumento de medida llamado higrómetro (Figura 2.27).

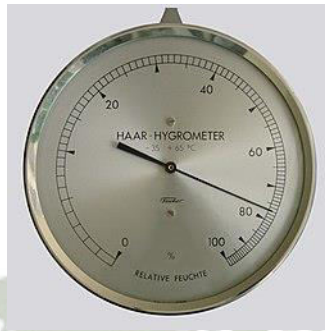


Figura2.27: higrómetro

2.13.3.-PRESION

La atmosfera en si está constituida por un conjunto de gases que conforman el aire, este último aunque no sea visible, tiene cierto volumen, peso y por lo tanto ocupa un lugar. Entonces, se denomina presión atmosférica, al peso que ejerce el aire sobre la superficie terrestre.

Según esto, la primera unidad de medida sería la presión que ejerce una fuerza de 1 newton sobre una superficie de 1 metro cuadrado. A esta unidad se le llama Pascal.

Otra unidad de medida usada es el milibar (mb) que es equivalente a 1 hectopascal (hPa) o 100 pascales. Siendo el barómetro (Figura2.28) el instrumento encargado de realizar estas medidas.³⁴

³³ <https://content.meteoblue.com/es/ayuda/variables-meteorologicas/humedad>

³⁴ http://apcd-spv.org/index.php?lang=es&secc=variables&type=pressio_barometrica



Figura2.28: Barómetro analógico

2.13.4.-INTENSIDAD LUMINOSA

La intensidad luminosa se define como la cantidad de flujo luminoso que emite una fuente por unidad de ángulo sólido. El nivel de iluminación o iluminancia, es la cantidad de luz que se mide en una superficie plana. La iluminancia se mide en lux (en el SI métrico). Un lux es un lumen por metro cuadrado. El instrumento para la medición de la intensidad lumínica en un punto dado se llama luxómetro (Figura 2.29).³⁵



Figura2.29: Luxómetro digital

2.13.5.- RADIACION ULTRAVIOLETA

La energía solar resulta del proceso de fusión nuclear que tiene lugar en el sol. Esta energía es el motor que mueve nuestro medio ambiente, siendo la energía solar que llega a la superficie terrestre 10.000 veces mayor que la

³⁵ <http://www.schreder.com/es-cl/centro-formacion/esencial-alumbrado/luminous-intensity>

energía consumida actualmente por toda la humanidad. La radiación es transferencia de energía por ondas electromagnéticas y se produce directamente desde la fuente hacia fuera en todas las direcciones. Estas ondas no necesitan un medio material para propagarse, pueden atravesar el espacio interplanetario y llegar a la Tierra desde el Sol.

La longitud de onda y la frecuencia de las ondas electromagnéticas, son importantes para determinar su energía, su visibilidad y su poder de penetración. Todas las ondas electromagnéticas se desplazan en el vacío a una velocidad de 299.792 Km/s. Estas ondas electromagnéticas pueden tener diferentes longitudes de onda.

El conjunto de todas las longitudes de onda se denomina espectro electromagnético. El conjunto de las longitudes de onda emitidas por el Sol se denomina espectro solar.

El instrumento encargado de realizar la medición es el pirómetro (Figura 2.30). Diseñado para medir la densidad de flujo de radiación.³⁶



Figura 2.30: Pirómetro

³⁶ http://apcd-spv.org/index.php?lang=es&secc=variables&type=radiacio_ultraviolada



3.1.-DIAGRAMA DE BLOQUES GENERAL

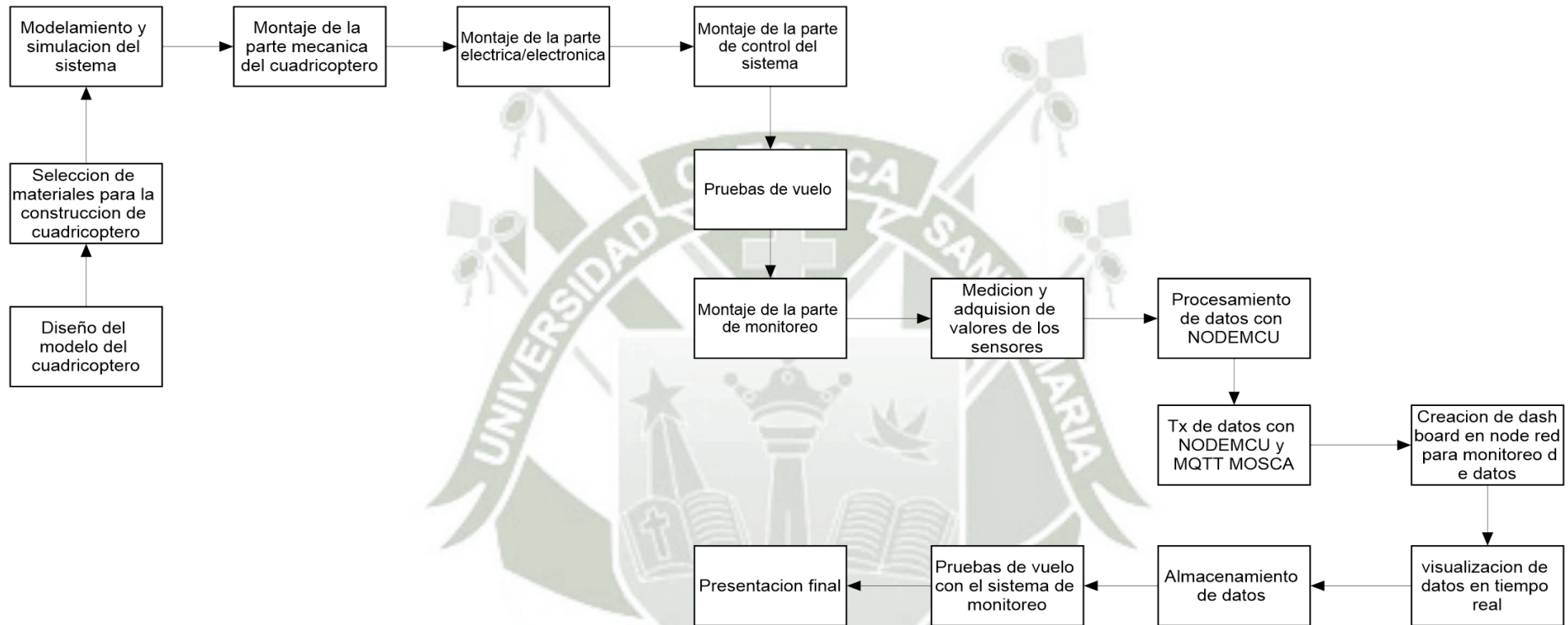


Diagrama de Diagrama general

3.2 DESCRIPCION DE LOS DISPOSITIVOS UTILIZADOS

3.2.1. SENSOR DE TEMPERATURA DHT22

El DHT22 es un sensor digital de medición de temperatura y humedad. Este sensor está basado en un termistor que sirve para medir el aire circundante (temperatura) e implementa un sensor interno capacitivo para la medición de humedad.

CARACTERISTICAS	PARAMETROS
Rango de medición de humedad	0-100% RH
Rango de medición de temperatura	-40 a 80 °C
Precisión de temperatura	±0.5°C
Precisión de humedad	±2 % RH
Alimentación	3 a 5 voltios

Tabla 3.1: Características Sensor DHT22

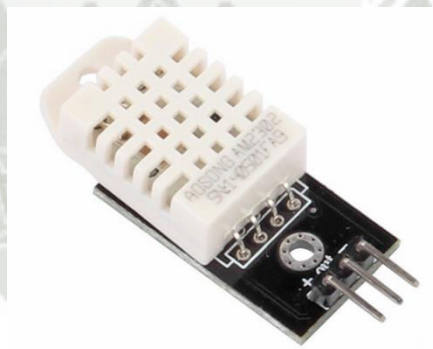


Figura3.1: Sensor DHT22

Fuente: www.sparkfun.com

3.2.2.-SENSOR DE PRESION BMP180

Ésta placa incluye un sensor de presión barométrica BMP180 de alta precisión con un rango de medida de entre 300 y 1100 hPa (Hecto Pascal) con un margen de error mínimo de tan sólo 0.03 hPa. Está basado en tecnología piezo-resistiva de alta eficiencia, linealidad y larga duración. El sensor tiene un rango de alimentación de entre 1,8V y 3,6 Vdc. Está diseñado para ser conectado directamente a un micro controlador

mediante su interfaz I2C. Dispone de dos resistencias pull-up de 4,7k sobre el bus I2C.

CARACTERISTICAS	
Interfaz	I2C
Alimentación	1.8-3.6voltios
Rango de presión	300 a 1100 hPa
Resolución	1 Pa
Frecuencia de muestreo	120 Hz
Consumo de energía	Bajo
Precisión de temperatura	1°C
Resolución de temperatura	0.1°C
Presión absoluta	1 hPa

Tabla 3.2: Características sensor bmp180

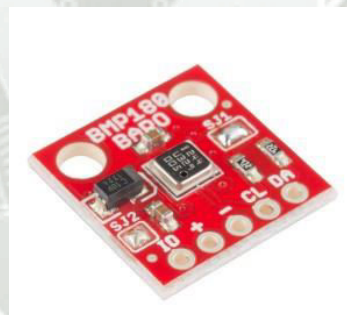


Figura3.2: Sensor bmp180

Fuente: www.sparkfun.com

3.2.3.-SENSOR DE LUMINOSIDAD BH1750

El BH1750 es un sensor digital del nivel de luz. A diferencia a otros sistemas de medición del nivel de luz, como por ejemplo las resistencias LDR, la respuesta espectral del BH1750 está diseñada para ser similar a la del ojo humano por lo que son capaces de proporcionar la medición de lux.

Rango de medición	0.11 a 100000 lux
ADC	16 bits (65535 niveles)
Tipo de comunicación	I2C
Voltaje de alimentación	2.4 a 3.6V.
Costo	Mínimo
Influencia al espectro infrarrojo	Baja
Rechazo al ruido de 50/60 Hz (luz artificial)	Si

Tabla3.3: Características sensor bh1750

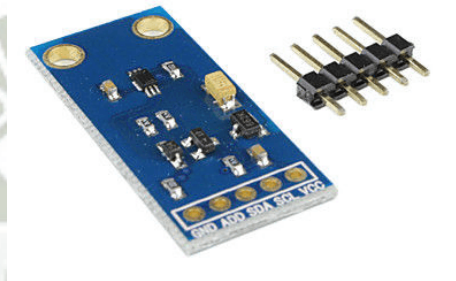


Figura3.3: Sensor bh1750

Fuente: www.sparkfun.com

3.2.4.-SENSOR DE RADIACION UV GYML8511

El sensor ML8511 detecta luz con una longitud de onda entre 280-390nm, este rango cubre tanto al espectro UV-B como al UV-A. La salida analógica está relacionada linealmente con la intensidad UV (mW/cm^2).

Características ML8511	
Voltaje De Funcionamiento	3.3-5 V
Tensión de Salida	DC 0-1 V
Precisión:	± 1 Índice UV
Corriente de trabajo	0.06mA
Longitud de onda	200nm-370nm
Temperatura de trabajo	-20~ 85 °C
Tamaño	19.80*15mm

Tabla3.4: Características sensor GYML8511

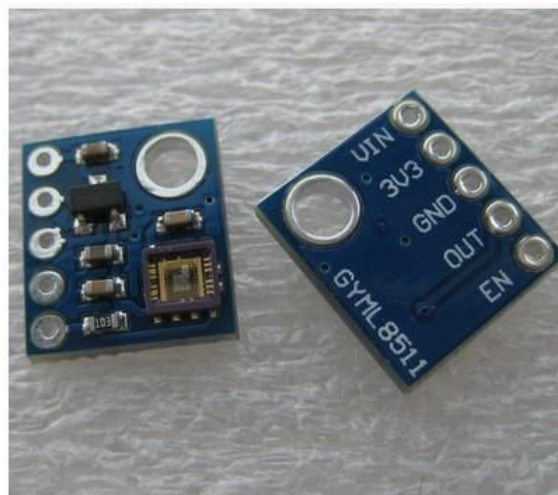


Figura3.4: Sensor GYML8511

Fuente: www.adafruit.com

3.2.5.-SENSOR DE DIOXIDO DE CARBONO MQ-135

El MQ-135 es un sensor de gases peligrosos utilizado para el control de la calidad del aire y es adecuado para la detección de NH₃ (amoníaco), alcohol, benceno, humo, CO₂, etc.

Este sensor no proporciona valores absolutos, sino que simplemente proporciona una salida analógica que debe ser monitoreado y se comparada con los valores de umbral.

CARACTERISTICAS SENSOR CO2	
Voltaje de operación	5 voltios
Corriente de operación	150mA
Potencia de consumo	800mW
Tiempo de precalentamiento	20 segundos
Resistencia de carga	potenciómetro ajustable
Detección de partes por millón	10ppm~1000ppm
Concentración detectable	Amoniaco,sulfuro,co2,humo,benceno
Concentración de oxigeno	2%~21%
Humedad de operación	<95%RH
Temperatura de operación	(-)20°C~70°C

Tabla3.5: Características sensor de CO2



Figura3.5: Sensor de CO2

Fuente: www.sparkfun.com

3.2.6.-SENSOR DE MONOXIDO DE CARBONO MQ-7

El MQ-7 puede detectar concentraciones de CO de 20 a 2000 ppm (partes por millón).

Este sensor tiene una alta sensibilidad y un tiempo de respuesta rápido. La salida del sensor es una resistencia análoga.

Características sensor MQ7	
detección	monóxido de carbono
temperatura de operación	(-)20 °C a 50 °C
Humedad relativa de operación	< 95%
Material sensible	SnO2
Voltaje de alimentación	5 V ±0.1
Resistencia de calefactor	33 Ω ±5%
Potencia consumida por el calefactor	0.35 W
Concentración de oxígeno (O2) en operación:	21%
Salida	Analógica
Concentraciones medibles	20 a 2000 ppm

Tabla3.6: Características sensor de CO



Figura3.6: Sensor de CO

Fuente: www.sparkfun.com

3.2.7.-SENSOR DE SONIDO

Este módulo incorpora un micrófono electret de 20KHz amplificado por el MAX4466, un chip diseñado específicamente para la tarea de preamplificar micrófonos.

En la parte posterior cuenta con un pequeño potenciómetro para ajustar la ganancia de 25x a 125x, la salida del op-amp es riel a riel, por lo que a máxima ganancia puede llegar a 5Vpp.



Figura3.7: Sensor de sonido

Fuente: www.adafruit.com

3.2.8. MODULO EXPANSOR ADS1115

El chip ADS1115 de este módulo tiene un convertor delta sigma de 16 bits de resolución, puede leer 4 señales individuales o 2 diferenciales, un comparador programable con salida directa al pin alrt, y ganancia

programable que le permiten leer hasta 256mV en escala completa; haciendo de este un dispositivo muy versátil.

Los datos digitales son leídos mediante bus I2C y puede alcanzar hasta 860 muestras por segundo. Dirección I2c por defecto: 0x48.

CARACTERISTICAS	
voltaje de operación	2.0V- 5.5V DC
corriente de alimentacion	200uA en modo de lectura continua
interface con microcontrolador	I2C
chip principal	ADS1115
muestras por segundo	8 - 860
direccion I2C	cuatro direcciones 0x48-0x4B.
rango de ganancia programable	1-128

Tabla3.7: Características de módulo expensor ads1115

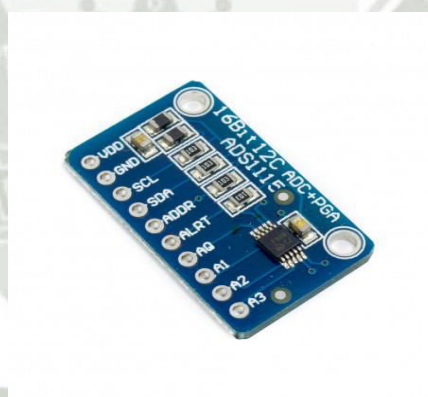


Figura3.8: Modulo expensor adc 16 bits

Fuente: www.adafruit.com

3.3.-NODEMCU ESP8266

El NodeMcu es un kit de desarrollo de código abierto basado en el popular chip ESP8266 (ESP-12E), que utiliza el lenguaje de programación Lua para crear un ambiente de desarrollo propicio para aplicaciones que requiera conectividad Wifi

Ofrece una solución completa y autónoma de redes Wi-Fi, lo que le permite alojar la aplicación o servir como puente entre Internet y un microcontrolador.El ESP8266 tiene potentes capacidades a bordo de procesamiento y almacenamiento que le permiten integrarse con sensores y dispositivos específicos de aplicación a través de sus GPIOs con un desarrollo mínimo y carga mínima durante el tiempo de ejecución.

3.3.1.-CARACTERISTICAS

NodeMCU presenta las siguientes características:

CHIP	ESP12E
Voltaje de entrada	5V
Voltaje de salida de los pines	3.3V
corriente nominal por pin	12mA
frecuencia de procesador	80MHz
Memoria	4Mb
consumo de corriente en stand-by	80mA
ADC	10 bits
estándar de comunicación	802.11 b/g/n

Tabla3.8: Características NodeMCU

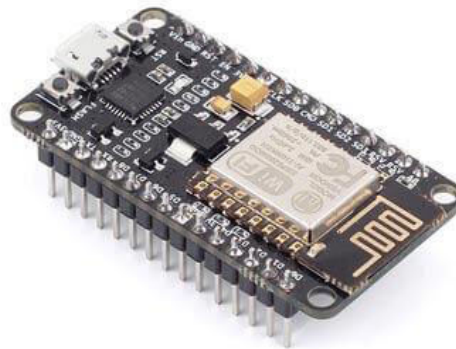


Figura3.9:NodeMCU ESP8266

3.3.2.-DISTRIBUCION DE PINES

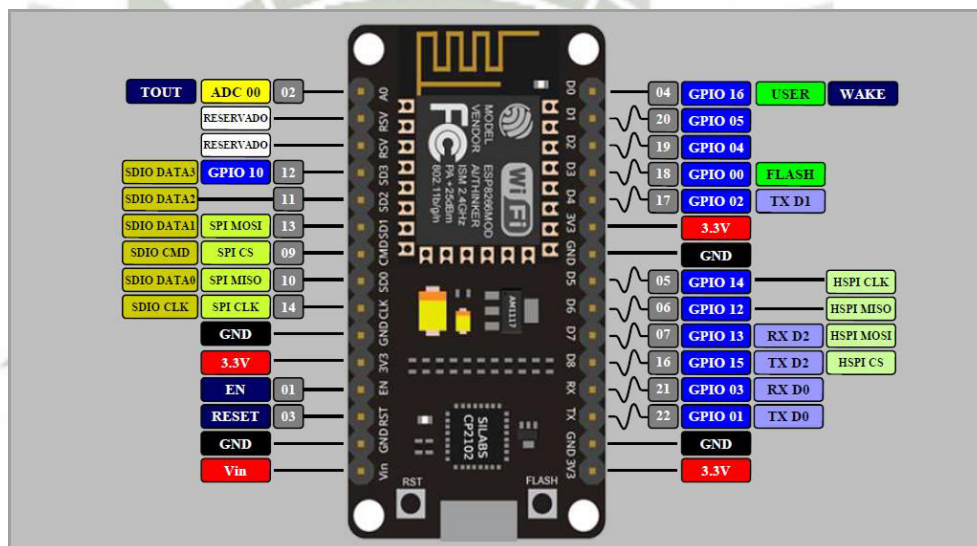


Figura3.10: Distribución de pines nodeMcu

3.4.-NODE JS

Node.js es un entorno JavaScript del lado del servidor, basado en eventos. Node ejecuta JavaScript utilizando el motor V8, desarrollado por Google para uso de su navegador Chrome. Aprovechando el motor V8 permite a Node proporcionar un entorno de ejecución del lado del servidor que compila y ejecuta JavaScript a velocidades increíbles. El aumento de velocidad es importante debido a que V8 compila JavaScript en código de

máquina nativo, en lugar de interpretarlo o ejecutarlo como bytecode. Node es de código abierto, y se ejecuta en Mac OS X, Windows y Linux.³⁷

3.5.-NODE-RED

Es una herramienta de visualización open-source creada por el equipo de tecnologías emergentes de IBM que nos permite interconectar todos nuestros elementos en internet para realizar el cableado de los dispositivos de hardware en conjunto. Estos elementos pueden ser desde dispositivos hardware a APIs (Interfaz de Programación de Aplicaciones) representa la capacidad de comunicación entre componentes de software.

Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla o servicios online.

Nos provee de un editor de flujo basada en un navegador web que hace que sea más fácil cablear los flujos utilizando los nodos de la amplia gama de la paleta que trae el entorno de trabajo para los desarrollos en Node Red. Además una biblioteca incorporada le permite guardar funciones útiles, plantillas que pueden ser reutilizadas para otros desarrollos.³⁸

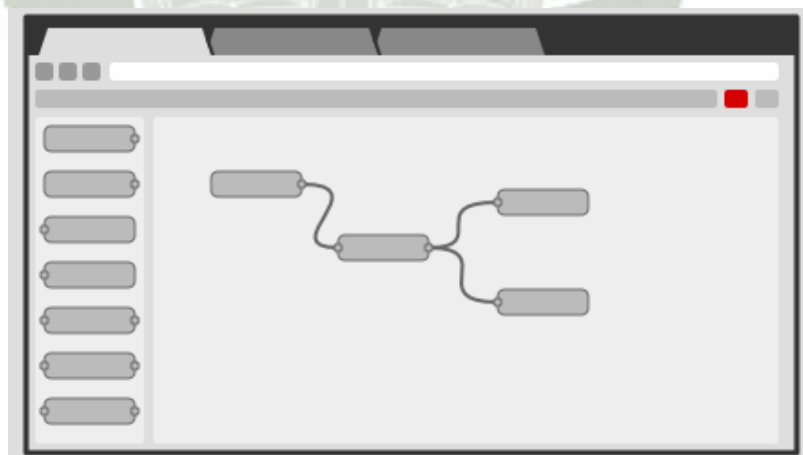


Figura3.11: Node red

Fuente:<https://nodered.org/>

³⁷ Fuente: <https://nodejs.org/es/>

³⁸ Fuente:<https://nodered.org/>

3.6.-FIREBASE

Firestore es una plataforma fundada en 2011 y comprada posteriormente por Google en 2014, que permite crear y desarrollar aplicaciones para dispositivos móviles de alta calidad.

Firestore está disponible tanto para Android, como para iOS o web. Además, contiene varias y diversas funcionalidades que todo desarrollador puede combinar y adaptar a sus propias necesidades.



Figura3.12: Logo Firebase

Fuente:<https://firebase.google.com/?hl=e>

Firestore dispone de diferentes funcionalidades, entre las que se encuentran:

Base de datos en tiempo real: una base de datos gestionada por Google basada en JSON que nos permite, mediante eventos, tener funcionalidades en tiempo real. También nos permite llevar una gestión automática de los datos en el caso de que la aplicación se encuentre sin conexión, sincronizando todos los cambios una vez haya recuperado dicha conectividad.

- Sistema de autenticación de usuarios tanto por email/contraseña como por otros sistemas como Facebook, twitter, google, github, etc.
- Cloud Messaging para el envío de notificaciones push a los dispositivos de una forma sencilla.

- Un sistema de almacenamiento y sincronización de ficheros con el dispositivo.
- Un sistema de hosting estático, ideal para realizar páginas estáticas o utilizarlo a modo de CDN de la App.
- Sistema de reporting de errores.
- Laboratorio de pruebas, para realizar pruebas en los dispositivos.
- Funciones lambda basadas en NodeJS para realizar mediante eventos ciertas operaciones.
- Conexión con AdMob para la monetización de la aplicación.
- Configuración remota de la aplicación, pudiendo cambiar dinámicamente funcionalidades de la misma.³⁹

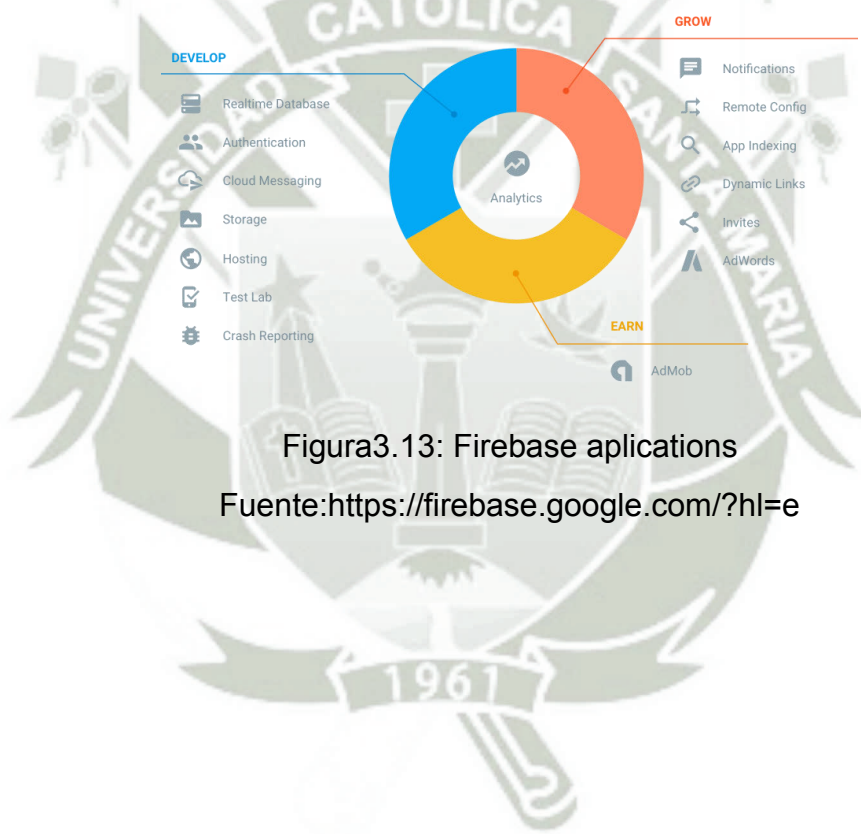


Figura3.13: Firebase aplicaciones

Fuente:<https://firebase.google.com/?hl=e>

³⁹ Fuente:<https://openwebinars.net/blog/que-es-firebase-de-google/>

3.7.-INSTALACION DE NODE RED

Para la correcta instalación de node-red necesitamos:

- Tener instalado node.js



Figura3.14: Descarga de node.js

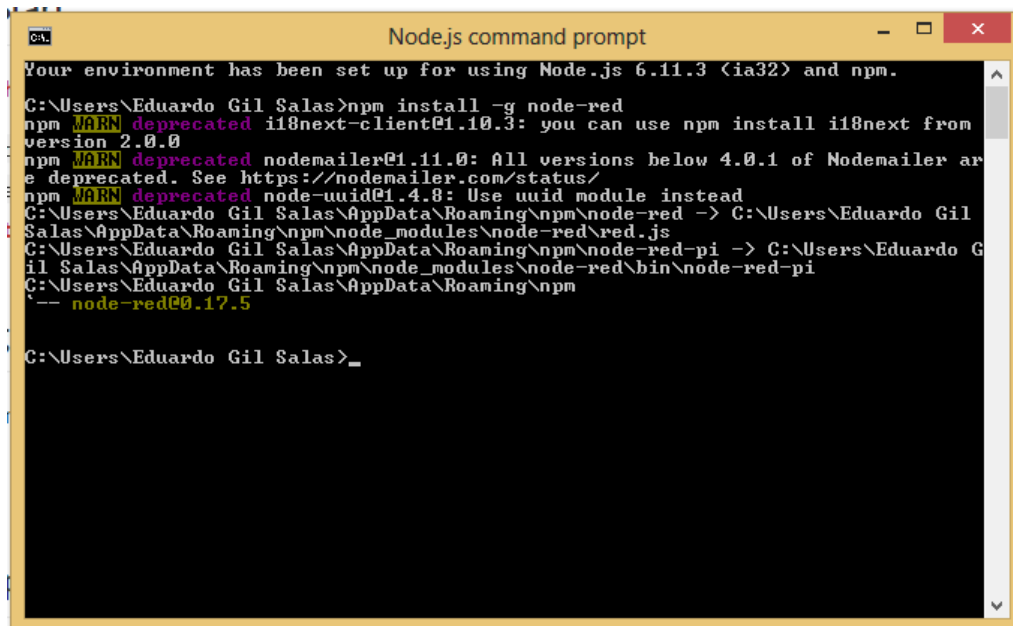
Fuente: <https://nodejs.org/es/>

- Tener instalado git bash (opcional), también podemos trabajar directamente con el node.js command prompt.
- Necesitamos también tener instalado npm (node package manager)
- Copiaremos el siguiente comando ya sea en node.js command prompt o en git bash, omitiendo (sudo) ya que este solo es usando si se va a instalar en linux/

```
sudo npm install -g --unsafe-perm node-red
```

Figura3.15:Cod. instalacion node-red

Fuente:<https://nodered.org/>



```

Node.js command prompt
Your environment has been set up for using Node.js 6.11.3 (ia32) and npm.

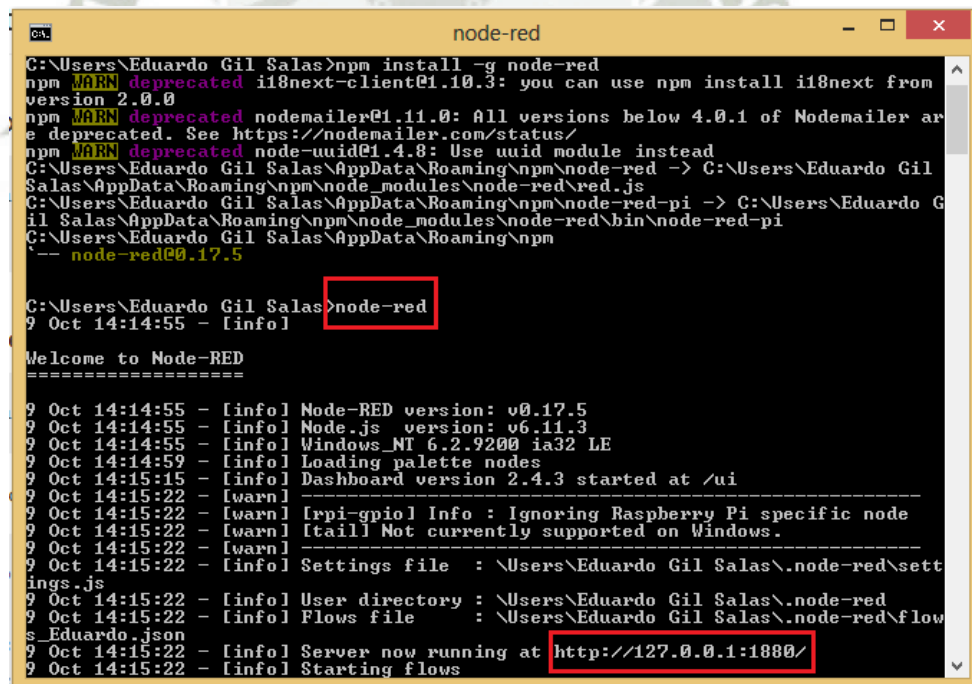
C:\Users\Eduardo Gil Salas>npm install -g node-red
npm WARN deprecated i18next-client@1.10.3: you can use npm install i18next from
version 2.0.0
npm WARN deprecated nodemailer@1.11.0: All versions below 4.0.1 of Nodemailer ar
e deprecated. See https://nodemailer.com/status/
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\node-red -> C:\Users\Eduardo Gil
Salas\AppData\Roaming\npm\node_modules\node-red\red.js
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\node-red -> C:\Users\Eduardo G
il Salas\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm
-- node-red@0.17.5

C:\Users\Eduardo Gil Salas>_
    
```

Figura3.16:Instalacion de node-red

Fuente:propia

- Una vez instalado node-red escribiremos el siguiente comando en el node.js command prompt para poder dar inicio a node-red.



```

node-red
C:\Users\Eduardo Gil Salas>npm install -g node-red
npm WARN deprecated i18next-client@1.10.3: you can use npm install i18next from
version 2.0.0
npm WARN deprecated nodemailer@1.11.0: All versions below 4.0.1 of Nodemailer ar
e deprecated. See https://nodemailer.com/status/
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\node-red -> C:\Users\Eduardo Gil
Salas\AppData\Roaming\npm\node_modules\node-red\red.js
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\node-red -> C:\Users\Eduardo G
il Salas\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm
-- node-red@0.17.5

C:\Users\Eduardo Gil Salas>node-red
9 Oct 14:14:55 - [info]
Welcome to Node-RED
=====
9 Oct 14:14:55 - [info] Node-RED version: v0.17.5
9 Oct 14:14:55 - [info] Node.js version: v6.11.3
9 Oct 14:14:55 - [info] Windows_NT 6.2.9200 ia32 LE
9 Oct 14:14:59 - [info] Loading palette nodes
9 Oct 14:15:15 - [info] Dashboard version 2.4.3 started at /ui
9 Oct 14:15:22 - [warn]
9 Oct 14:15:22 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
[tail] Not currently supported on Windows.
9 Oct 14:15:22 - [info] Settings file : \Users\Eduardo Gil Salas\.node-red\sett
ings.js
9 Oct 14:15:22 - [info] User directory : \Users\Eduardo Gil Salas\.node-red
9 Oct 14:15:22 - [info] Flows file : \Users\Eduardo Gil Salas\.node-red\flow
s\Eduardo.json
9 Oct 14:15:22 - [info] Server now running at http://127.0.0.1:1880/
9 Oct 14:15:22 - [info] Starting flows
    
```

Figura3.17:Ejecucion de node-red

Fuente:propia

- Copiaremos la siguiente dirección en el buscador que estamos usando.

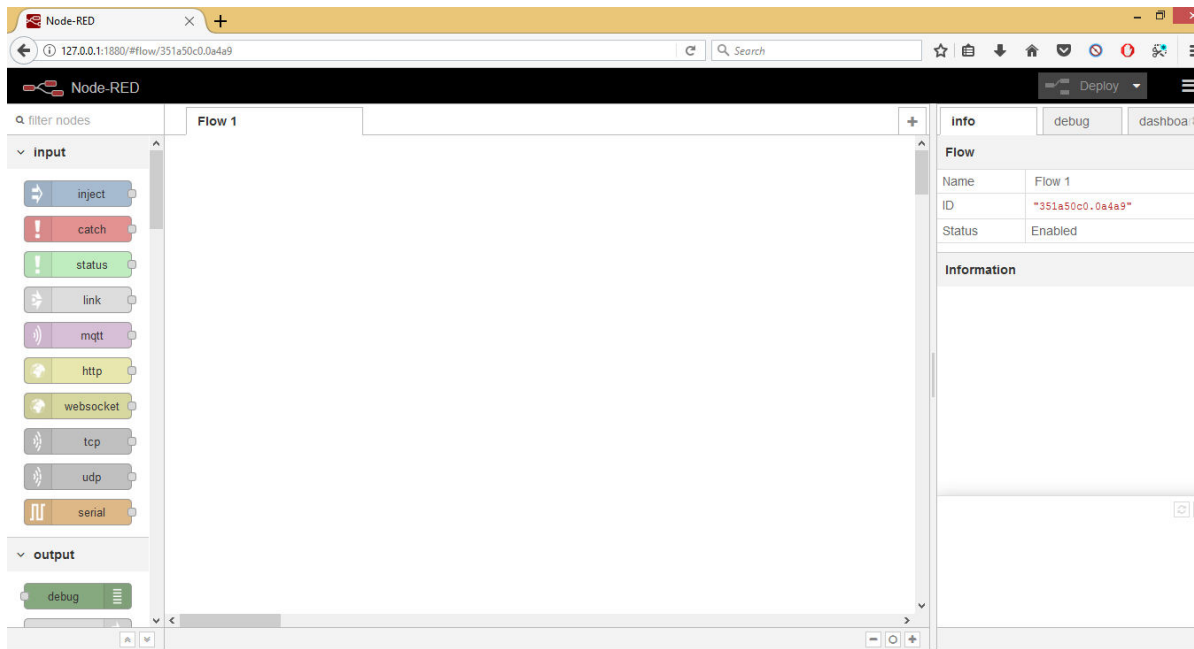


Figura3.18:Entorno de trabajo Node-red

Fuente:propia

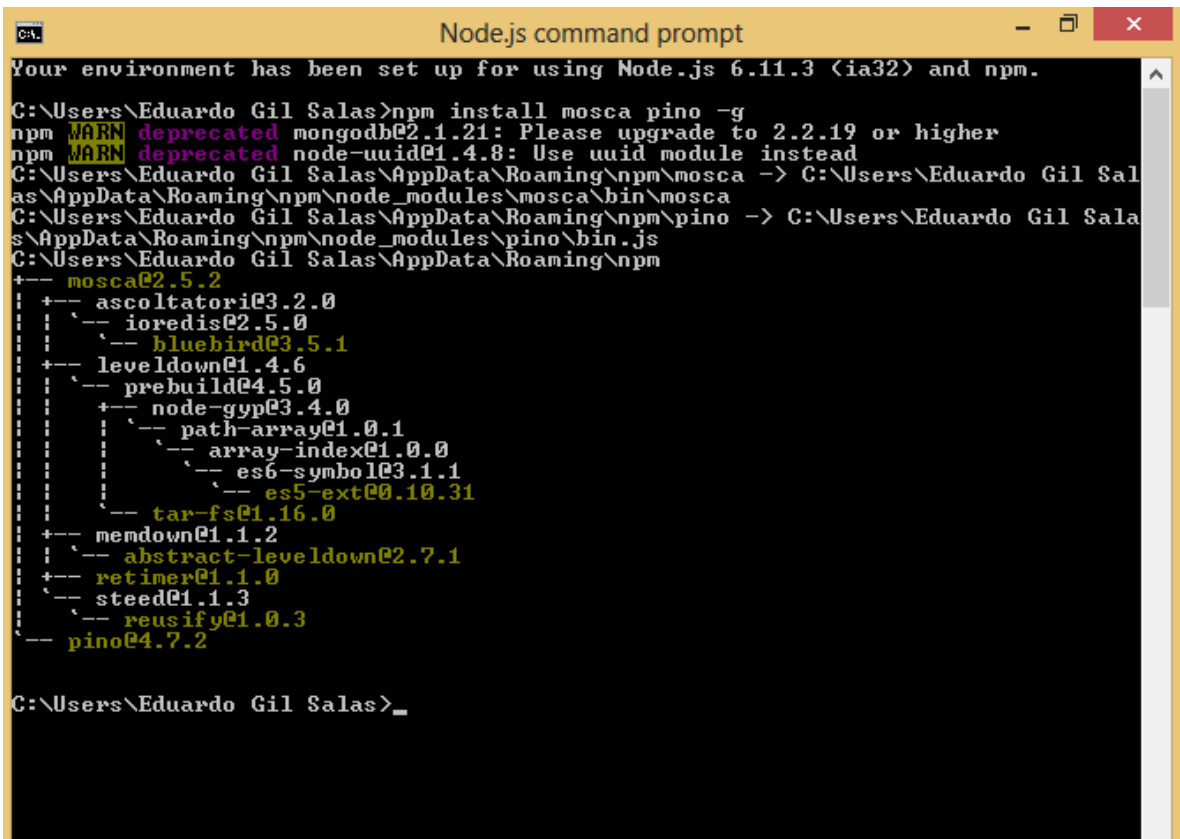
3.8.-INSTALACION DE MOSCA MQTT

- De la misma manera para la instalación de MOSCA MQTT necesitaremos:
- Node.js
- Tener instalado npm (gestor de paquetes de node)
- Tener instalado git-bash o podemos trabajar con el command prompt de node.js
- Escribiremos el siguiente código en el command prompt

```
npm install mosca pino -g
```

Figura3.19: Código de instalación mosca

Fuente:<https://www.npmjs.com/package/mosca>



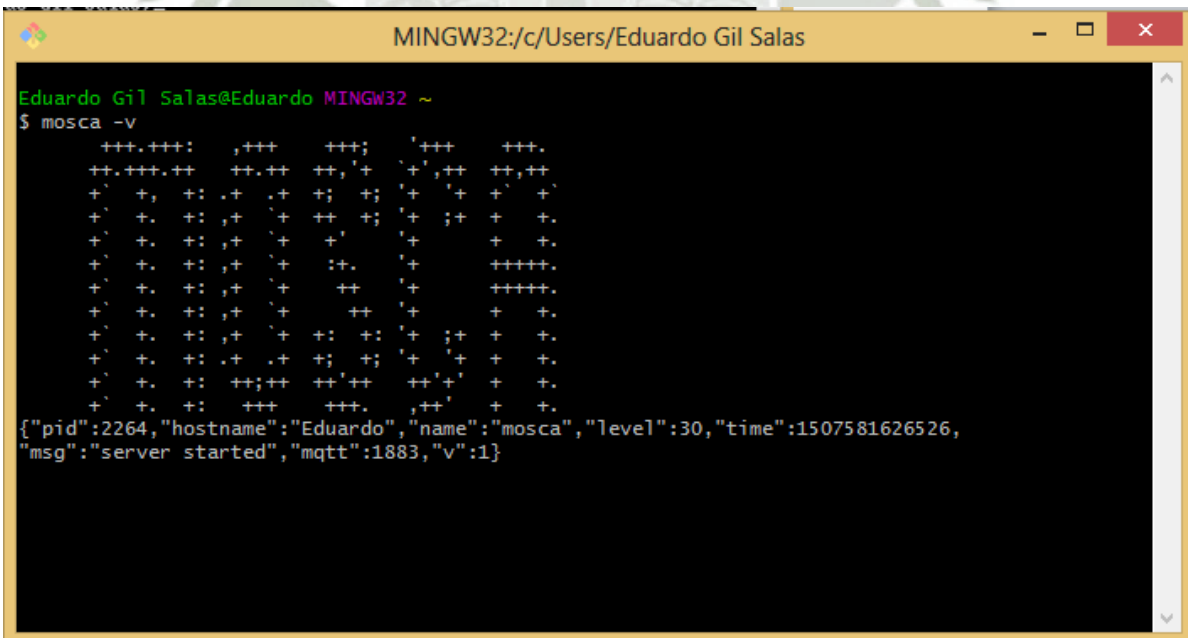
```

Node.js command prompt
Your environment has been set up for using Node.js 6.11.3 (ia32) and npm.
C:\Users\Eduardo Gil Salas>npm install mosca pino -g
npm WARN deprecated mongodb@2.1.21: Please upgrade to 2.2.19 or higher
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\mosca -> C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\node_modules\mosca\bin\mosca
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\pino -> C:\Users\Eduardo Gil Salas\AppData\Roaming\npm\node_modules\pino\bin.js
C:\Users\Eduardo Gil Salas\AppData\Roaming\npm
+-- mosca@2.5.2
|   +-- ascoltatori@3.2.0
|   |   '-- ioredis@2.5.0
|   |       '-- bluebird@3.5.1
|   +-- leveldown@1.4.6
|   |   '-- prebuild@4.5.0
|   |       +-- node-gyp@3.4.0
|   |       |   '-- path-array@1.0.1
|   |       |       '-- array-index@1.0.0
|   |       |           '-- es6-symbol@3.1.1
|   |       |               '-- es5-ext@0.10.31
|   |       '-- tar-fs@1.16.0
|   +-- memdown@1.1.2
|   |   '-- abstract-leveldown@2.7.1
|   +-- retimer@1.1.0
|   |   '-- steed@1.1.3
|   |       '-- reusify@1.0.3
|   '-- pino@4.7.2
C:\Users\Eduardo Gil Salas>_
    
```

Figura3.20: Instalación de mosca

Fuente: Propia

- Una vez instalado procedemos a correr el protocolo mosca mqtt



```

MINGW32:/c/Users/Eduardo Gil Salas
Eduardo Gil Salas@Eduardo MINGW32 ~
$ mosca -v
+++ .+++ : ,+++ +++; +++ +++
++ .+++ .++ ++.++ ++, + +',++ ++,++
+ +, +: .+ .+ +; +; + + + +
+ +. +: ,+ + ++ +; + ;+ + +
+ +. +: ,+ + + + + + + + +
+ +. +: ,+ + ++ + + + + + +
+ +. +: ,+ + ++ + + + + + +
+ +. +: ,+ + ++ +; +; + + + +
+ +. +: .+ .+ +; +; + + + +
+ +. +: +++; ++ ++'++ ++'++ + +
+ +. +: +++ +++ ++, ++ + +
{"pid":2264,"hostname":"Eduardo","name":"mosca","level":30,"time":1507581626526,
"msg":"server started","mqtt":1883,"v":1}
    
```

Figura: 3.21: Ejecución de mosca

Fuente: propia

3.9.-INSTALACION DE BASE DE DATOS FIREBASE

Existen dos maneras de poder instalar firebase en node red:

- Instalación mediante el prompt de node.js
- Instalación directamente desde node-red.

En este caso se optó por instalar directamente desde el entorno de trabajo de node-red. Tenemos que situarnos en el icono de opciones y seleccionar manage palette.

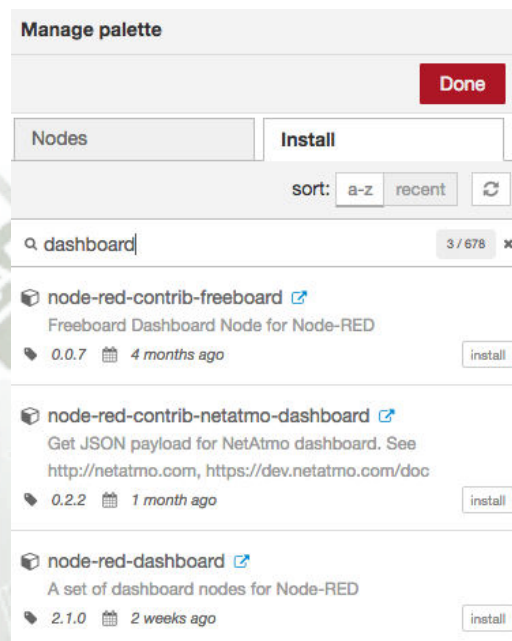


Figura3.22: manage palette

Fuente: propia

En la barra de búsqueda escribiremos firebase y daremos en instalar, esperaremos algunos minutos mientras se instala el paquete de nodos de firebase.

Finalmente terminada la instalación veremos que en la columna de nodos, estará ubicado casi al final.

3.10.-DIAGRAMA DE BLOQUES DEL SISTEMA DE MONITOREO



Diagrama de bloques 3.2: Sistema de monitoreo

3.11.-DESCRIPCION DEL DIAGRAMA DE BLOQUES

3.11.1.-Sensores

En esta etapa se encuentra la red de sensores con la que trabajaremos, algunos de estos sensores no necesitan alguna etapa de amplificación o filtrado; los valores que estos obtengan serán tratados por nuestra tarjeta de adquisición de datos.

3.11.2.-NodeMCU

Esta tarjeta se encargara de recibir y tratar los datos obtenidos por los sensores, esta tarjeta se eligió debido a que posee comunicación Wifi.

3.11.3.-MQTT Mosca

Este protocolo está orientado a la comunicación de sensores, debido a que consume muy poco ancho de banda y puede ser utilizado en la mayoría de los dispositivos empotrados con pocos recursos (CPU, RAM).

3.11.4.-Node Red

Optamos trabajar en node red- debido a que se trata de una plataforma de para internet de las cosas y es open-source.

3.11.5.-Base de datos

Después de haber realizado la adquisición, tratamiento y visualización de nuestros datos en nuestra plataforma de node-red, se procede a crear una base de datos para poder tener un histórico de todos nuestros datos capturados por la red de sensores.

3.12.-PRINCIPIO DE FUNCIONAMIENTO SENSOR DHT22

El envío de datos se realiza por un único pin digital en forma serial, en este caso se conecta a uno de los pines digitales del microcontrolador. El sensor transmite 40 bits de datos, los cuales 16 bits son para la humedad, 16 bits para la temperatura y 8 bits de paridad.

Una transmisión de datos completa es de 40 bits. Donde obtenemos la temperatura y la humedad. Ejemplo: Recibimos 40 bits:

0011 0101 0000 0000 0001 1000 0000 0000 0100 1101

High humidity 8 + Low humidity 8 + High temp. 8 + Low temp. = 8 Parity bit
Calculando :

0011 0101+0000 0000+0001 1000+0000 0000= 0100 1101

Datos correctos recibidos :

Humedad : 0011 0101 = 35H = **53% RH**

Temperatura : 0001 1000 = 18H = **24°C**

El microcontrolador externo y el microcontrolador que lleva integrado el sensor, se hablan entre sí de la siguiente manera:

Se inicia la comunicación.

El sensor responde estableciendo un nivel bajo de 80us y un nivel alto de 80us.

El sensor envía 5 bytes con la información de temperatura y humedad.

3.13.-DIAGRAMA DE FLUJOS TEMPERATURA Y HUMEDAD

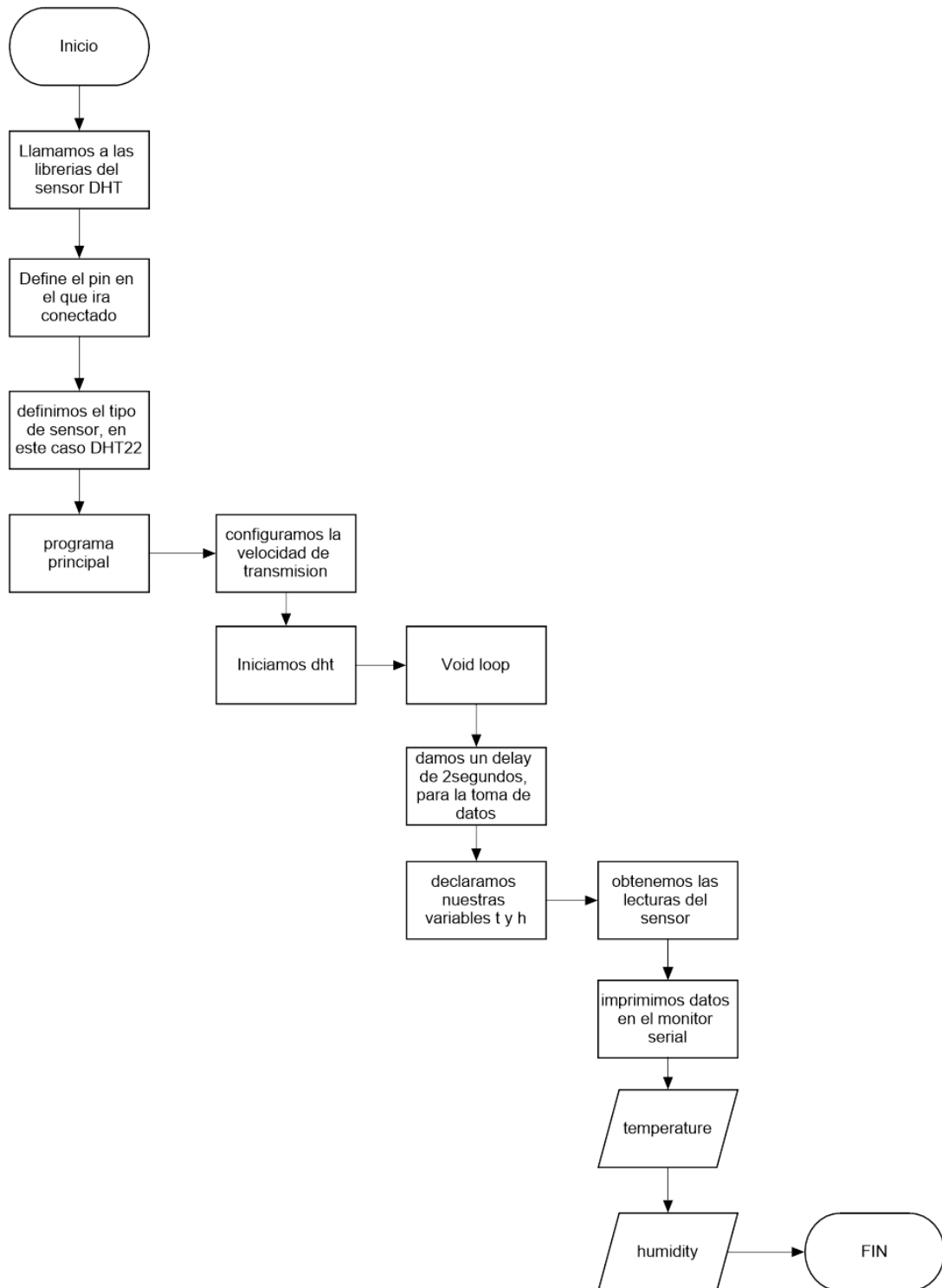


Diagrama de flujos3.1: humedad y temperatura

3.14.-RUTINA DE MEDICION DE TEMPERATURA Y HUMEDAD

```
void setup() {  
  Serial.begin(9600);  
  dht.begin();  
}  
  
void loop() {  
  delay(200);  
  
  float h = dht.readHumidity();  
  float t = dht.readTemperature();  
  float f = dht.readTemperature(true);  
  Serial.print("Humedad: ");  
  Serial.print(h);  
  Serial.print(" %");  
  Serial.print("Temperatura: ");  
  Serial.print(t);  
  Serial.print(" *C ");  
  Serial.print("temperatura en Grados F: ");  
  Serial.print(f);  
  Serial.println(" *F");  
}
```

3.15.-MUESTRA DE DATOS POR MONITOR SERIE

En la figura3.21 se muestran los datos obtenidos por el sensor dht22.

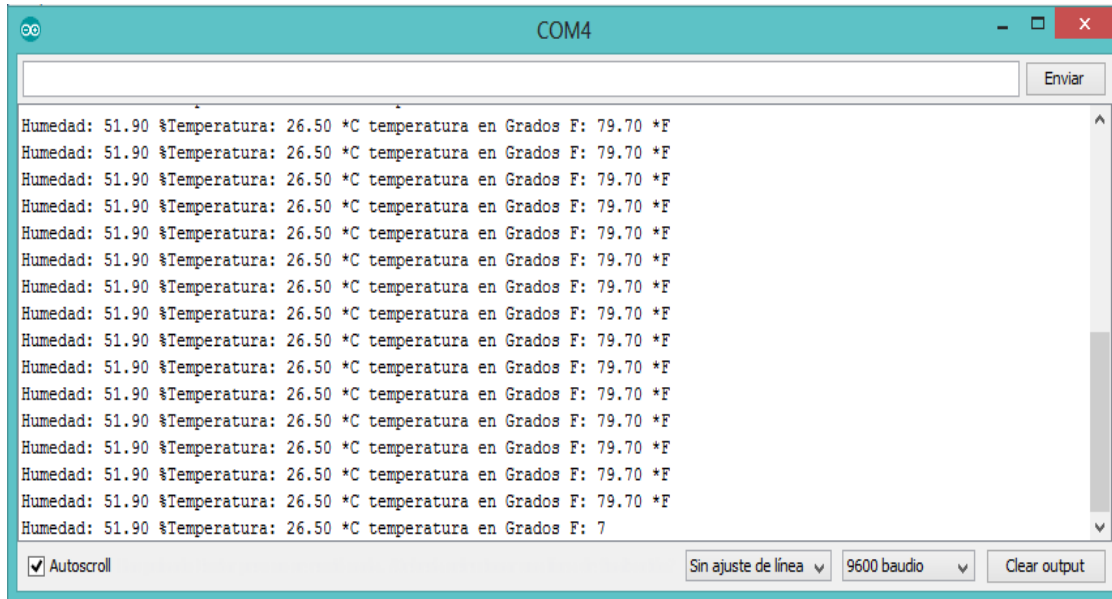


Figura3.23: Datos obtenidos de dht22

Fuente: propia

En la gráfica que se muestra a continuación se observa el cambio de temperatura y humedad con respecto al tiempo.

Donde:

Línea verde=temperatura en grados fahrenheit

Línea roja= temperatura en grados centígrados

Línea azul = humedad relativa

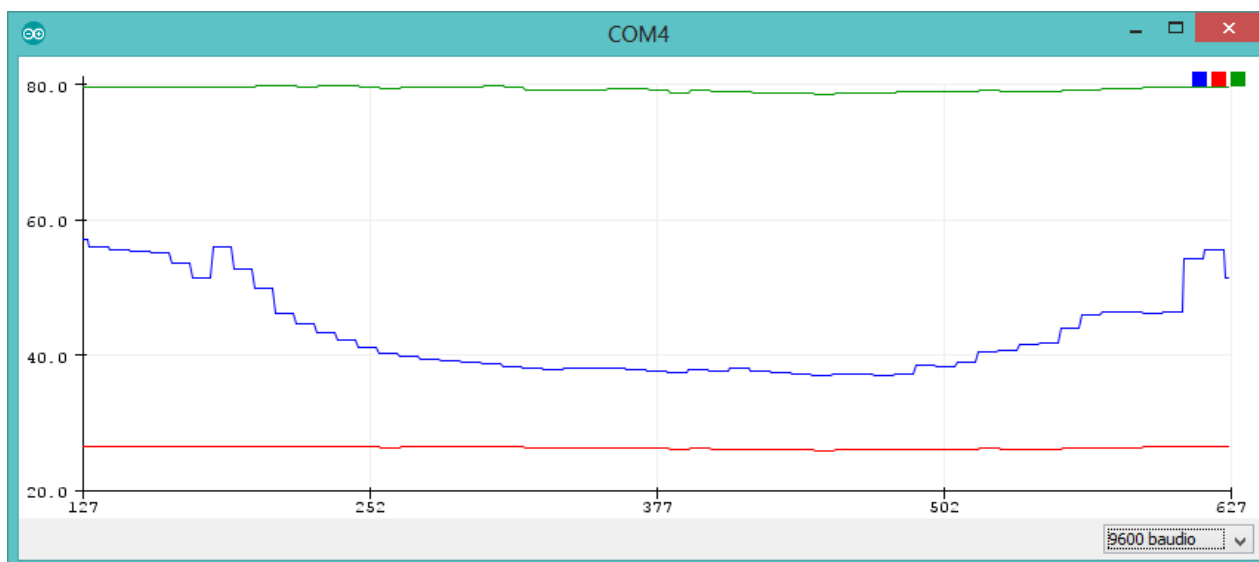
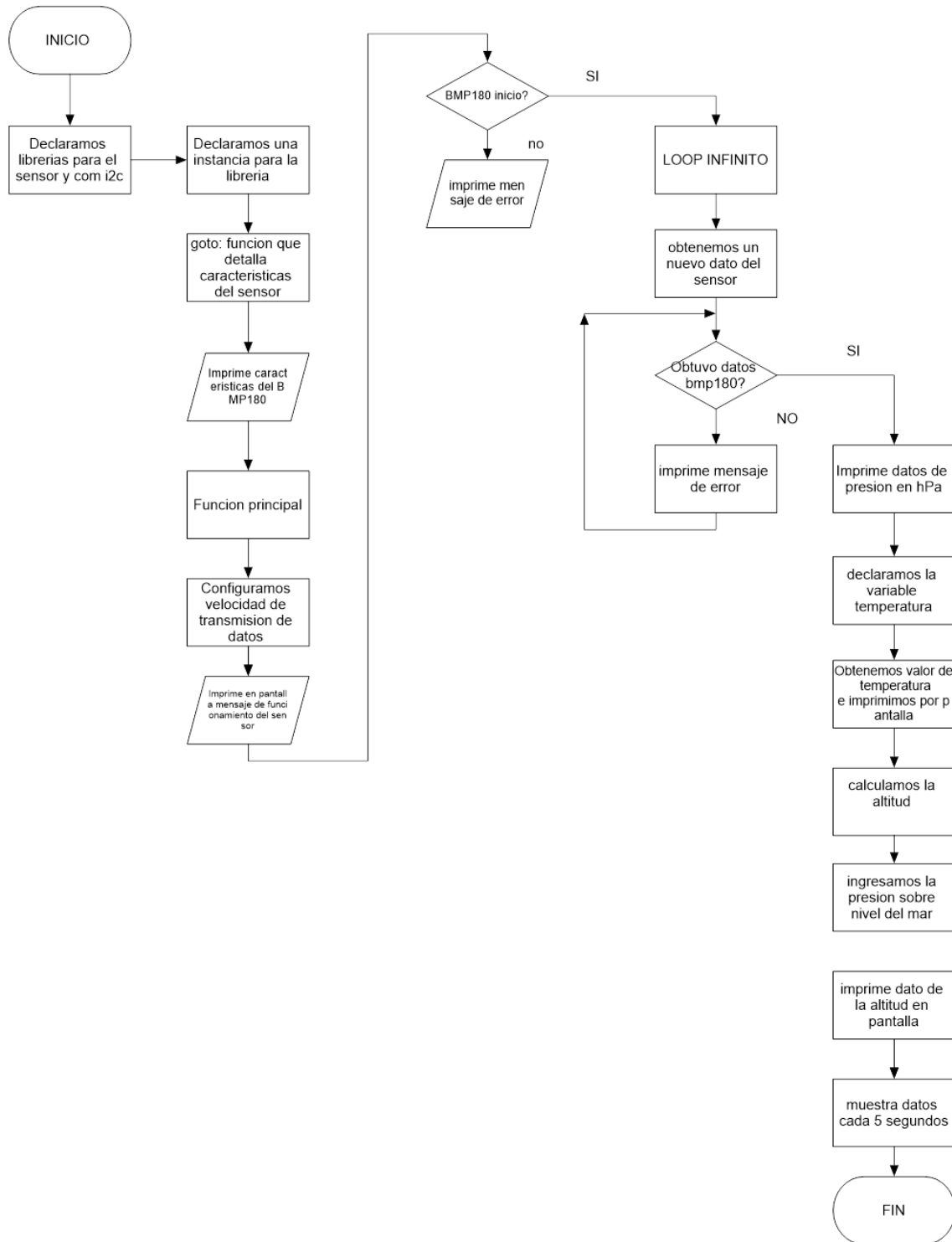


Figura3.24: Grafica de humedad/temperatura/temperatura en grados F

Fuente: propia



3.16.-DIAGRAMA DE FLUJOS PRESION BAROMETRICA



Flujograma3.2: Presión

3.17. PRINCIPIO DE FUNCIONAMIENTO

3.17.1.-COEFICIENTES DE CALIBRACION

La EEPROM de 176 bits está dividida en 11 palabras de 16 bits cada una. Estos contienen 11 coeficientes de calibración. Cada módulo sensor tiene coeficientes individuales. Antes del primer cálculo de temperatura y presión, el maestro lee los datos EEPROM. La comunicación de datos puede verificarse verificando que ninguna de las palabras tenga el valor 0 o 0xFFFF.

Parameter	BMP180 reg adr	
	MSB	LSB
AC1	0xAA	0xAB
AC2	0xAC	0xAD
AC3	0xAE	0xAF
AC4	0xB0	0xB1
AC5	0xB2	0xB3
AC6	0xB4	0xB5
B1	0xB6	0xB7
B2	0xB8	0xB9
MB	0xBA	0xBB
MC	0xBC	0xBD
MD	0xBE	0xBF

Figura3.25: Coeficientes de calibración bmp180

Fuente: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>

3.17.2.-CALCULO DE ALTITUD ABSOLUTA

Con la presión medida p y la presión al nivel del mar p_0 , $p_0 = 1013.25\text{hPa}$, la altitud en metros se puede calcular con la fórmula barométrica internacional:

$$\text{altitud} = 44330 * \left(1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5.255}} \right)$$

Figura3.26: Calculo de altitud absoluta

Fuente: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>

Por lo tanto, un cambio de presión de $\Delta p = 1\text{hPa}$ corresponde a 8.43m a nivel del mar.

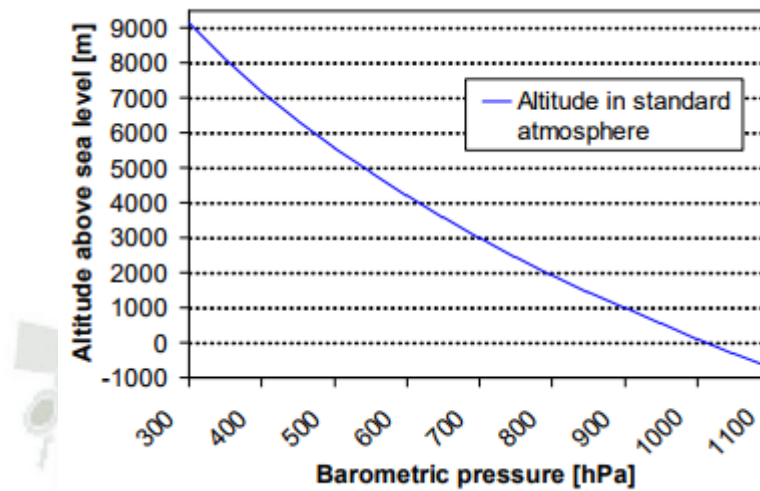


Figura3.27: Variación de altitud con respecto a la presión

Fuente: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>

3.17.3.-CALCULO DE PRESION A NIVEL DE MAR

Con la presión medida p y la altitud absoluta, se puede calcular la presión al nivel del mar:

$$p_0 = \frac{p}{\left(1 - \frac{\text{altitude}}{44330}\right)^{5.255}}$$

Figura3.28: Calculo de presión a nivel del mar

Fuente: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>

Por lo tanto, una diferencia en altitud de $\Delta\text{altitude} = 10\text{m}$ corresponde a un cambio de presión de 1.2hPa a nivel del mar.

3.18.-RUTINA DE MEDICION DE PRESION Y ALTITUD

```
void loop() {
    float tempe = bmp.readTemperature();
    double presion = (bmp.readPressure())*0.01;
    double alt = bmp.readAltitude();

    //Leemos los valores del sensor y sacamos la temperatura por el
    monitor serie
    Serial.print("Temperatura.BMP180 = ");
    Serial.print(tempe);
    Serial.println(" *C");

    //Leemos los valores del sensor y sacamos la presión atmosférica por
    el monitor serie
    Serial.print("Presión = ");
    Serial.print(presion);
    Serial.println("mBar");

    Serial.print("Altitud = ");
    Serial.print(alt);
    Serial.println(" metros");

    Serial.println(); //Espaciado entre lecturas
    delay(500);      //Pausa de 500 mili segundos entre lecturas
}
```

3.19.- MUESTRA DE DATOS POR MONITOR SERIE

En la figura 3.29 se muestran los datos obtenidos por el sensor bmp180 al realizar la medición.

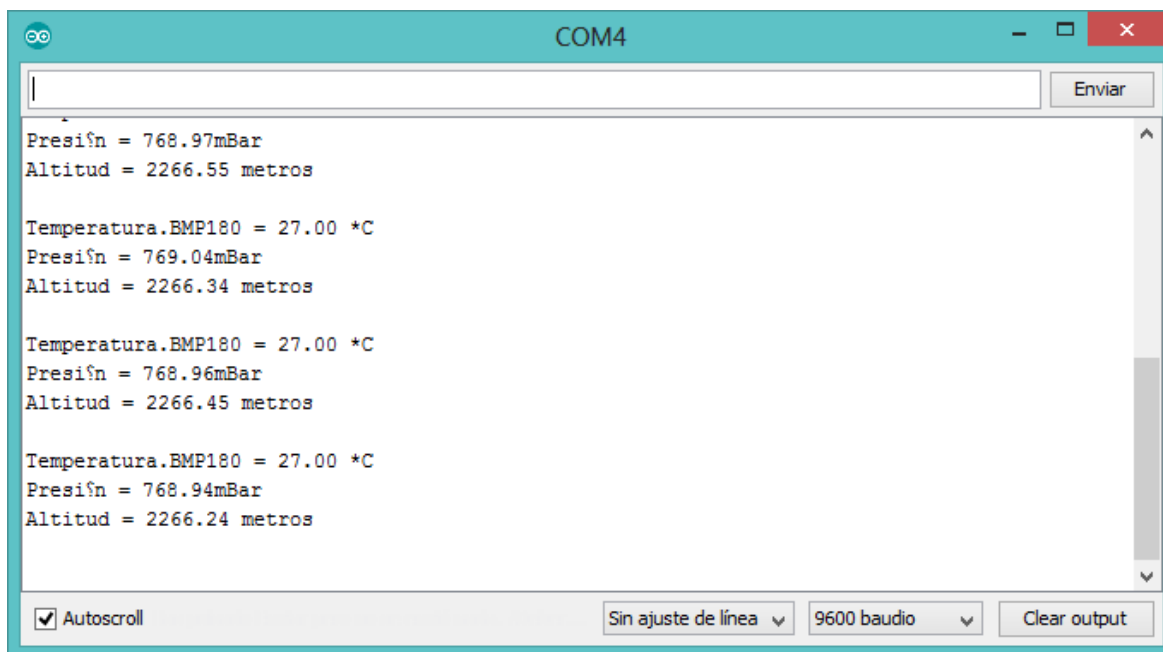
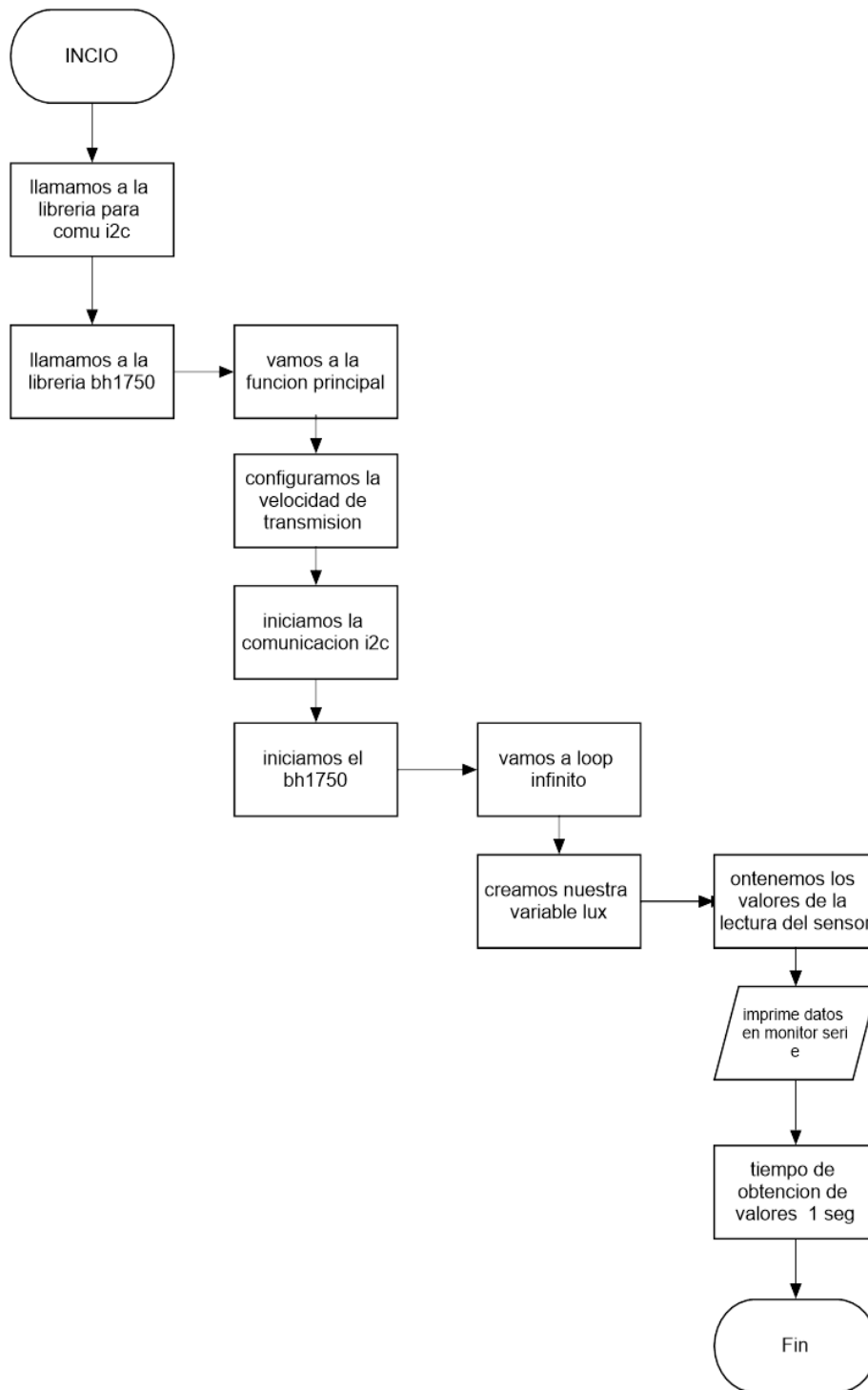


Figura3.29: Datos obtenidos por bmp180

Fuente: Propia

3.20.-DIAGRAMA DE FLUJOS MEDICION DE LUMINOSIDAD



Flujograma3.3: Luminosidad

3.21.- RUTINA DE ADQUISICION DE LUMINOSIDAD

```
void loop() {  
  
  uint16_t lux = luxometro.readLightLevel();  
  Serial.print("Luminosidad: ");  
  Serial.print(lux);  
  Serial.println(" lx");  
  delay(500);  
  
}
```

3.22.- MUESTRA DE DATOS POR MONITOR SERIAL

En la figura3.23 se muestran los datos obtenidos por nuestro sensor

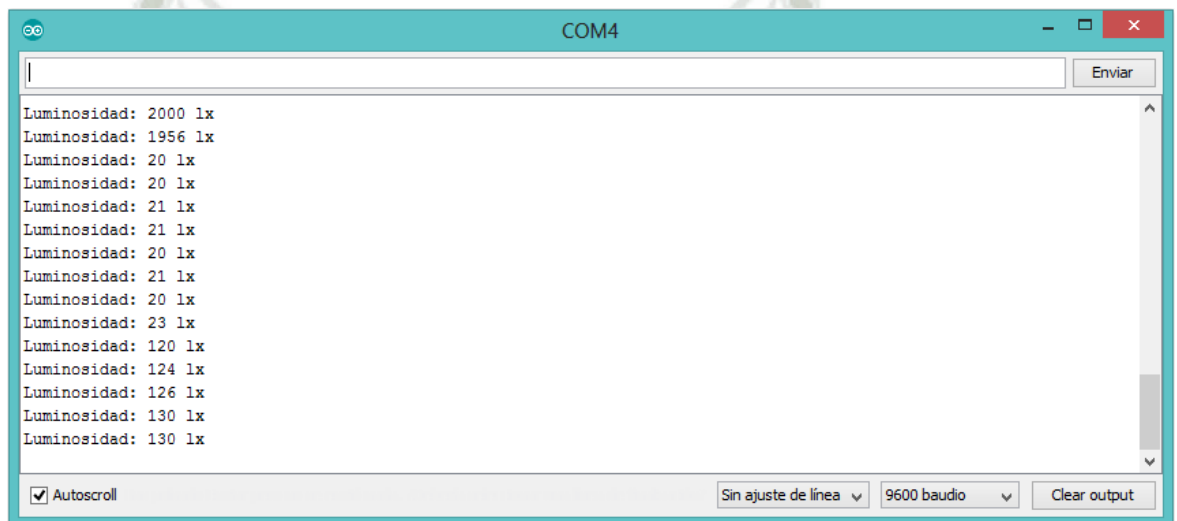


Figura3.30: Muestra de datos obtenidos por sensor bh1750

Fuente: Propia

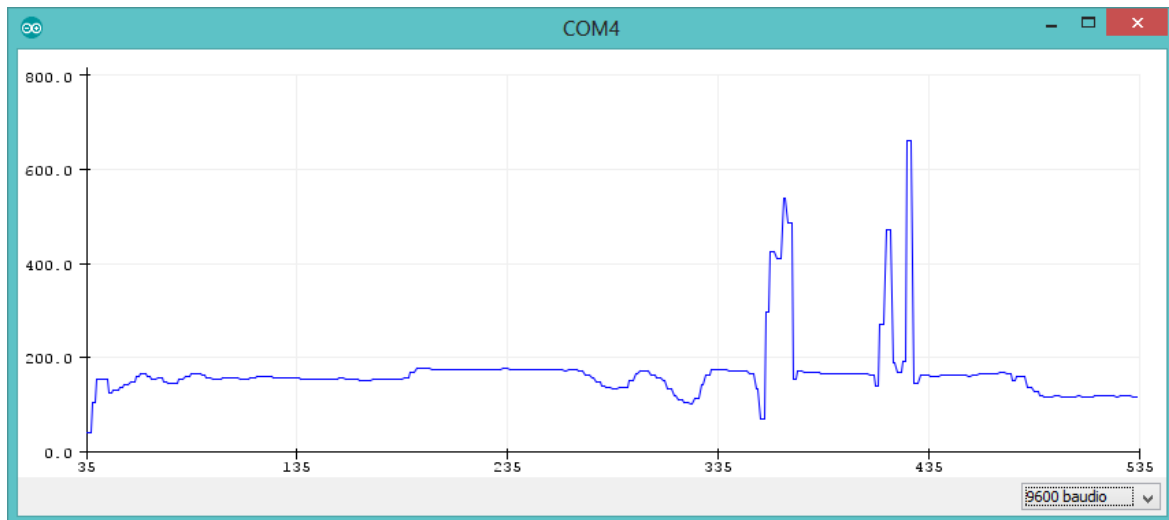


Figura3.31: Grafica de medición de luminosidad con respecto al tiempo

Fuente: Propia

3.23.- PRINCIPIO DE FUNCIONAMIENTO SENSOR UV

El índice UV es un sistema estándar sencillo para medir la radiación ultravioleta que llega desde el sol a la tierra que sirve como orientación para determinar el riesgo potencial para la salud. La Organización Mundial de la Salud publica una guía práctica sobre el índice UV en la que explica los riesgos para la salud de la radiación ultravioleta y propone algunas medidas de protección en función de su intensidad.



Figura3.32: Límites de radiación UV

Fuente: Organización Mundial de la Salud pública

Diagrama de bloques funcional del sensor ML8511

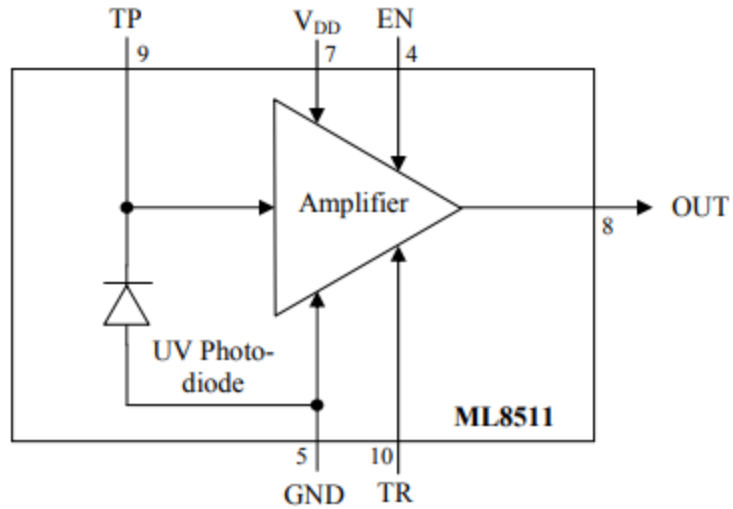


Figura3.33: Diagrama de bloques funcional del sensor ML8511

Fuente: <https://www.mcs.anl.gov/research/projects/waggle/downloads/datasheets/lightsense/ml8511.pdf>

El fabricante del módulo proporciona una tabla con los valores de salida con los que se alcanzan los diferentes niveles del índice UV.

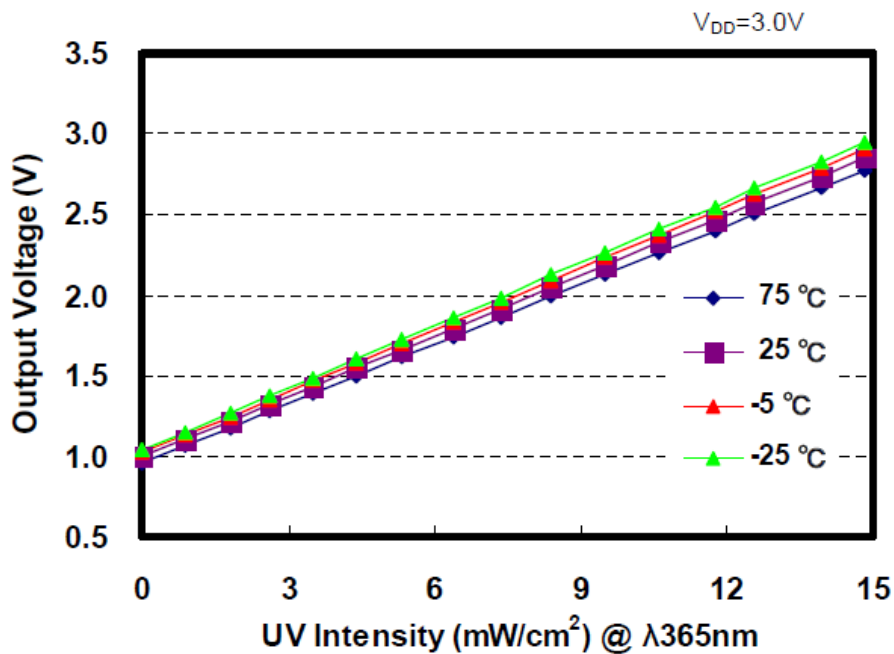


Figura3.34: Relacion de volt out con intensidad UV

Fuente: <https://learn.sparkfun.com/tutorials/ml8511-uv-sensor-hookup-guide>

Curva característica espectral

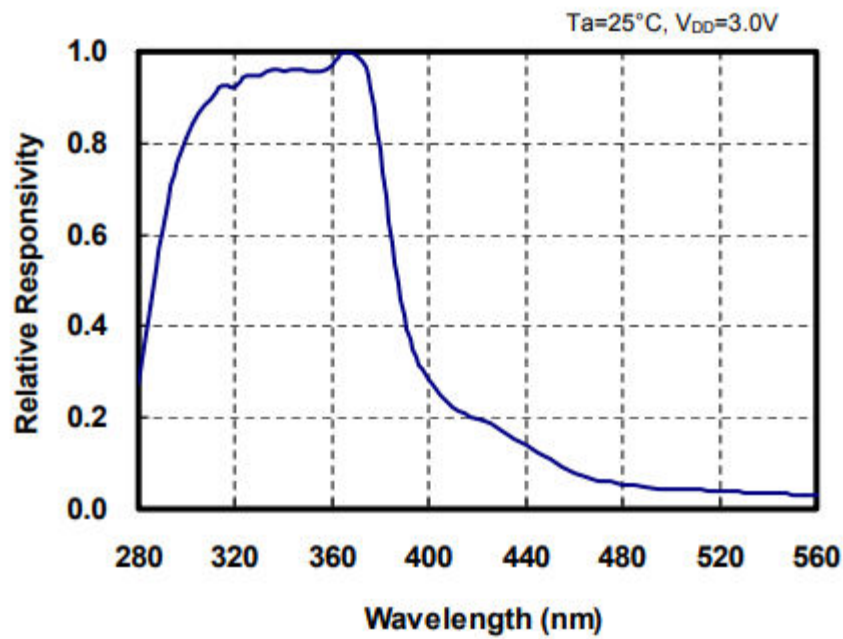
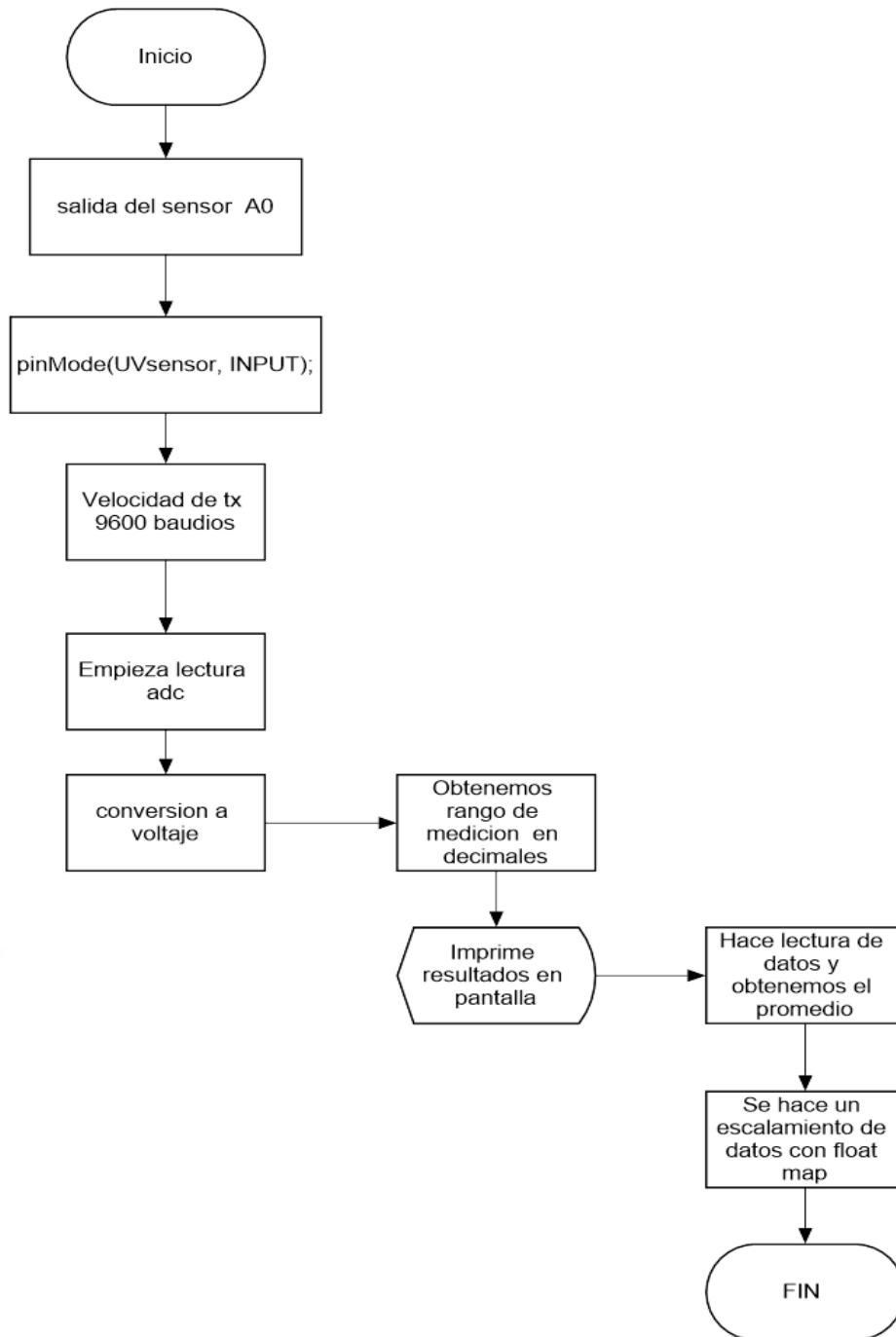


Figura3.35: Respuesta espectral del sensor ml8511

Fuente:<https://www.mcs.anl.gov/research/projects/waggle/downloads/datasheets/lightsense/ml8511.pdf>

3.24.-DIAGRAMA DE FLUJO MEDICION DE RADIACION UV



Flujograma3.4: Radiación uv

3.25.- RUTINA DE MEDICION DE RADIACION UV

```
void loop()
{
  int nivel_UV = averageAnalogRead(UVsensor);

  float voltaje_out = 3.3 * nivel_UV/1024;
  float intensidad_UV = mapfloat(voltaje_out, 0.99, 2.5, 0.0,
15.0);

  Serial.print(" intensidad uv: ");
  Serial.print(intensidad_UV);
  Serial.print(" mW/cm^2");
  Serial.println();
  delay(500);
}

// toma un promedio de lecturas obtenidas por el pin
//nos devuelve el promedio
int averageAnalogRead(int pinToRead)
{
  byte numerolecturas = 8;
  unsigned int valorejec = 0;

  for(int x = 0 ; x < numerolecturas ; x++)
    valorejec += analogRead(pinToRead);
  valorejec /= numerolecturas;

  return(valorejec);
}

float mapfloat(float x, float in_min, float in_max, float out_min,
float out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min;
}
```

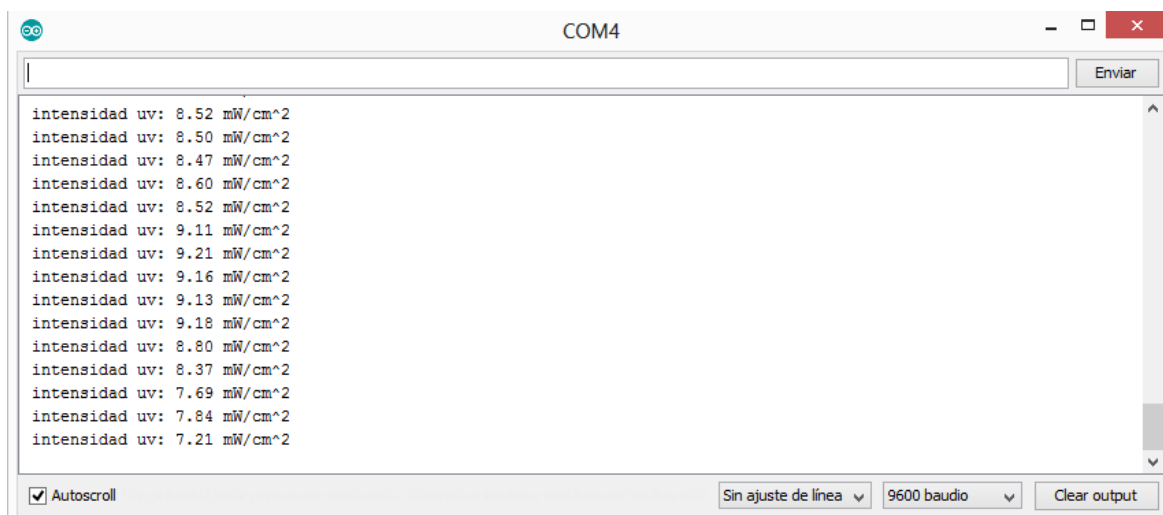


Figura3.36: Muestra de datos obtenidos por el sensor ML8511

Fuente: Propia

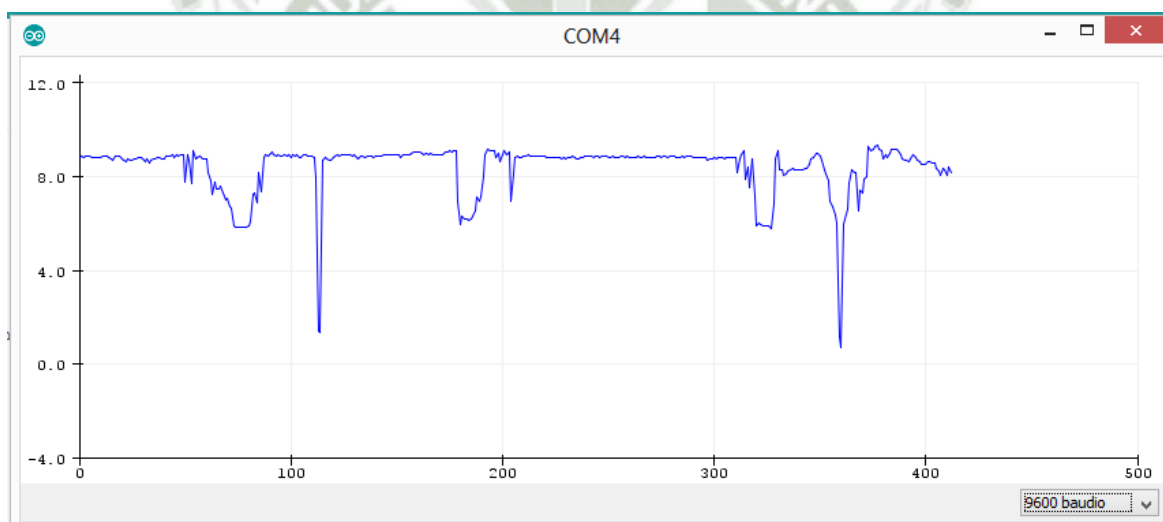
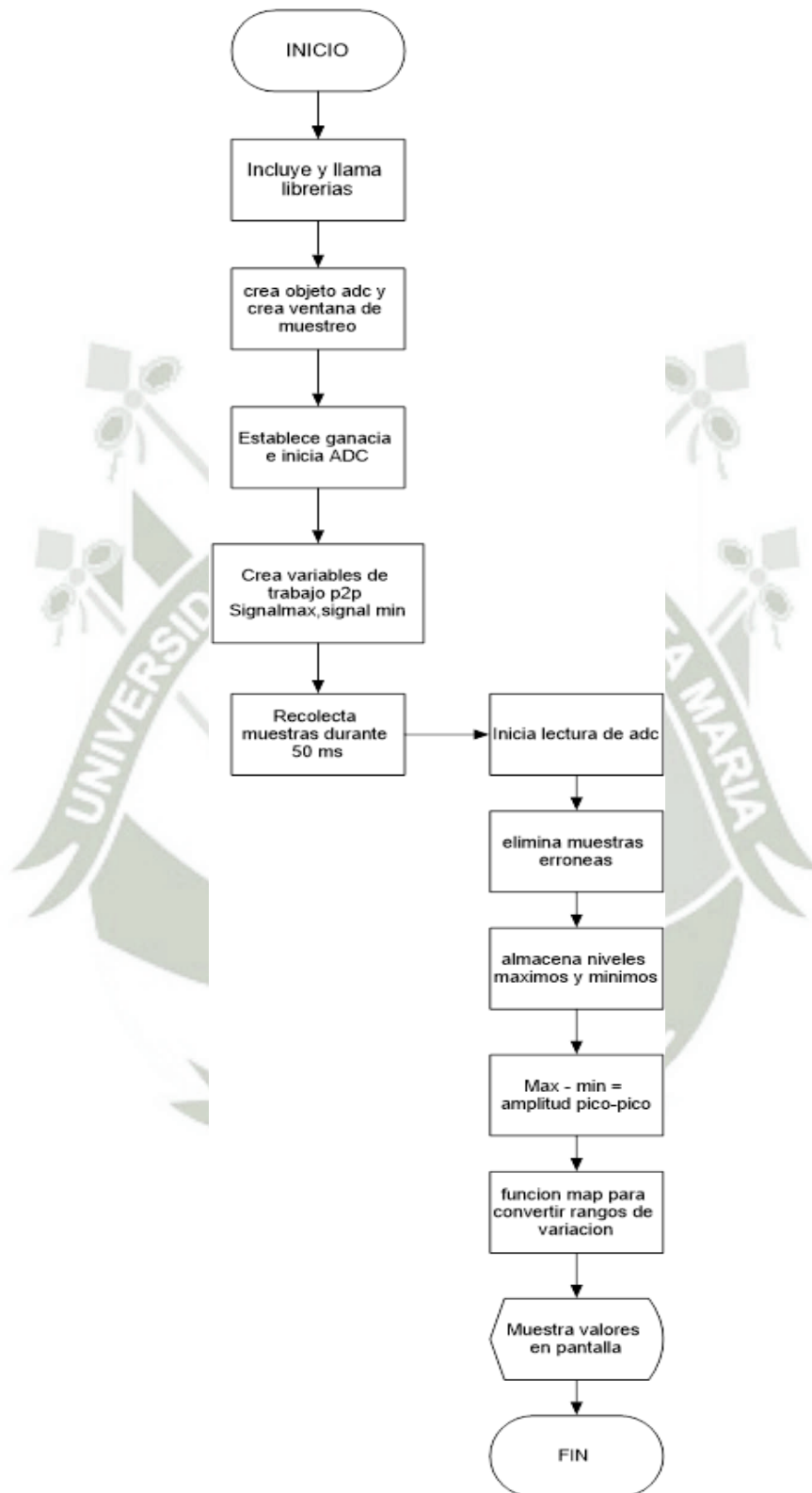


Figura3.37: Variación de muestras de radiación uv con respecto al tiempo

Fuente: Propia

3.26.-DIAGRAMA DE FLUJOS MEDICION DE NIVEL DE PRESION SONORA EN DB



Flujograma3.5: Medición de db

3.27.- LIMITES PERMISIBLES DE CONTAMINACION ACUSTICA

Para medir la contaminación sonora, se siguen las pautas contenidas en el Decreto Supremo N° 085-2003-PCM - Reglamento de Estándares Nacionales de Calidad Ambiental para Ruido (en adelante, Reglamento ECA Ruido), documento a través del cual se establecieron los estándares nacionales de calidad ambiental para ruido (ECA Ruido) y los lineamientos para no excederlos.

Los ECA Ruido son instrumentos de gestión ambiental prioritarios para prevenir y planificar el control de la contaminación sonora. Representan los niveles máximos de ruido en el ambiente que no deben sobrepasarse para proteger la salud humana, según cuatro zonas de aplicación:

Zonas de protección especial (es decir, áreas donde se encuentren ubicados establecimientos de salud, centros educativos, asilos y orfanatos)

- Zonas residenciales
- Zonas comerciales
- Zonas industriales

A cada zona de aplicación le corresponde un nivel de ruido para horarios diurnos y uno para horarios nocturnos, tal como se detalla en la siguiente tabla.

ZONAS DE APLICACIÓN	VALORES EXPRESADOS EN LAEQT ⁴	
	HORARIO DIURNO (07:01 A 22:00)	HORARIO NOCTURNO (22:01 A 07:00)
Zona de protección especial	50 dB	40 dB
Zona residencial	60 dB	50 dB
Zona comercial	70 dB	60 dB
Zona industrial	80 dB	70 dB

Tabla3.9: Límites permisibles de ruido

Fuente: https://www.oefa.gob.pe/?wpfb_dl=19087

3.28.- RUTINA DE MEDICION

```
void loop() {
  int16_t  adc2;
  unsigned long startMillis= millis(); //empieza ventana de muestreo
  unsigned int peakToPeak = 0; // nivel pico-pico
  unsigned int signalMax = 0;
  unsigned int signalMin = 65536;

  // recoleta datos durante 50 mS
  while (millis() - startMillis < sampleWindow)
  {
    adc2 = ads.readADC_SingleEnded(2);
    if (adc2 < 65536) // elimina muestras erróneas
    {
      if (adc2 > signalMax)
      {
        signalMax = adc2; // salva solo los niveles máximos
      }
      else if (adc2 < signalMin)
      {
        signalMin = adc2; // salva solo los mínimos niveles
      }
    }
  }
  peakToPeak = (signalMax - signalMin)/40;//Max - min = amplitud
  pico-pico
  float db = map( peakToPeak,20,900,40.50,130.50);

  Serial.println(db);
  delay(200);
}
```

3.29.- VISUALIZACION DE DATOS POR MONITOR SERIE

En la figura se muestran los datos obtenidos por el sensor de sonido

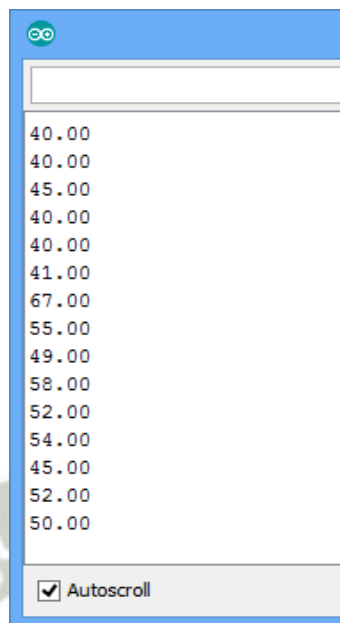


Figura3.38: Obtención de datos de sensor de sonido

Fuente: Propia

3.30.- GRAFICA DE DB CON RESPECTO AL TIEMPO

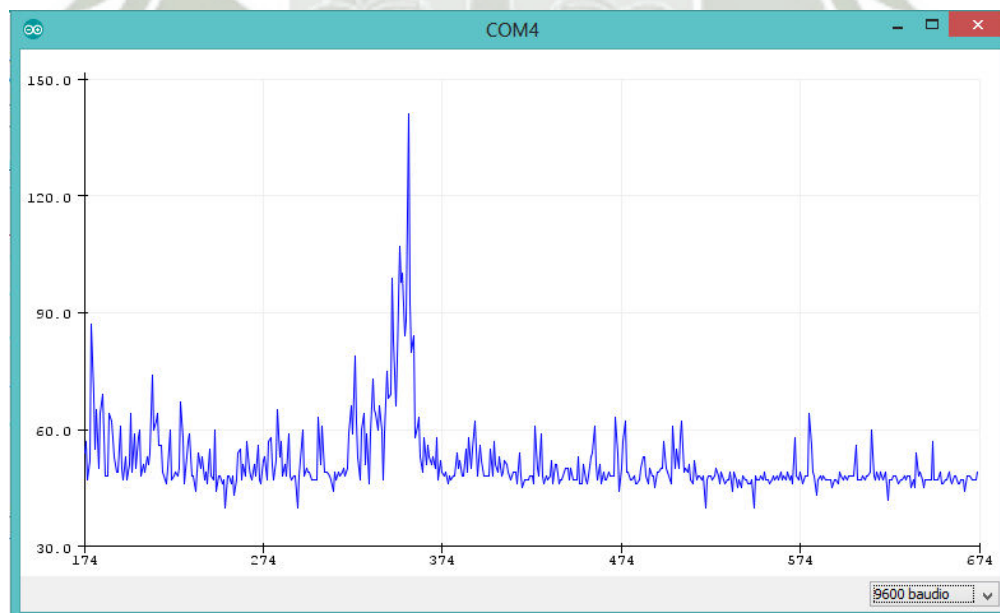
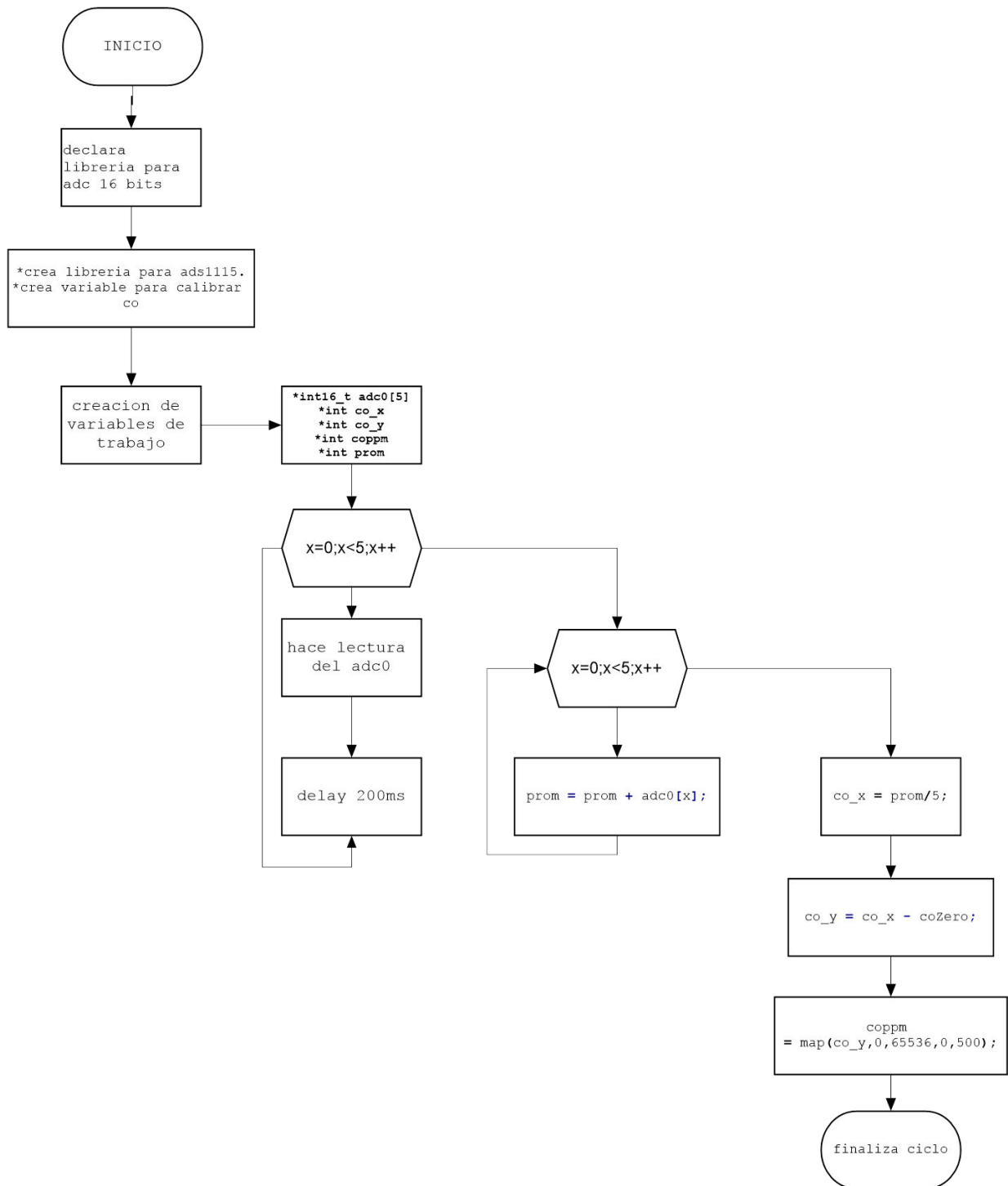


Figura3.39: Variación de db con respecto al tiempo

Fuente: Propia

3.31.-DIAGRAMA DE FLUJO MEDICION DE MONOXIDO DE CARBONO



Flujograma3.6: Medición de CO

3.32.- PRINCIPIO DE FUNCIONAMIENTO PRINCIPIO DE FUNCIONAMIENTO

La figura muestra las características de sensibilidad típicas del MQ-7 para varios gases.

Entre estas están:

- Temperatura: 20 °C.
- Humedad: 65%,
- concentración de O2 21%
- RL = 10kΩ
- Ro: resistencia del sensor a 100ppm CO en el aire limpio.

Rs: resistencia del sensor a varias concentraciones de gases.

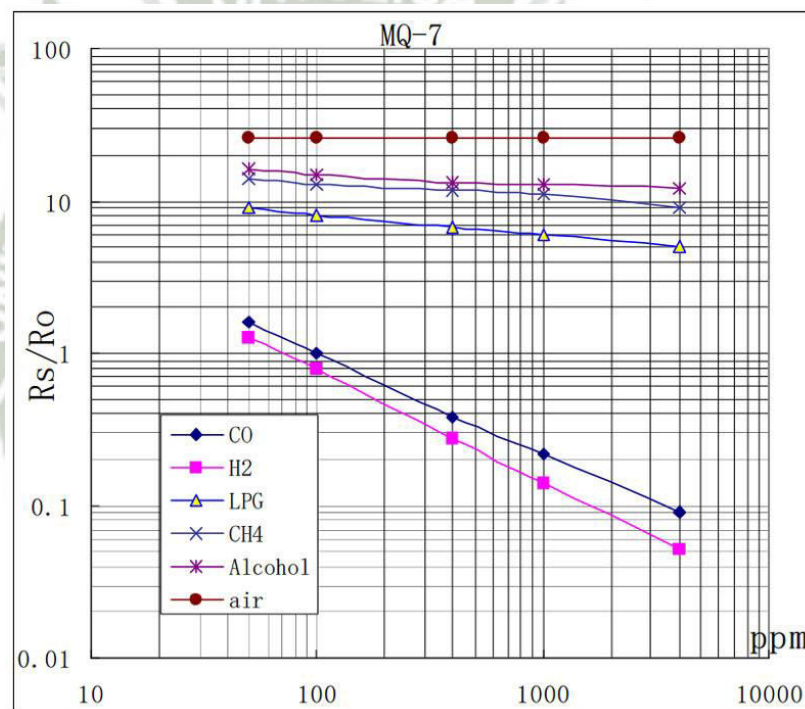


Figura3.40: Curva característica de sensor mq7

Fuente: http://www.electronicoscaldas.com/datasheet/MQ-7_Hanwei-Electronics.pdf

La resistencia de la superficie del sensor Rs se obtiene a través de la salida de señal de voltaje efectuada de la resistencia a la carga RL que se coloca en serie. La relación entre ellos se describe:

$$Rs \setminus RL = (Vc - VRL) / VRL$$

En la figura3.33 Se muestra la dependencia típica del sensor MQ-7 en la temperatura y la humedad.

Ro: resistencia del sensor a 100ppm CO en aire a 33% HR y 20 grados.

Rs: resistencia del sensor a 100ppm de CO a diferentes temperaturas y humedades.

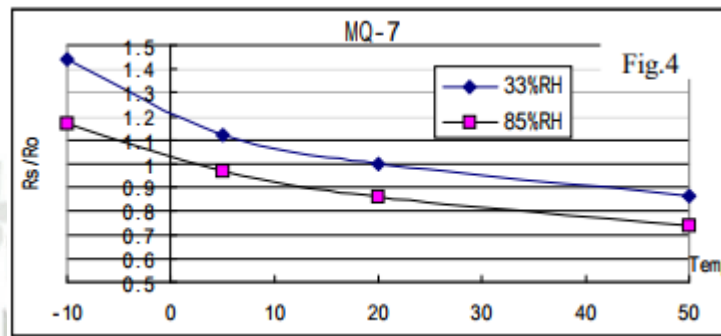


Figura3.41: Dependencia de Co con respecto a temperatura y humedad

Fuente: http://www.electronicoscaldas.com/datasheet/MQ-7_Hanwei-Electronics.pdf

3.33.-LIMITES PERMISIBLES DE CO EN EL AIRE

PPM CO in Air	Percent CO in Air	Workplace CO Exposure Limits, Air Quality and CO Alarm Standards
9 ppm	0.0009%	EPA National Ambient Air Quality Standard (NAAQS) over 8-hour TWA
25 ppm	0.0025%	ACGIH Threshold Limit Value (TLV) and Cal/OSHA PEL over 8-hour TWA
35 ppm	0.0035%	NIOSH Recommended Exposure Limit (REL) over 8-hour TWA
50 ppm	0.0050%	OSHA Permissible Exposure Limit (PEL) over 8-hour TWA
70 ppm	0.0070%	UL 2034 Residential CO Alarm Activation within 60-240 minutes
150 ppm	0.0150%	UL 2034 Residential CO Alarm Activation within 10-50 minutes
200 ppm	0.0200%	Cal/OSHA and NIOSH Ceiling Limit (C)
400 ppm	0.0400%	UL 2034 Residential CO Alarm Activation within 4-15 minutes
1200 ppm	0.1200%	NIOSH Immediately Dangerous to Life or Health value (IDLH)

Tabla3.10: Límites permisibles de CO

Fuente: <http://www.fireengineering.com/articles/2014/11/gas-and-carbon-monoxide-awareness.html>

3.34.- RUTINA DE MEDICION DE CO

```
void loop() {
  int16_t adc0[5];
  int co_x = 0; //int para valor
sin procesar de co
  int co_y = 0; //int para valor
compensado co
  int coppm = 0; //int para ppm
calculado
  int prom = 0; //int para valor
promedio
  for (int x = 0 ; x<5 ; x++)
  {
    adc0[x] = ads.readADC_SingleEnded(0);
    delay(200);
  }
  for (int x = 0 ; x<5 ; x++){ //sumamos las
muestras obtenidas con el promedio y alacena en promedio
    prom = prom + adc0[x];

  }
  co_x = prom/5; //dividimos las
muestras entre 10
  co_y = co_x - coZero; //obtenemos un valor
ponderado
  coppm = map(co_y,0,65536,0,500); //hace un mapeo de valores
de co en la atmosfera
  Serial.print("Nivel de CO: "); //imprime etiqueta
  Serial.print(coppm); //imprimimos el
valor de co
  Serial.println(" ppm"); //imprimimos
unidades
}
```

3.35.- VISUALIZACION DE DATOS POR MONITOR SERIE

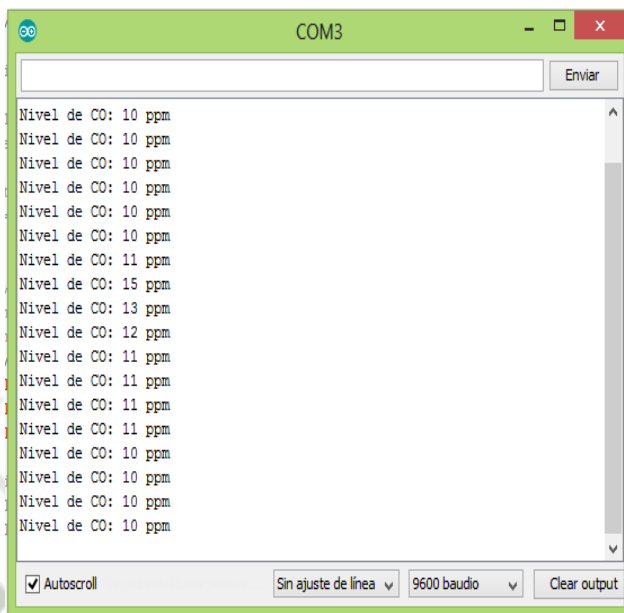
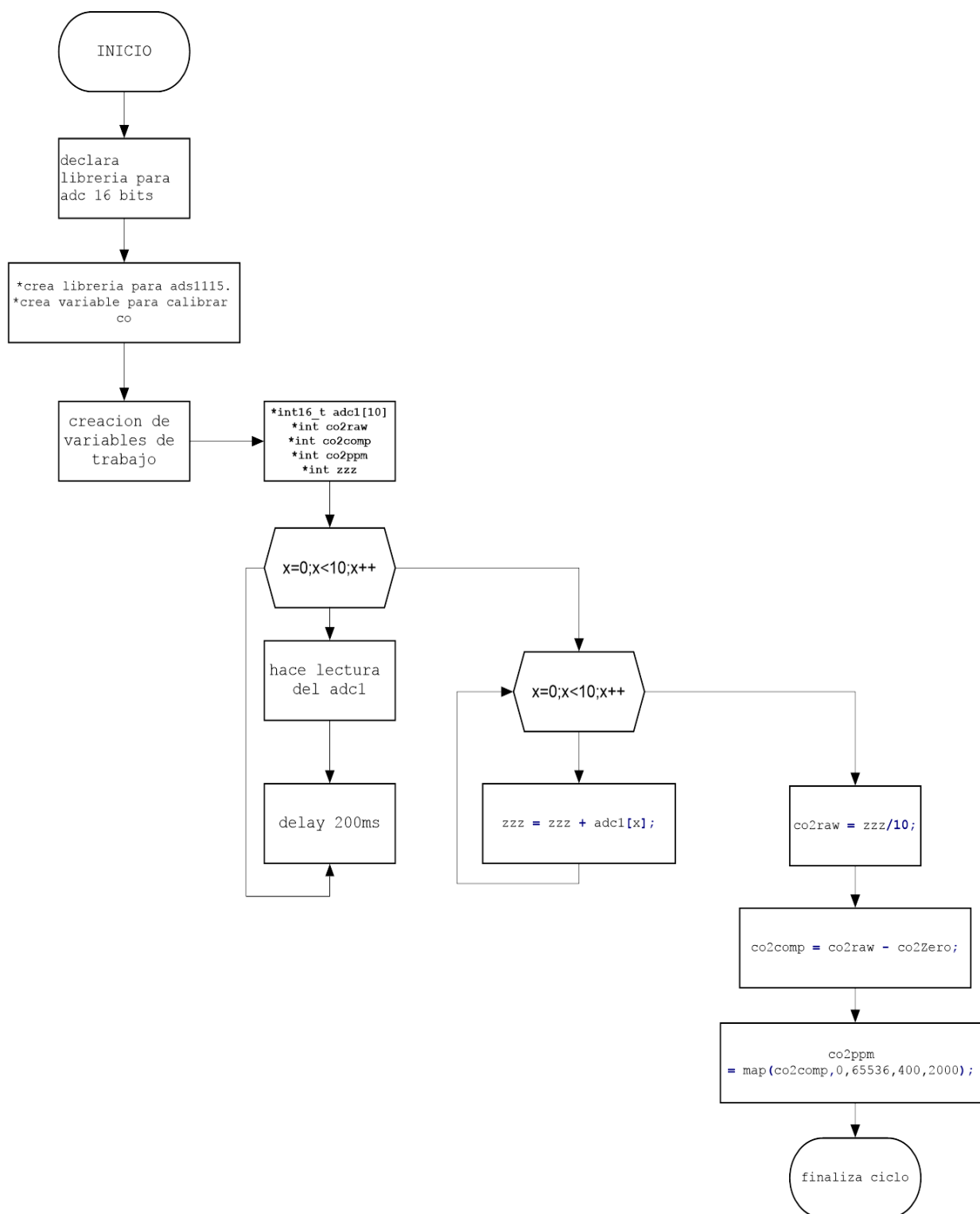


Figura3.42: Niveles de CO mostrados en monitor serie

Fuente: Propia

3.36.-DIAGRAMA DE FLUJO MEDICION DE DIOXIDO DE CARBONO



Flujograma3.7: Medición de co

3.37.- CURVA CARACTERISTICA DE SENSOR MQ135

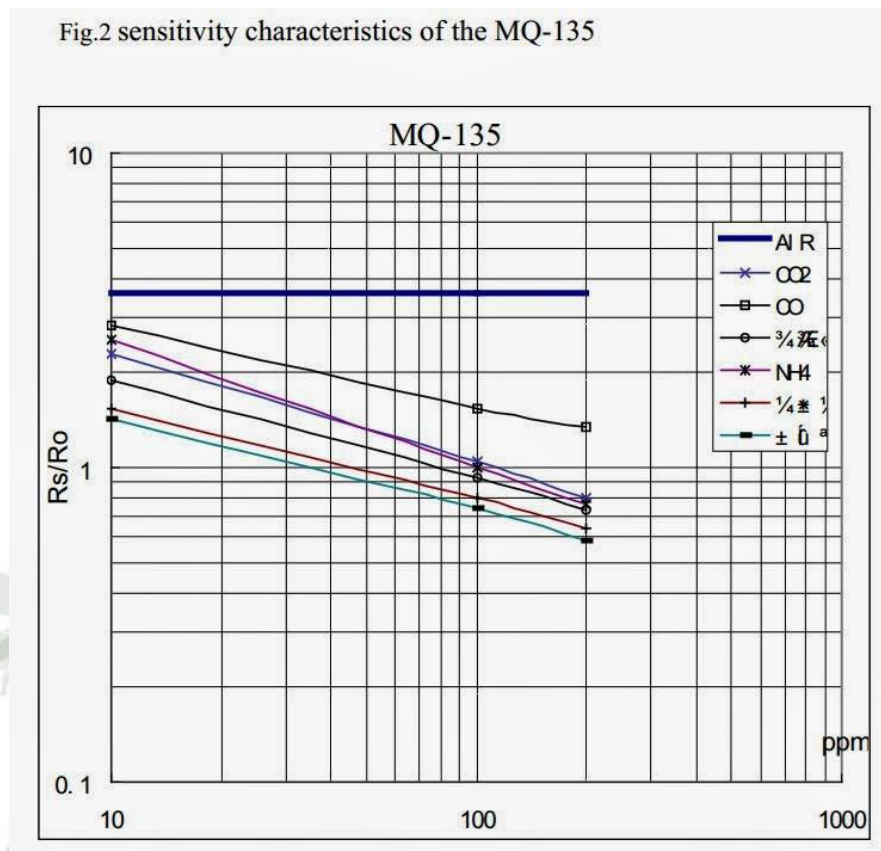


Figura3.43: Curva característica de sensor mq125

Fuente: <https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/resources/SNS-MQ135.pdf>

3.38.- RUTINA DE MEDICION DE CO2

```
void loop()
{
    int16_t adc1 [10];           //int array para las lecturas de co2
    int co2raw = 0;             //int para valores sin procesar de co2
    int co2comp = 0;           //int para valor compensado co2
    int co2ppm = 0;            //int para ppm calculado
    int zzz = 0;               //int for promedio

    for (int x = 0 ; x<10 ; x++)
    {
        adc1[x] = ads.readADC_SingleEnded(1);
        delay(200);
    }
    for (int x = 0 ; x<10 ; x++){ //sumamos las muestras obtenidas
        con el promedio y alacena en promedio
        zzz=zzz + adc1[x];
    }
    co2raw = zzz/10;           //dividimos las muestras entre 10
    co2comp = co2raw - co2Zero; //obtenemos un valor ponderado
    co2ppm = map(co2comp,0,65536,400,2000); //hace un mapeo de
    valores de co2 en la atmosfera
}
```

3.39.-DATOS OBTENIDOS DE LA RUTINA DE MEDICION

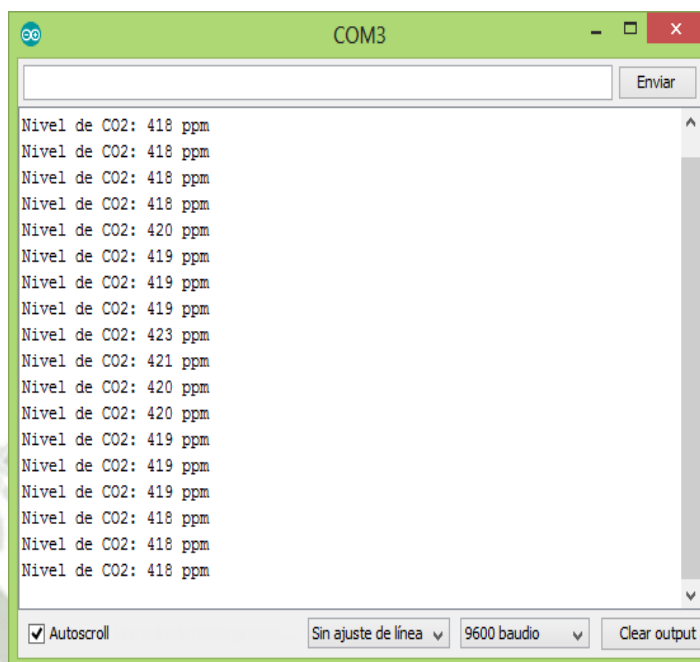
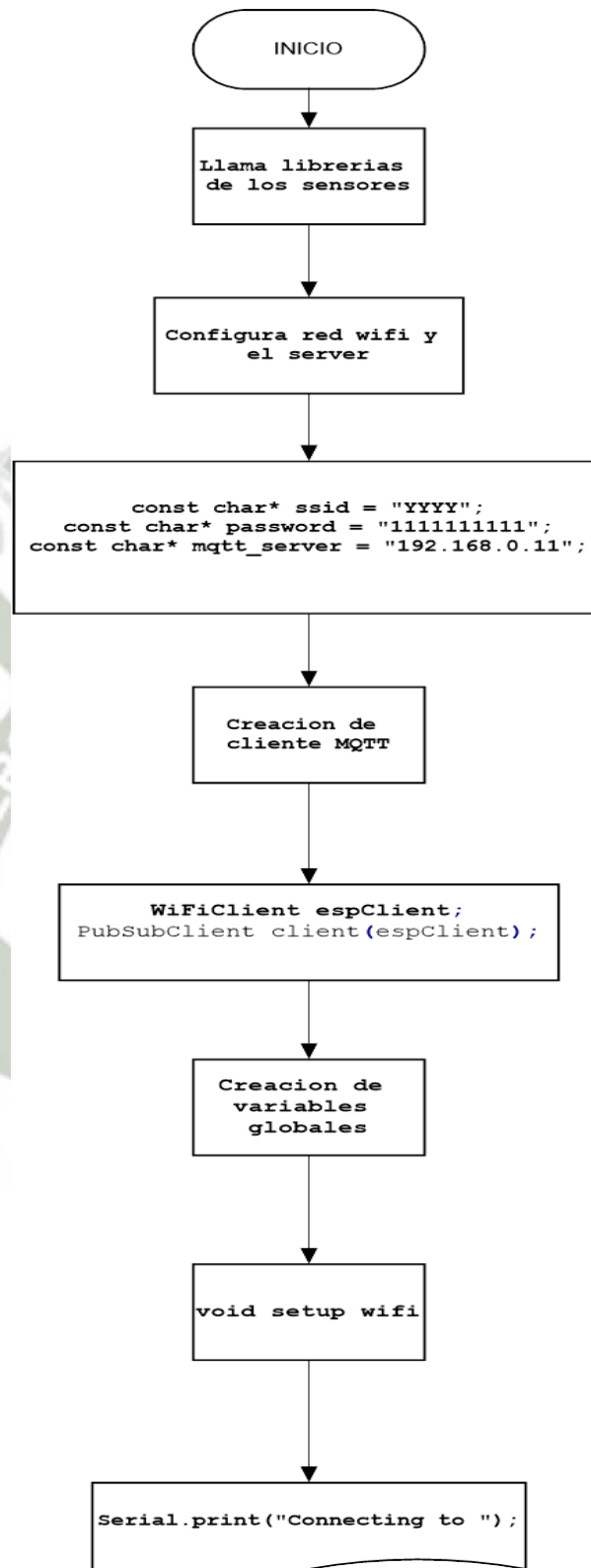


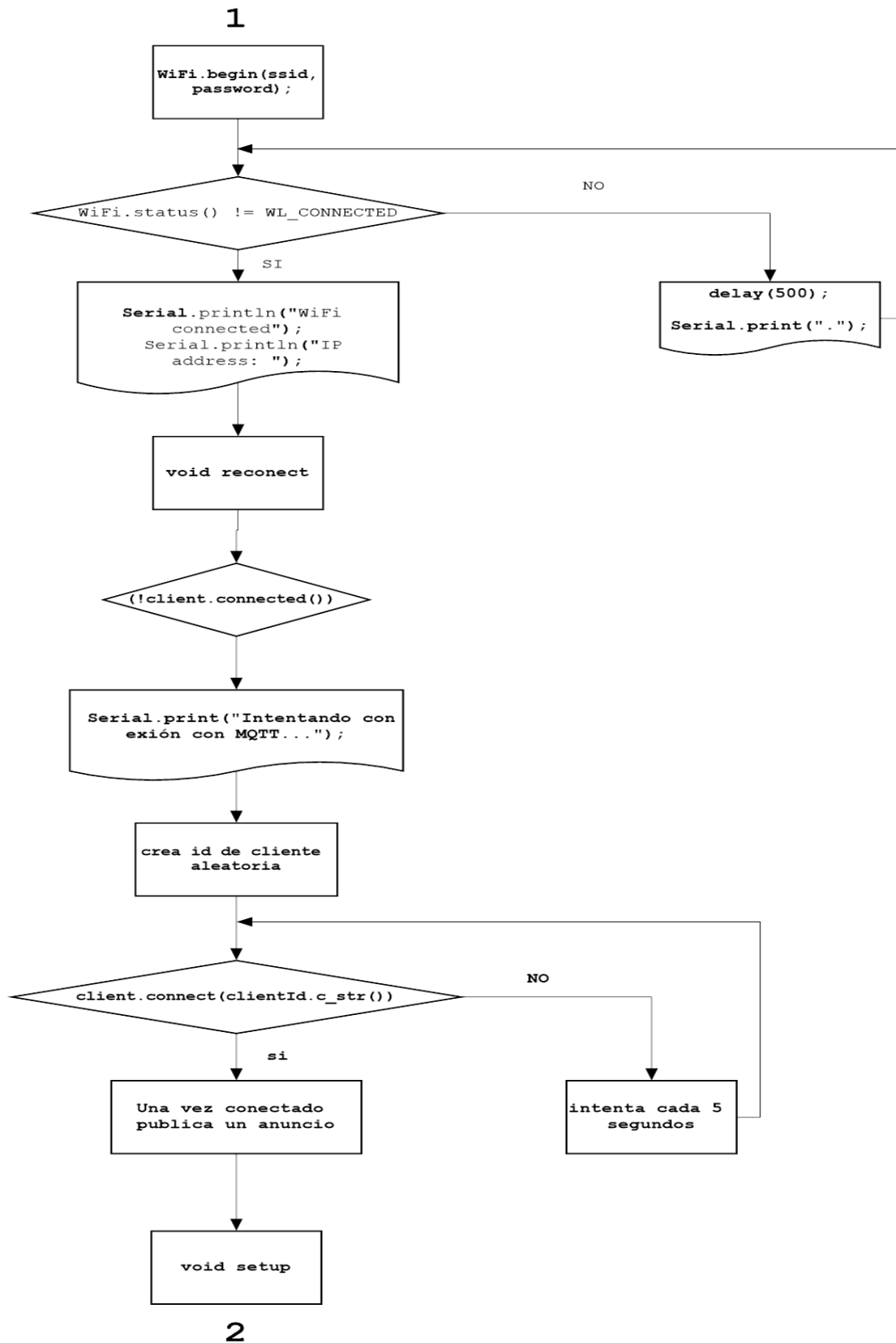
Figura3.44: datos obtenidos por sensor de co2

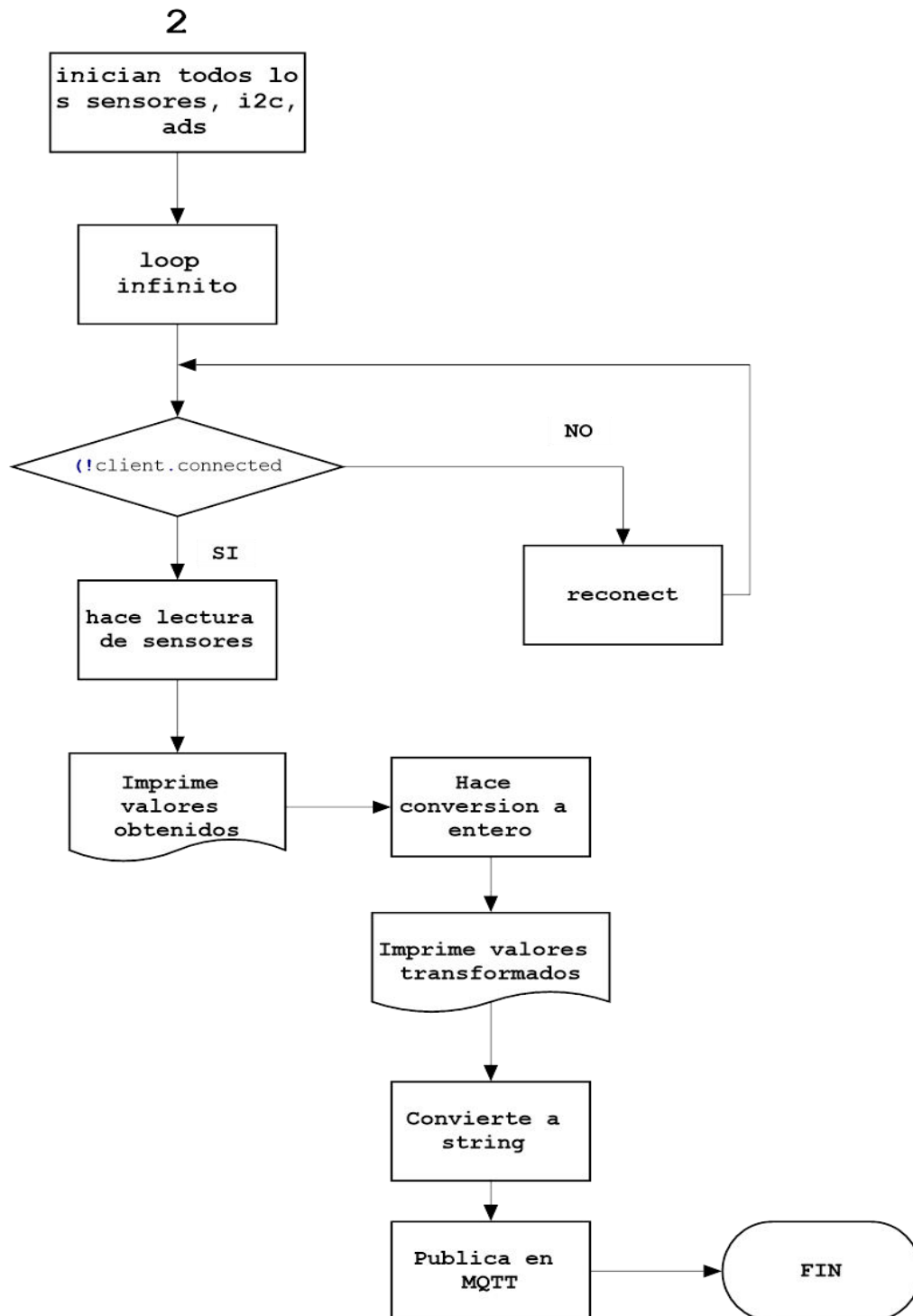
Fuente: Propia

3.40.-DIAGRAMA DE FLUJOS PROGRAMA GENERAL



1





Flujograma3.8: diagrama general

3.41.- DESARROLLO DE DASHBOARD PARA VISUALIZACION EN TIEMPO REAL EN NODE RED

3.41.1.- CONFIGURACION PARA LA COMUNICACIÓN ENTRE NODE RED Y NODEMCU

*Configuramos nuestro servidor, en este caso es el ip de nuestra máquina de trabajo

```

Adaptador de LAN inalámbrica Wi-Fi:
Sufijo DNS específico para la conexión. . . :
Uínculo: dirección IPv6 local. . . . . : fe80::9d93:31a7:5364:eeae%13
Dirección IPv4. . . . . : 192.168.0.11
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.0.1
    
```

Figura3.45: Verificación de server (ip)

Fuente: Propia

Una vez hecho eso, copiamos nuestro bróker el cual es 1883. En la siguiente imagen se muestra la configuración de nuestro server.

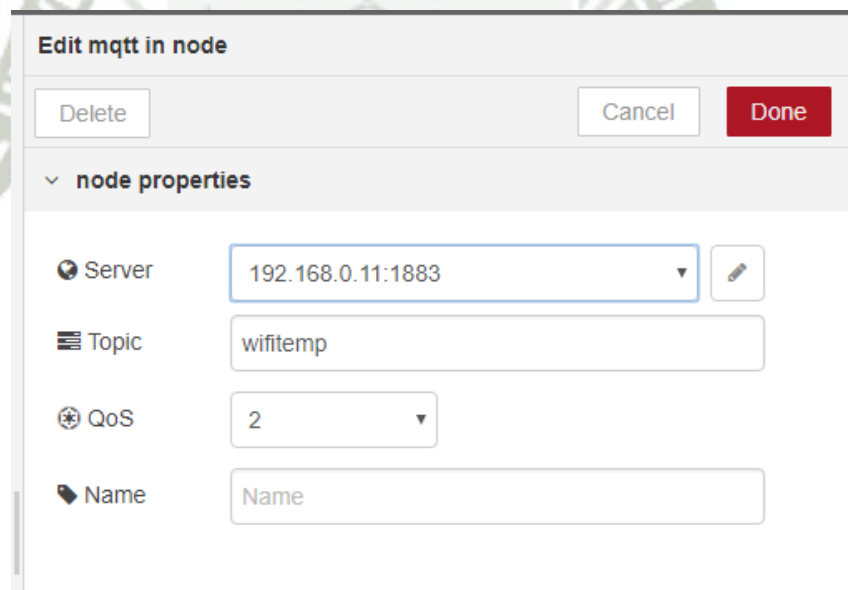
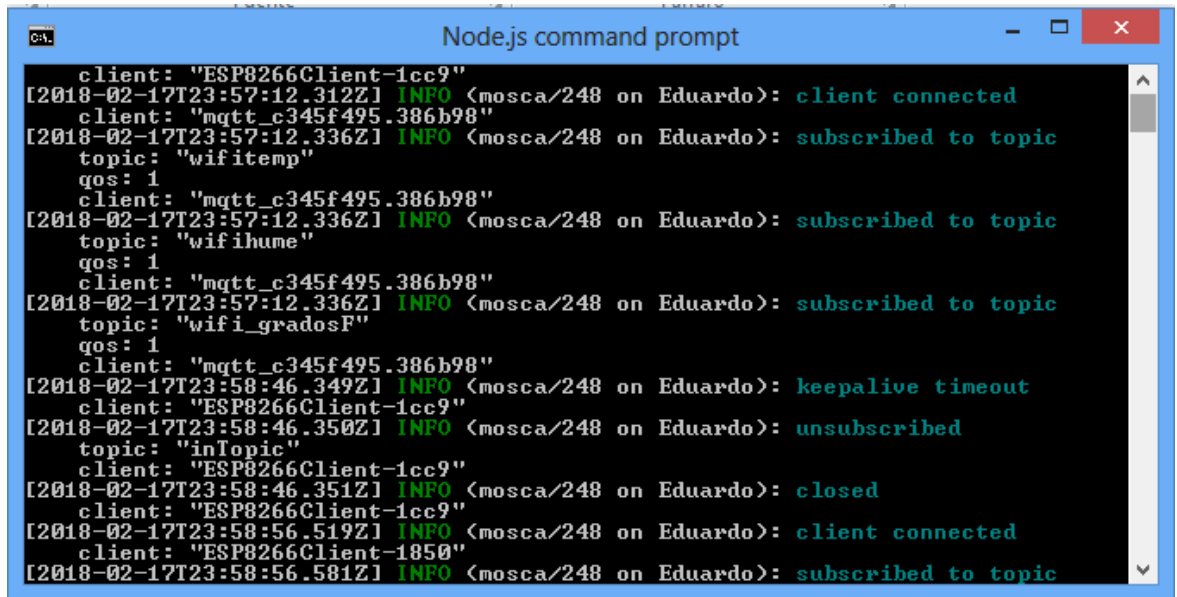


Figura3.46: Asignacion de server y mqtt

Fuente: Propia

Esto se hizo para la temperatura, de la misma manera se hará para el resto de sensores que estarán conectados.

En la figura se muestra por el puerto serial que la conexión con nuestro bróker Mqtt ha sido exitosa una vez realizada esa conexión podremos obtener los datos del sensor DHT22 y BH1750 también mostramos los datos que enviaremos a node-red, para esto hicimos una conversión de datos de float a string, este proceso se realizara para el resto de sensores ya que solo podemos enviar cadenas de caracteres.

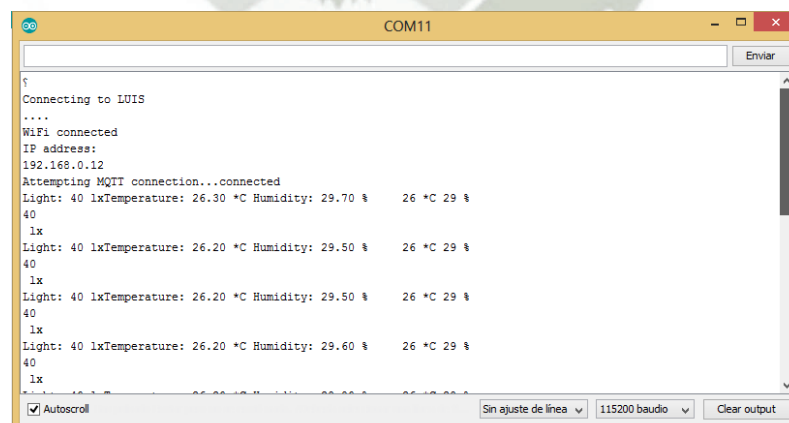


```

Node.js command prompt
client: "ESP8266Client-1cc9"
[2018-02-17T23:57:12.312Z] INFO (mosca/248 on Eduardo): client connected
client: "mqtt_c345f495.386b98"
[2018-02-17T23:57:12.336Z] INFO (mosca/248 on Eduardo): subscribed to topic
topic: "wifitemp"
qos: 1
client: "mqtt_c345f495.386b98"
[2018-02-17T23:57:12.336Z] INFO (mosca/248 on Eduardo): subscribed to topic
topic: "wifihume"
qos: 1
client: "mqtt_c345f495.386b98"
[2018-02-17T23:57:12.336Z] INFO (mosca/248 on Eduardo): subscribed to topic
topic: "wifi_gradosF"
qos: 1
client: "mqtt_c345f495.386b98"
[2018-02-17T23:58:46.349Z] INFO (mosca/248 on Eduardo): keepalive timeout
client: "ESP8266Client-1cc9"
[2018-02-17T23:58:46.350Z] INFO (mosca/248 on Eduardo): unsubscribed
topic: "inTopic"
client: "ESP8266Client-1cc9"
[2018-02-17T23:58:46.351Z] INFO (mosca/248 on Eduardo): closed
client: "ESP8266Client-1cc9"
[2018-02-17T23:58:56.519Z] INFO (mosca/248 on Eduardo): client connected
client: "ESP8266Client-1850"
[2018-02-17T23:58:56.581Z] INFO (mosca/248 on Eduardo): subscribed to topic
    
```

Figura3.47: Comunicación exitosa con MQTT

Fuente: Propia



```

COM11
Connecting to LUIS
....
WiFi connected
IP address:
192.168.0.12
Attempting MQTT connection...connected
Light: 40 lxTemperature: 26.30 *C Humidity: 29.70 % 26 *C 29 %
40
lx
Light: 40 lxTemperature: 26.20 *C Humidity: 29.50 % 26 *C 29 %
40
lx
Light: 40 lxTemperature: 26.20 *C Humidity: 29.50 % 26 *C 29 %
40
lx
Light: 40 lxTemperature: 26.20 *C Humidity: 29.60 % 26 *C 29 %
40
lx
    
```

Figura3.48: Conexión con mqtt y muestra de datos por monitor serie

Fuente: Propia

En la figura se muestra el nodo mqtt, el cual como se menciona es el encargado de emitir los mensajes, este se conectó con un intermediario (NODEMCU). También está conectado con un depurador el cual estará encargado de mostrar los datos enviados a la salida. Los topic que se reciben son wifitemp, wifihume y wifiluminosidad.

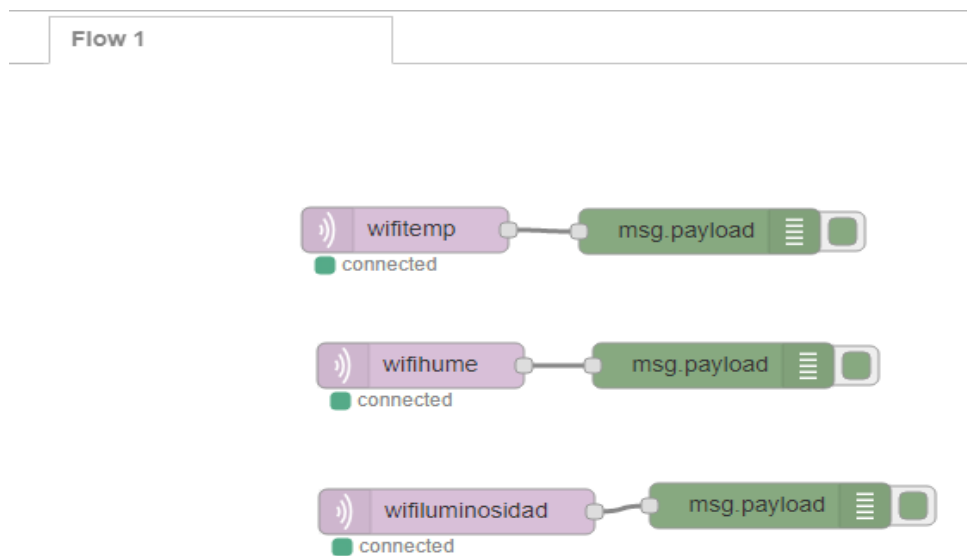


Figura3.49: Debug de temperatura,humedad,luminosidad

Fuente:Propia

Una vez realizada toda esta tarea se procede a correr el programa y en la figura se muestran los mensajes emitidos a través de mqtt y nodemcu. Se puede observar también la conversión de los datos que hicimos en el programa y como se vuelve a recalcar solo se recibe cadena de caracteres (String).

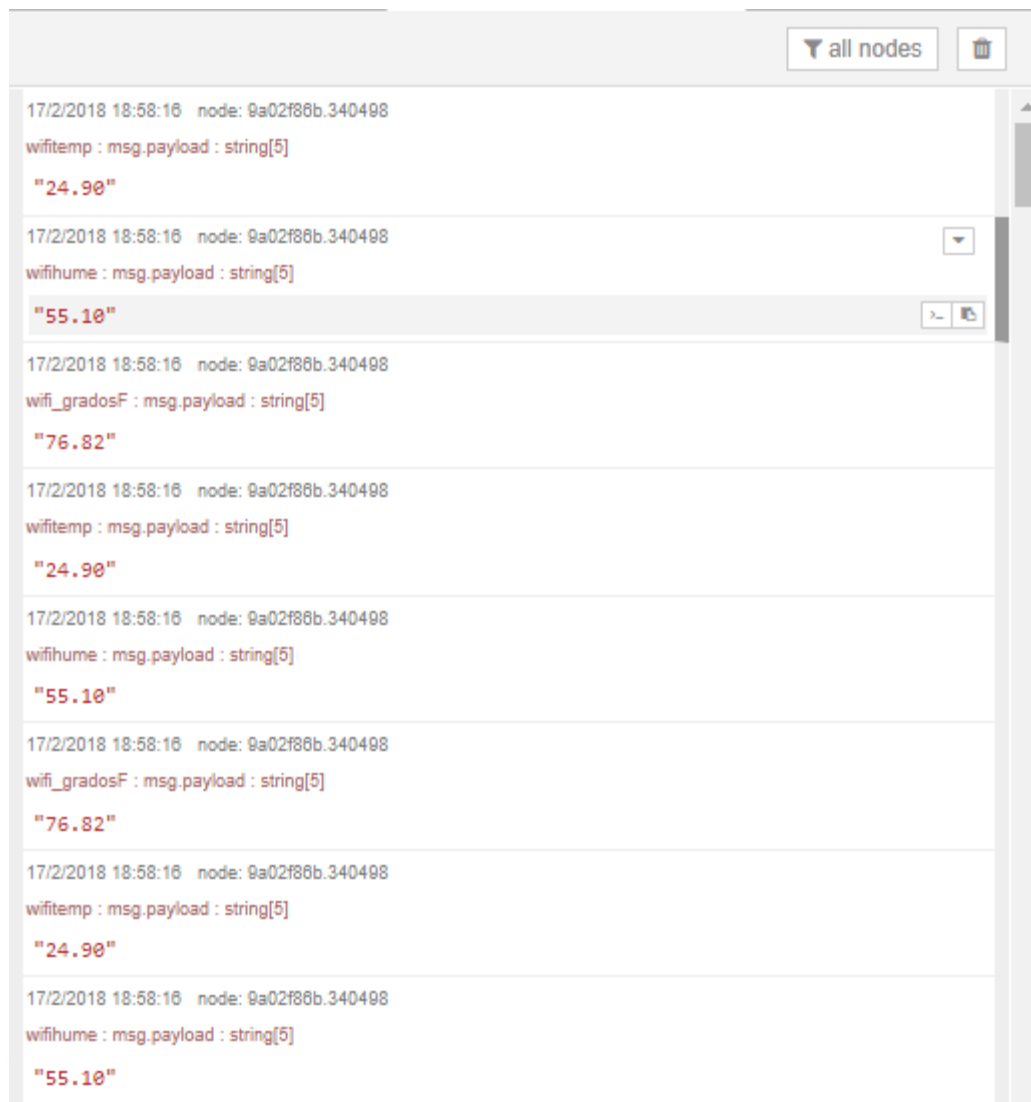


Figura3.50: Muestra de datos recibidos en node red

Fuente: Propia

En la figura que se encuentra a continuación se presentan hasta el momento algunos parámetros como: la temperatura, humedad, presión barométrica, altitud, la temperatura censada por el bmp180.

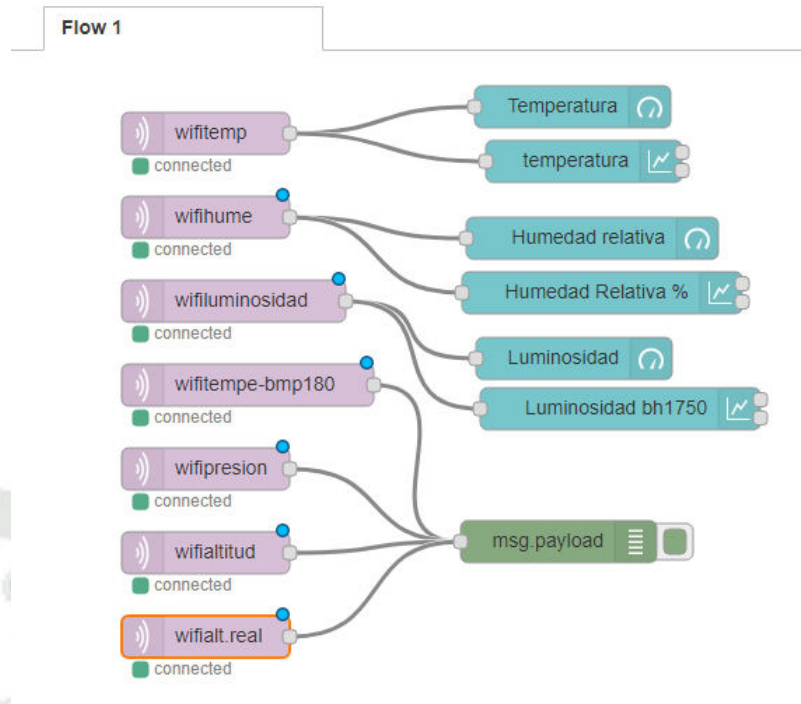


Figura3.51: elaboración de dashboard con nodos

Fuente: Propia

Se realizó el mismo procedimiento ya descrito anteriormente, como se ve muestran la configuración de los guages y graficas chart y el depurador que se encarga de mostrar los datos del sensor bmp180.

```

wifipresion : msg.payload : string[8]
"76672.00"

11/10/2017 13:09:26 node: eb2132e5.49581
wifialtitud : msg.payload : string[7]
"2290.73"

11/10/2017 13:09:26 node: eb2132e5.49581
wifialt.real : msg.payload : string[7]
"-177.81"

11/10/2017 13:09:29 node: eb2132e5.49581
wifitempe-bmp180 : msg.payload : string[5]
"27.50"

11/10/2017 13:09:29 node: eb2132e5.49581
wifipresion : msg.payload : string[8]
"76676.00"

11/10/2017 13:09:30 node: eb2132e5.49581
wifialtitud : msg.payload : string[7]
"2290.42"

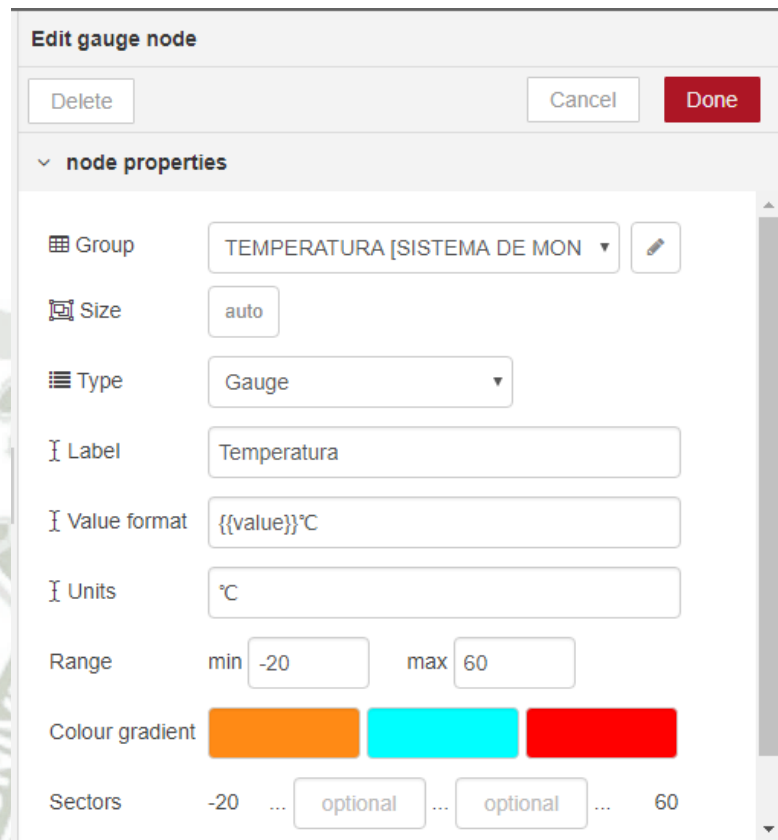
11/10/2017 13:09:30 node: eb2132e5.49581
wifialt.real : msg.payload : string[7]

```

Figura3.52: Depuracion de datos enviados por nodeMcu

Fuente:Propia

A continuación se muestran la configuración de parámetros para poder realizar la visualización en el dashboard de node-red, de acuerdo a las características descritas de cada sensor.



Edit gauge node

Delete Cancel Done

node properties

Group TEMPERATURA [SISTEMA DE MON

Size auto

Type Gauge

Label Temperatura

Value format {{value}}°C

Units °C

Range min -20 max 60

Colour gradient

Sectors -20 ... optional ... optional ... 60

Figura3.53: Configuración de parámetro para gauge temperatura

Fuente: Propia

- En la etiqueta group se designa el nombre de nuestro gauge
- Por lo general en la etiqueta size es preferible dejarlo en automático
- Seleccionamos nuestro tipo de gráfica, en este caso Gauge
- En label asignamos el nombre de nuestro grafico
- Asignamos el valor de formato y las unidades
- Asignamos el rango de medición

Una vez realizados todo lo anterior queda configurado nuestro diagrama de gauge que encargara de mostrar el valor medido del sensor.

Para la obtención de la gráfica de datos en tiempo real, haremos uso del nodo chart, a continuación se muestra su respectiva configuración.

En la siguiente figura se muestra como se configuraron los parámetros para la gráfica de temperatura, de la misma forma se hará para el resto de sensores.

- Damos un nombre a la etiqueta
- Seleccionamos el tipo de gráfica, en nuestro caso será line chart
- Con respecto al eje X se trabajara en base al tiempo de medición, esto quiere decir que la medición de estos parámetros dependerá del tiempo.
- Elegimos el tipo de formato en este caso será horas/minutos/segundos.
- En el eje Y se colocara el rango de medición de cada sensor.

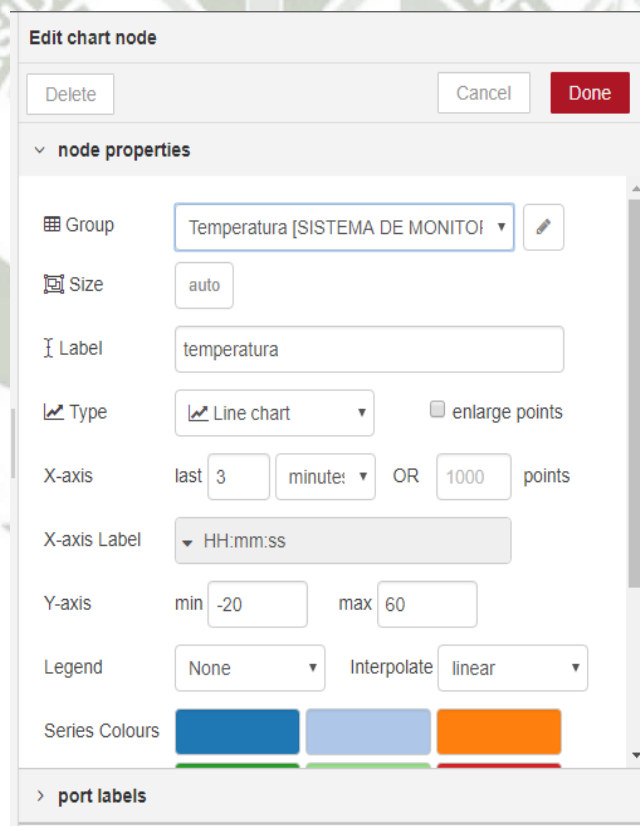


Figura3.54: Configuración de parámetros grafica chart

Fuente: Propia

En la figura se muestra ya implementados nuestros nodos tanto de gauge y chart para la visualización de nuestros datos en tiempo real.

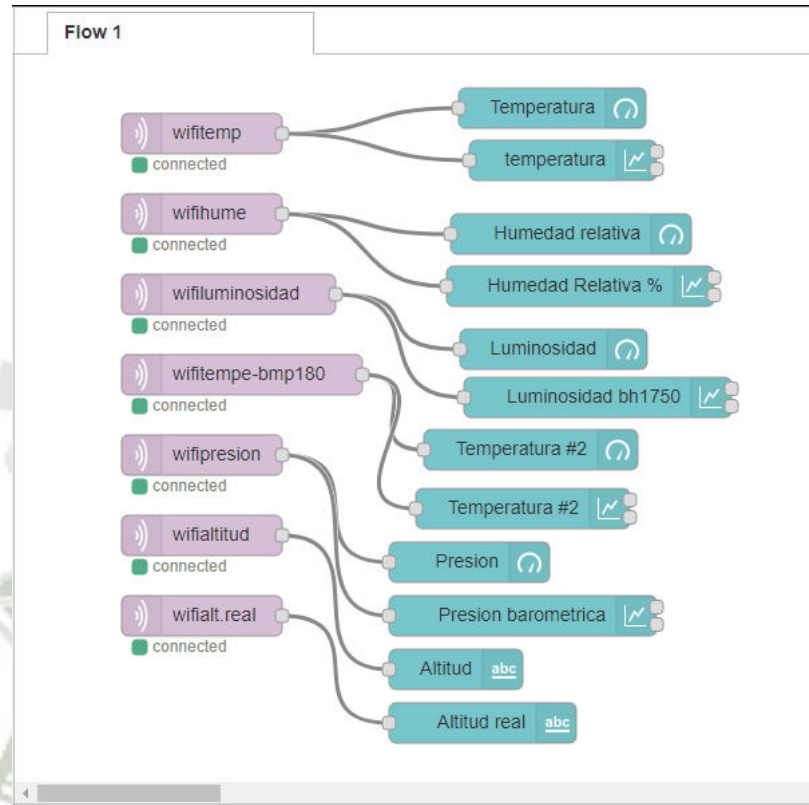


Figura3.55: nodos mqtt

Fuente: Propia

A continuación en las figuras se muestra nuestra dashboard con los sensores de temperatura, humedad, luminosidad y presión, dB, radiación UV, presión, monóxido de carbono, dióxido de carbono, altitud.

Se puede visualizar los diagramas de gauge y las gráficas chart que muestra en la medición de los parámetros ya mencionados.

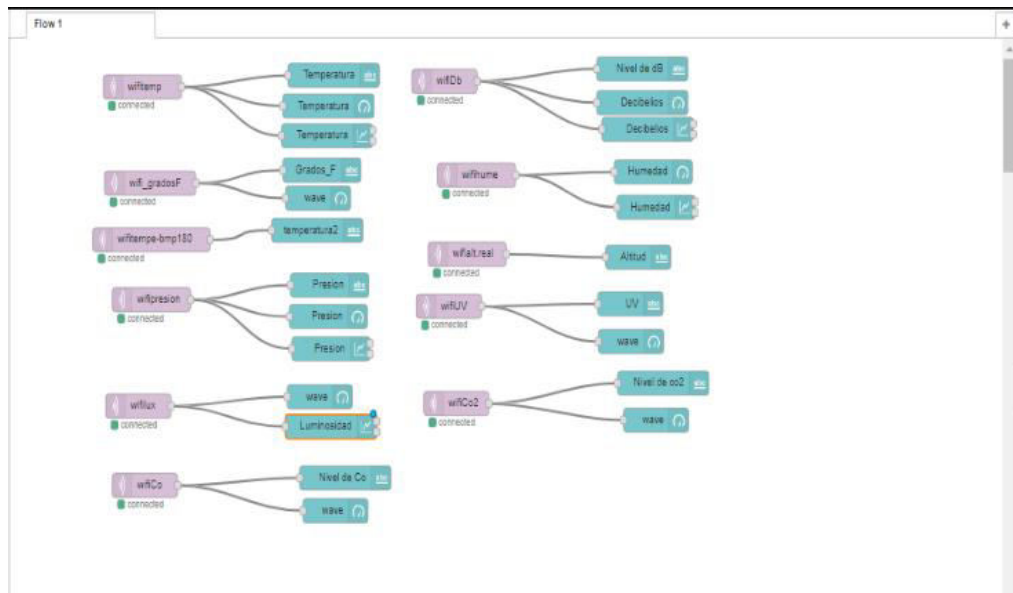


Figura3.56: parámetros para ser graficados

Fuente: Propia

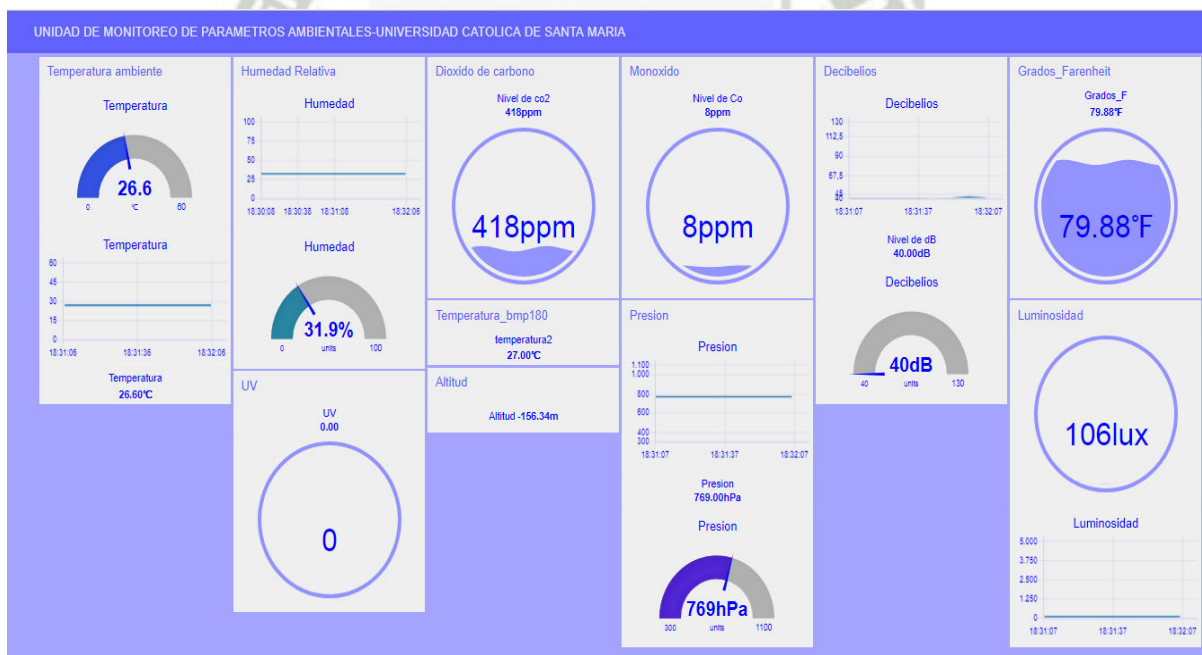


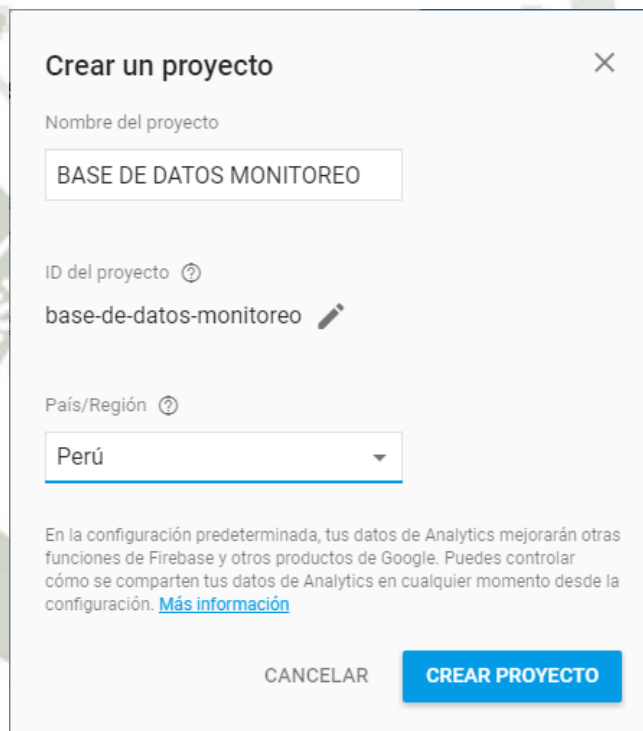
Figura3.57: Dashboard elaborado

Fuente: Propia

3.41.2.-INTEGRACION DE BASE DE DATOS EN NODE RED

Firebase Real time Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando se compilan apps multiplataforma con SDK de iOS, Android y JavaScript, todos tus clientes comparten una instancia de Realtime Database y reciben actualizaciones de forma automática con los datos más recientes.

3.41.2.1. CREACION DE BASE DE DATOS



Crear un proyecto

Nombre del proyecto

BASE DE DATOS MONITOREO

ID del proyecto ⓘ

base-de-datos-monitoreo ✎

País/Región ⓘ

Perú ▼

En la configuración predeterminada, tus datos de Analytics mejorarán otras funciones de Firebase y otros productos de Google. Puedes controlar cómo se comparten tus datos de Analytics en cualquier momento desde la configuración. [Más información](#)

CANCELAR CREAR PROYECTO

Figura3.58: Creación de proyecto firebase

Fuente: Propia

Una vez creado nuestro proyecto que lleva por nombre (BASE DE DATOS MONITOREO) como se ve en la figura (3.50). El siguiente paso que se nos pide realizar es optar por una forma de autenticación como se ve en la figura. Como forma de autenticación se optó por medio de correo electrónico.

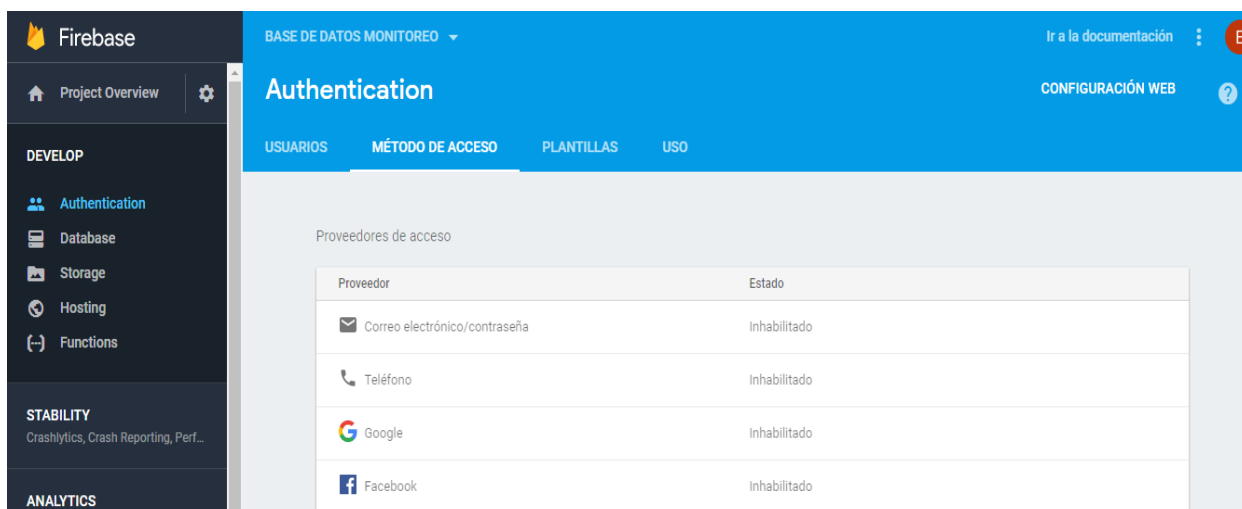


Figura3.59: Autenticación de cuenta

Fuente: Propia

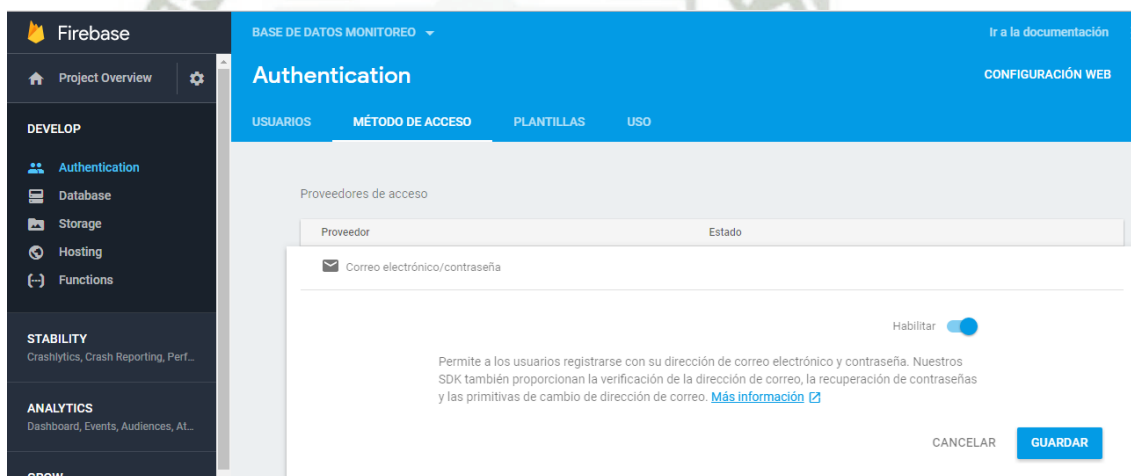


Figura3.60: Confirmación de autenticacion

Fuente: Propia

Una vez creado nuestro proyecto y haber confirmado nuestra cuenta de correo, se muestra en la figura3.53, el panel principal de database.

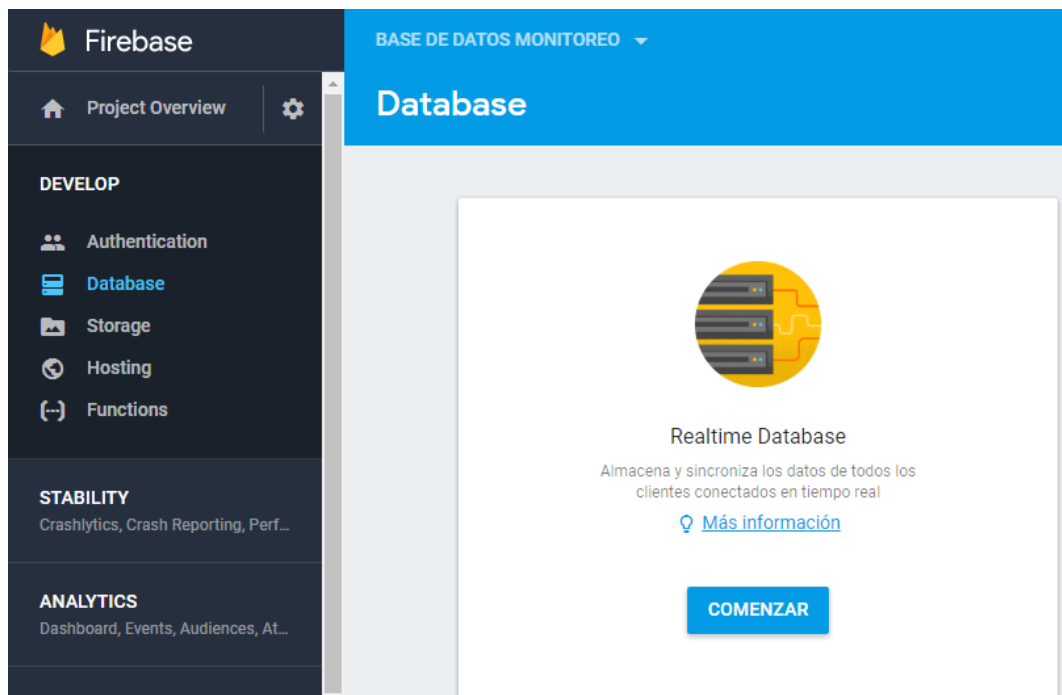


Figura3.61: database firebase

Fuente: Propia

Dado que en firebase existen modalidades de uso gratuito y de pago, es necesario que cambiemos unas líneas de código que nos permitirán como un usuario libre.

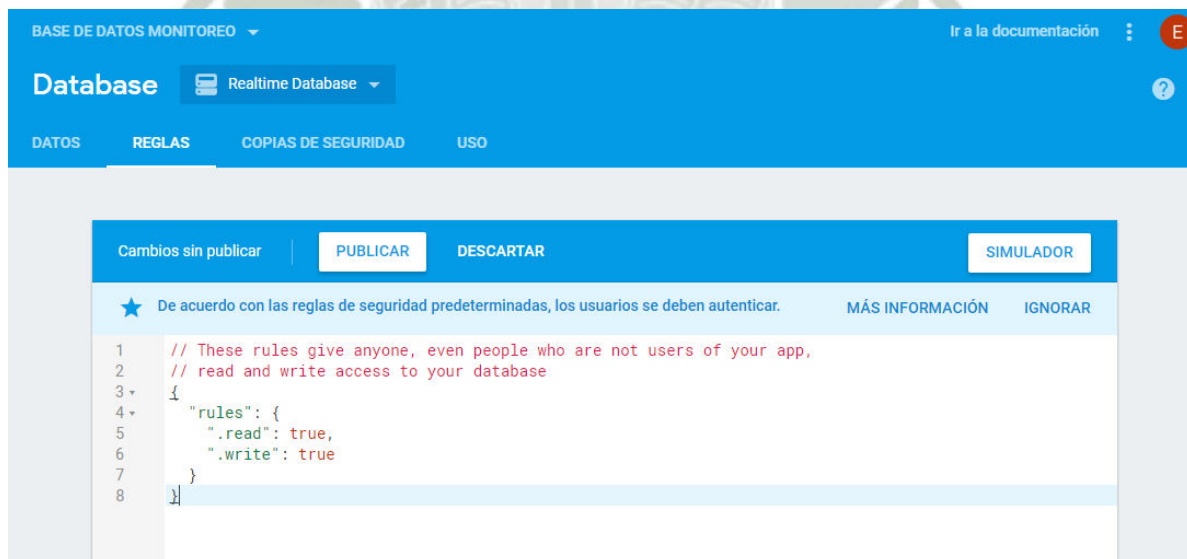


Figura3.62: Código de autenticación

Fuente:Propia

3.41.2.2.-RUTINA PARA HACER PÚBLICA BASE DE DATOS

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

En la figura se muestran todas las variables que se insertaron en la base de datos

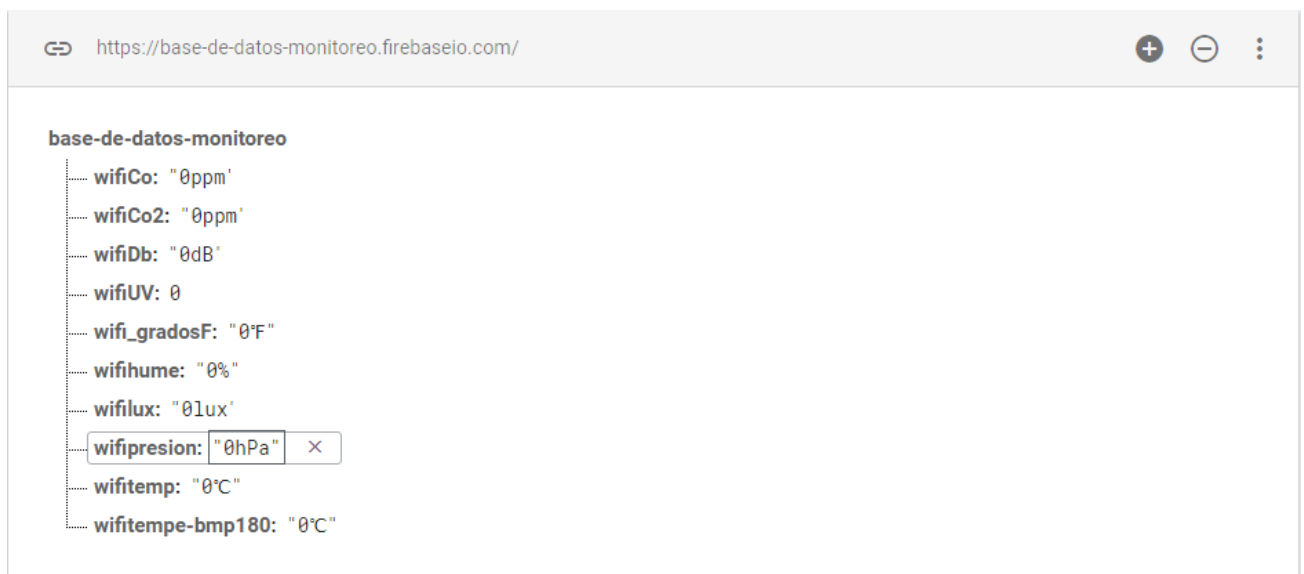


Figura3.63: Creación de variables a almacenar

Fuente: Propia

A continuación en node-red usaremos los nodos de firebase que se instalaron y los agregaremos en el orden que se muestra a continuación:

MQTT>FIREBASE>GRAPHICS.

Quiere decir que al momento de recibir los valores obtenidos por los sensores por la tarjeta NodeMCU, estos serán almacenados directamente en firebase, luego de esa operación serán graficados tal como se muestra en el flujo de nodos mostrados en la figura.

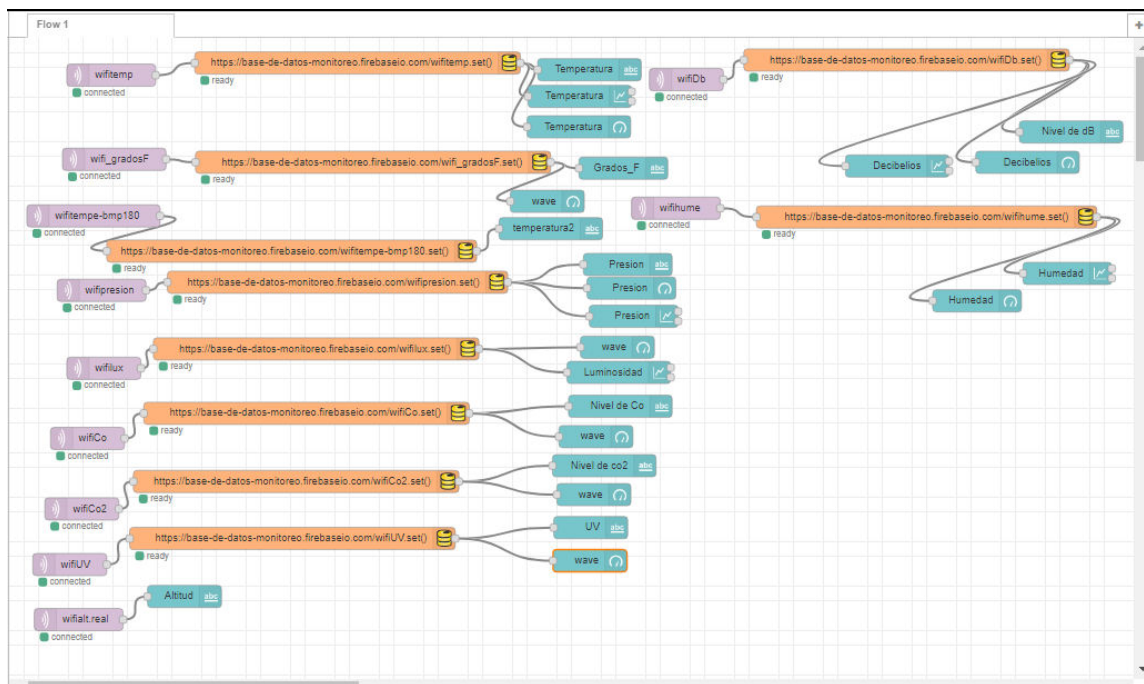


Figura3.64: Nodos firebase integrados

Fuente: Propia

Cuando se resalta de color naranja determinada variable, significa que hay una variación en el valor obtenido por el sensor, como se muestra en la figura hay una modificación del valor en las variables wifigradosF, wifihume, wifitemp, wifitempe-bmp180.

Firestore nos informara cuando ocurra este evento automáticamente.

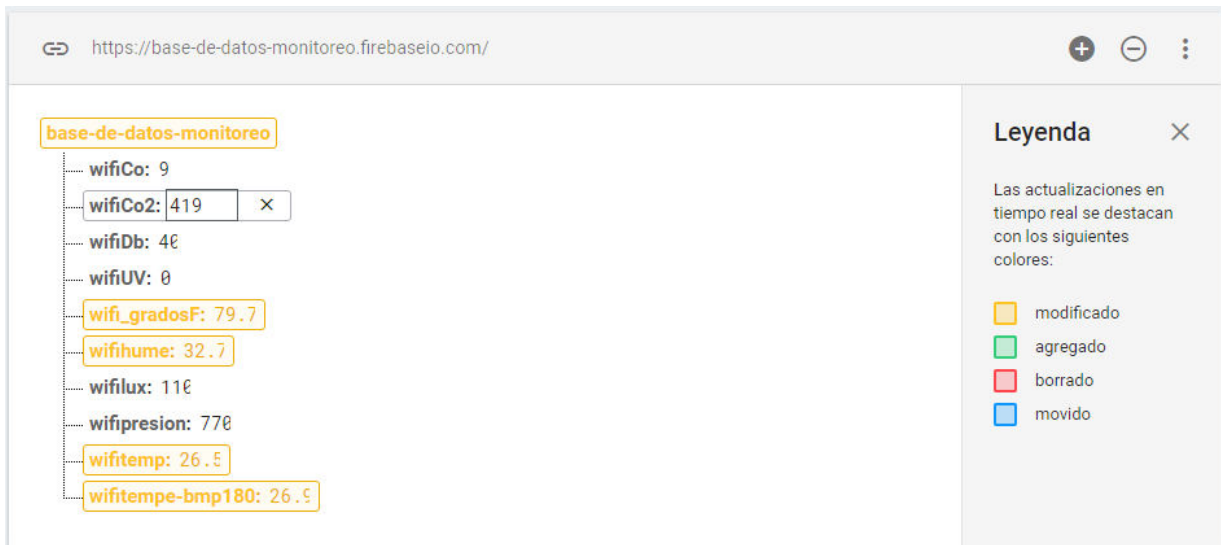


Figura3.65: Almacenamiento en tiempo real

Fuente: Propia

3.41.3.-ELABORACION DE PLACA PARA EL SISTEMA DE MONITOREO

Para poder trabajar con la placa de desarrollo NodeMCU de forma más adecuada y que se acomode a nuestras necesidades, se optó por una placa de cobre perforada, la cual permitió libertad a la hora de elegir por la ubicación tanto de la tarjeta como del módulo analógico ADS1115.

En la figura se muestra ya soldados y ubicados los rieles hembra y macho donde se colocaran nuestros dispositivos de adquisición de datos.

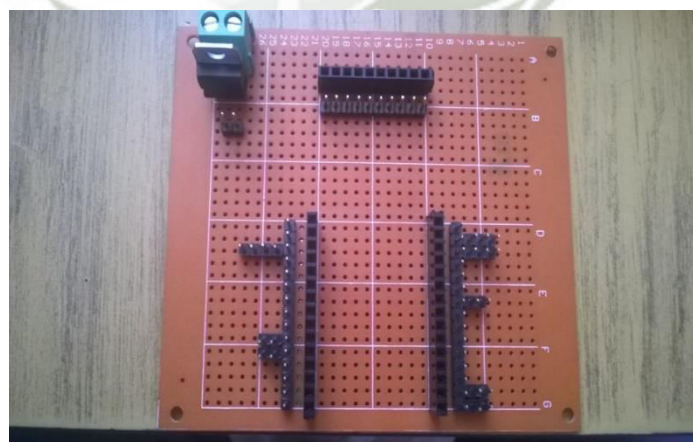


Figura3.66: Elaboración de placa de control

Fuente: Propia

En la figura se muestra ya terminada la placa tanto con la tarjeta NodeMCU y el ADS1115.

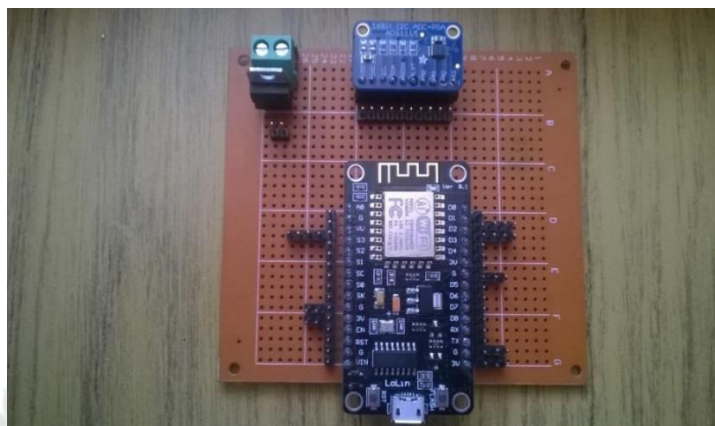


Figura3.67: Placa de control con tarjetas incluidas

Fuente: Propia

En la figura se muestran ubicados los sensores que tomaran los datos del ambiente para luego ser procesados, enviados y almacenados.

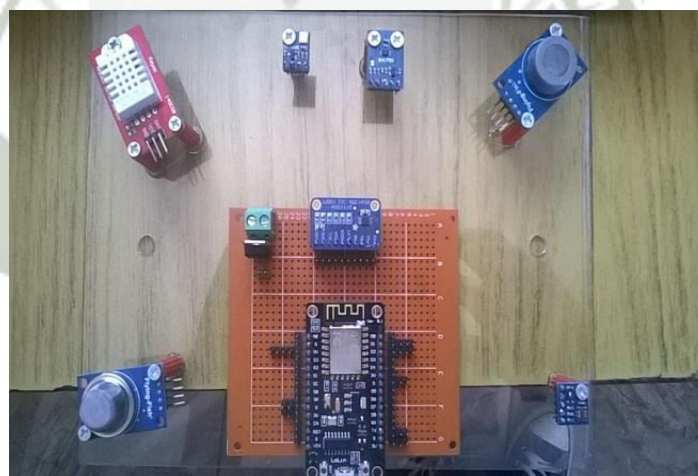


Figura3.68: Sensores y tarjeta NodeMCU

Fuente: propia

En la figura se muestran ya conectados los sensores tanto al ADS1115 como a la tarjeta NodeMCU.

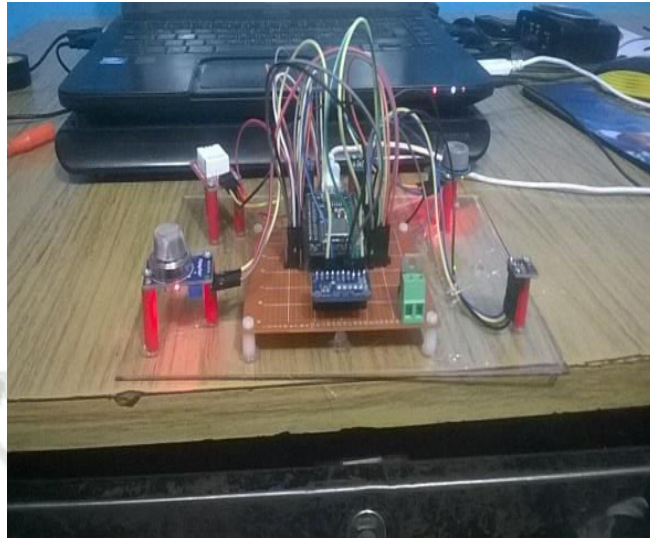


Figura3.69: Sensores conectados NODEMCU

Fuente: propia

CAPITULO IV

MODELAMIENTO Y CONSTRUCCION

4.1 MODELAMIENTO MATEMATICO DEL SISTEMA

4.1.1.-MODELO DEL UAV

En esta sección se desarrolla el modelo basado en leyes físicas que describan la posición y orientación de nuestro cuadricoptero. El modelo dinámico del cuadricoptero se presenta bajo dos formulaciones matemáticas:

Newton-Euler

LaGrange-Euler

El control del movimiento se logra variando la velocidad angular de cada uno de los motores, la razón principal de este análisis es que un cuadricoptero posee tres tipos de movimientos:

- alabeo (roll)
- cabeceo (pitch)
- guiñada (yaw)

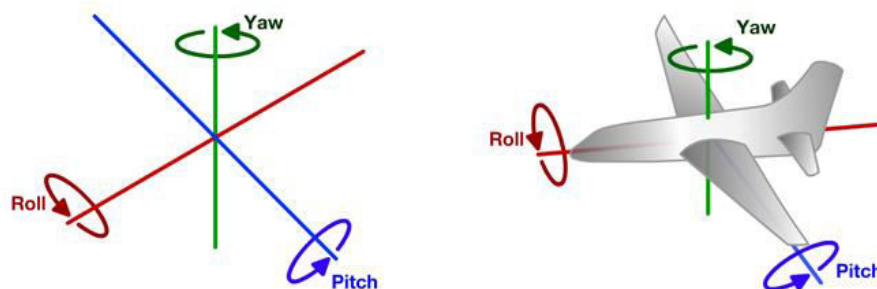


Figura4.1: Yaw, pitch, roll

Consideraremos a nuestro cuadricoptero como un cuerpo sólido rígido de 6 grados de libertad, dividiendo así el sistema en coordenadas traslacionales y rotacionales.

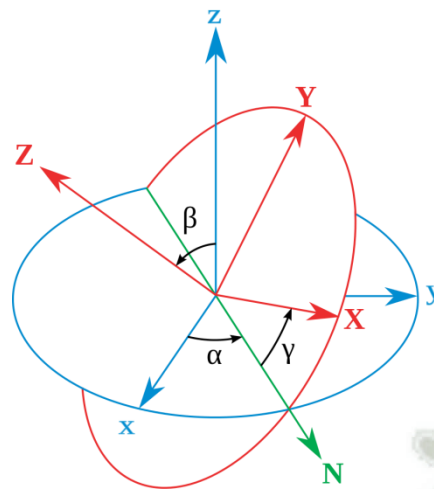


Figura4.2: sistema en coordenadas traslacionales y rotacionales.

Los ángulos de Tait-Bryan son tres ángulos que describirán la rotación de un cuerpo rígido interpretándolo como consecuencia de la rotación sucesiva en torno a los tres ejes X_L, Y_L, Z_L .

Se muestra la rotación de un cuerpo rígido a través de rotaciones sucesivas y se definen los ángulos de rotación que serán de mucha importancia para definir la matriz de rotación R .

La matriz de rotación de cada uno de los ejes esta dada por :

$$R_{z,\varphi} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$R_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (4.2)$$

$$R_{x,\emptyset} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \emptyset & -\sin \emptyset \\ 0 & \sin \emptyset & \cos \emptyset \end{bmatrix} \quad (4.3)$$

Hemos descrito a estas rotaciones en términos de matrices de rotación $R_{z,\psi}, R_{y,\theta}, R_{x,\phi}$; entonces nuestra matriz de rotación general R puede ser descrita como:

$$R = R_{z,\psi} * R_{y,\theta} * R_{x,\phi} \quad (4.4)$$

Resolviendo la operación nuestra matriz R quedaría de la siguiente manera:

$$R = \begin{bmatrix} \cos \varphi \cos \theta & -\sin \varphi \cos \theta + \cos \varphi \sin \theta \sin \phi & \sin \varphi \sin \theta + \cos \varphi \cos \theta \sin \phi \\ \sin \varphi \cos \theta & \cos \varphi \cos \theta + \sin \varphi \sin \theta \sin \phi & -\cos \varphi \sin \theta + \sin \varphi \cos \theta \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (4.5)$$

Una vez obtenida nuestra matriz R, procedemos a utilizar el modelo de Euler-LaGrange, el cual relaciona la energía cinética de un cuerpo con su energía potencial.

La energía cinética de traslación (x,y,z) viene dada por:

$$E_{CT} = \frac{1}{2} * m * \varepsilon^2 \quad (4.6)$$

Donde:

$m =$ masa del cuadricoptero

$\varepsilon =$ es la velocidad lineal en cada uno de los ejes(x, y, z)

La energía cinética rotacional esta dada por :

$$E_{CR} = \frac{1}{2} * I * \eta^2 \quad (4.7)$$

Donde:

$I =$ momento de inercia

$\eta =$ es la velocidad angular en cada uno de los ejes

A continuación mostramos la energía potencial que viene dada por la siguiente formula:

$$E_p = m * g * z \quad (4.8)$$

$m =$ masa del cuadricoptero

$g = \text{gravedad}$

$z = \text{viene a ser la altura que se encuentra el cuadricoptero en el espacio}$

Una vez obtenidas estas ecuaciones podemos calcular el Lagrangiano que viene dado por la siguiente ecuacion:

$$L = K - U \quad (4.9)$$

Donde:

$L = \text{lagrangiano}$

$K = \text{energia cinetica(tanto rotacional como traslacional)}$

$U = \text{energia potencial}$

Entonces la ecuacion queda de la siguiente manera :

$$L = \left(\left(\frac{1}{2} * I * \eta^2 \right) + \left(\frac{1}{2} * m * \varepsilon^2 \right) \right) - m * g * z \quad (4.10)$$

El modelo dinamico del cuadricoptero, se obtiene por el metodo de euler-lagrange dado que este se basa en la energia del sistema, con la fuerza externa generalizada, obtenemos:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = F_\varepsilon \quad (4.11)$$

Donde:

$F_\varepsilon = \text{Fuerzas generalizadas aplicadas sobre las articulaciones(motores)}$

$\partial q_i = \text{coordenadas generalizadas(posicion de las articulaciones)}$

$\partial \dot{q}_i = \text{velocidades de las articulaciones}$

$\partial L = \text{derivada parcial del lagrangiano}$

Donde

F_ε representa la fuerza traslacional aplicada al cuadricoptero debido a la fuerza de sustentacion de los 4 motores en la direccion del eje z inercial.

F_b es la fuerza que se le aplica al vehiculo. La fuerza de sustentacion del cuadricoptero se encuentra en el eje z no inercial ,por lo que se multiplica

por la matriz de orientacion del vehiculo R, unicamente en la componente z de nuestra matriz de rotacion R que fue calculada mediante los angulos de euler.

Esta componente la obtenemos de la matriz R que ya se determino:

$$F_{\varepsilon} = \begin{bmatrix} \sin\Psi\sin\Phi + \cos\Psi\cos\Phi\sin\theta \\ -\cos\Psi\sin\Phi + \sin\theta\sin\Psi\cos\Phi \\ \cos\Phi\cos\theta \end{bmatrix} * F_b \quad (4.12)$$

Ahora podemos descomponer la fuerza traslacional en 3 fuerzas, una para cada variable traslacional generalizada.

$$F_{\varepsilon x} = (\sin\Psi\sin\Phi + \cos\Psi\cos\Phi\sin\theta) * F_b \quad (4.13)$$

$$F_{\varepsilon y} = (-\cos\Psi\sin\Phi + \sin\theta\sin\Psi\cos\Phi) * F_b \quad (4.14)$$

$$F_{\varepsilon z} = (\cos\Phi\cos\theta) * F_b \quad (4.15)$$

Podemos separar estas ecuaciones en dinamica de traslacion y rotacion, las fuerzas traslacionlaes dependen de las variables generalizadas de orientacion.

A continuacion se determina la ecuacion de movimiento con respecto a al eje X:

$$L(\varepsilon, \dot{\varepsilon}) = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} \quad (4.16)$$

$$L(\varepsilon, \dot{\varepsilon}) = E_{CT} - E_P \quad (4.17)$$

$$L(\varepsilon, \dot{\varepsilon}) = \frac{1}{2}m * (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) - mgz \quad (4.18)$$

Usando euler lagrange:

$$\left(\frac{\partial L}{\partial \dot{x}} \right) = m\dot{x} \quad (4.19)$$

Derivando la energia potencial con respecto a X:

$$\frac{\partial L}{\partial x} = 0 \quad (4.20)$$

Derivando la velocidad con respecto al tiempo(aceleracion)

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) = m\ddot{x} \quad (4.21)$$

Determinamos la ecuacion del movimiento con respecto a Y:

Usando euler lagrange:

$$\left(\frac{\partial L}{\partial \dot{y}}\right) = m\dot{y} \quad (4.22)$$

Derivando la energia potencial con respecto a Y:

$$\frac{\partial L}{\partial y} = 0 \quad (4.23)$$

Derivando la velocidad con respecto al tiempo(aceleracion)

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}}\right) = m\ddot{y} \quad (4.24)$$

Determinamos la ecuacion del movimiento con respecto a Z:

Usando euler lagrange:

$$\left(\frac{\partial L}{\partial \dot{z}}\right) = m\dot{z} \quad (4.25)$$

Derivando la energia potencial con respecto a Y:

$$\frac{\partial L}{\partial z} = mg \quad (4.26)$$

Derivando la velocidad con respecto al tiempo(aceleracion)

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{z}}\right) = m\ddot{z} \quad (4.27)$$

Escribiendo las tres ecuaciones de movimiento traslacional:

$$F_{\epsilon x} = m\ddot{x} \quad (4.28)$$

$$F_{\epsilon y} = m\ddot{y} \quad (4.29)$$

$$F_{\epsilon z} = m\ddot{z} + mg \quad (4.30)$$

Reemplazando las ecuaciones en las ecuaciones:

$$m\ddot{x} = (\sin\Psi\sin\Phi + \cos\Psi\cos\Phi\sin\theta) * F_b \quad (4.31)$$

$$m\ddot{y} = (-\cos\Psi\sin\Phi + \sin\theta\sin\Psi\cos\Phi) * F_b \quad (4.32)$$

Usamos el metodo de euler lagrange para obtener las ecuaciones traslacionales, ya que la posicion espacial del cuadricoptero se va a medir fisicamente respecto a un marco fijo al vehiculo no inercial.

4.1.2.-OBTENCION DE ECUACIONES DE MOVIMIENTO ROTACIONAL (FORMULACION NEWTON –EULER)

Usaremos el metodo newton-euler para conocer la orientacion del vehiculo respecto a un marco de referencia no inercial, debido a que los angulos reales se presentan en el cuadricoptero son medidos por la IMU (unidad de medicion inercial) gracias al acelerometro y giroscopio.

A continuacion obtendremos solamente las tres ecuaciones de rotacion respecto al marco no inercial o al marco ligado al cuadricoptero mediante la formulacion de Newton-Euler:

$$I_{3 \times 3} \dot{\omega} + \omega * (I_{3 \times 3} \omega) = \tau \quad (4.33)$$

Donde:

$I_{3 \times 3}$

= matriz de inercia tiene una estructura simetrica con respecto X, Y, Z

por lo tanto la matriz de inercia es la diagonal

$\dot{\omega}$ = velocidad angular

τ = torque o momentos generalizados

DESARROLLO DE LA ECUACION:

$$I_{3 \times 3} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (4.34)$$

$$\dot{\omega} = \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} \quad (4.35)$$

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (4.36)$$

$$\tau = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (4.37)$$

Reemplazando las variables en la ecuación :

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} * \left(\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \right) = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (4.38)$$

Resolviendo en sistema obtenemos:

$$\tau_\phi = I_{xx}\dot{\omega}_1 + \omega_2\omega_3I_{zz} - \omega_2\omega_3I_{yy} \quad (4.39)$$

$$\tau_\theta = I_{yy}\dot{\omega}_2 + \omega_1\omega_3I_{xx} - \omega_1\omega_3I_{zz} \quad (4.40)$$

$$\tau_\psi = I_{zz}\dot{\omega}_3 + \omega_2\omega_1I_{xx} - \omega_2\omega_1I_{yy} \quad (4.41)$$

DESPEJAMOS $\dot{\omega}_1, \dot{\omega}_2, \dot{\omega}_3$

$$\dot{\omega}_1 = \tau_\phi - \frac{\omega_2\omega_3(I_{zz}-I_{yy})}{I_{xx}} \quad (4.42)$$

$$\dot{\omega}_2 = \tau_\theta - \frac{\omega_1\omega_3(I_{xx}-I_{zz})}{I_{yy}} \quad (4.43)$$

$$\dot{\omega}_3 = \tau_\psi - \frac{\omega_2\omega_1(I_{xx}-I_{yy})}{I_{zz}} \quad (4.44)$$

4.2.-ESTRUCTURA DE CUADRICOPTERO

4.2.1.-CHASIS DJI F450 MM

Consiste en una estructura que sostiene y aporta rigidez y forma a un vehículo, este se encargará de dar la ubicación correcta a los elementos que componen el prototipo, permitiendo versatilidad y buena funcionalidad entre los mismos.

4.2.1.1.-CARACTERÍSTICAS

Características de chasis F450	
Ancho	450 mm
Altura	55 mm
Peso	280 gramos (sin electrónica)
Distancia entre ejes	17.7in/450mm
Agujeros de tornillos	16/19mm

Tabla 4.1: Características chasis dji f450

4.2.2 PLACA DE DISTRIBUCION DE VOLTAJE

Placa que se encargara de la distribucion de energia,siendo una opcion util y definitiva para la distribucion uniforme y ordenada de la energia .

Incluye una entrada principal por la cual se alimentara a la placa y asi poder distribuir el voltaje a los variadores.

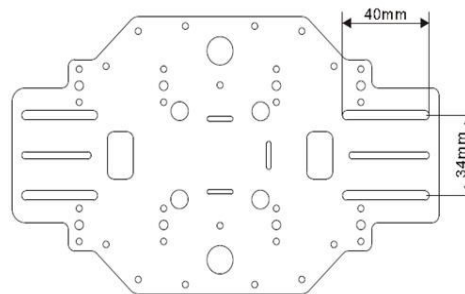


Figura4.3: Placa de distribución de voltaje

4.2.2.1.-HELICES

Estos elementos son los encargados de aprovechar el giro del motor brushless y convertirlo en un flujo de aire ascendente, vienen de diferentes tamaños y características, se debe seguir la siguiente secuencia en el cuadricoptero, que por cada helice instalada debe haber una inversa para que pueda existir un equilibrio en el vuelo.



Figura4.4: Hélices 10x45

CARACTERISTICAS HELICES	
COLOR	variado
Material	ABS +FIBRA
Diametro helice	25,4 cm
diametro del eje	6,0 mm
cojines propulsor	2mm / 3mm / 4mm / 5mm / 6mm/7,8 mm
espesor del centro	9,7 mm
distancia entre ejes	550mm - 700mm
motor recomendado	800KV - 1100KV
Longitud	10 pulgadas
Peso	14 gramos

Tabla4.2: Características de hélices

4.2.2.2.BATERIA LIPO C30 4000 mAh HOBBYLINE



Figura4.5: Batería 30C ChinaHobbyLine

Características	
Capacidad	4000mAh
Voltaje	11.1v/3 celdas/3s
Tasa constante de descarga	30C
Tamaño	26X45X148mm
Peso	328g
tasa constante de carga	5C Max
Descarga (picos)	60 C

Tabla4.3: Características batería lipo hobbyline

4.2.2.3.BATERIA LIPO C45 5400 mAh TIGER POWER



Figura4.6: Batería 45C Tiger Power

CARACTERISTICAS	
Voltaje nominal	11.1 voltios
Temperatura de trabajo	0~45C
Capacidad	5400mAh
Peso	365 gramos
Tasa constante de carga	5C
Descarga	90C
Tasa constante de descarga	45C
Tamaño	140X43x30mm

Tabla4.4: Características batería Tiger Power

4.2.2.4.-MOTOR BRUSHLESS A2212/13T 1000KV

CARACTERISTICAS A2212/13T	
MODELO	A2212/13
ELEMENTOS	2-3 Li-Poly (11.1V)
RPM	1000 rpm
Eficiencia máxima	80%
Eficiencia máxima	4-10A (>75%)
Corriente sin carga	10 V:0,5 A
Corriente máxima	12A/60 segundos
Resistencia interna	90 mΩ
Dimensiones	27.5x30 mm
Diámetro del eje	3.17 mm
Peso	47 gramos

Tabla4.5: Características motor brushless A2212



Figura4.7: motor A2212/13T

4.2.2.5.-OBTENCION DE GRAFICAS DE DESEMPEÑO

Las graficas de desempeño obtenidas por el software drive calculator son las siguientes:

Se hizo la simulacion con el motor A2212-26 que posee las mismas características técnicas que el A2212/13T y también porque se hizo la simulacion con las helices 1045

- En la grafica se muestra el consumo de corriente aproximado de cada motor y las rpm alcanzadas por cada uno de ellos.



Figura4.8: Grafica de velocidad, potencia y eficiencia del motor

Fuente: Propia

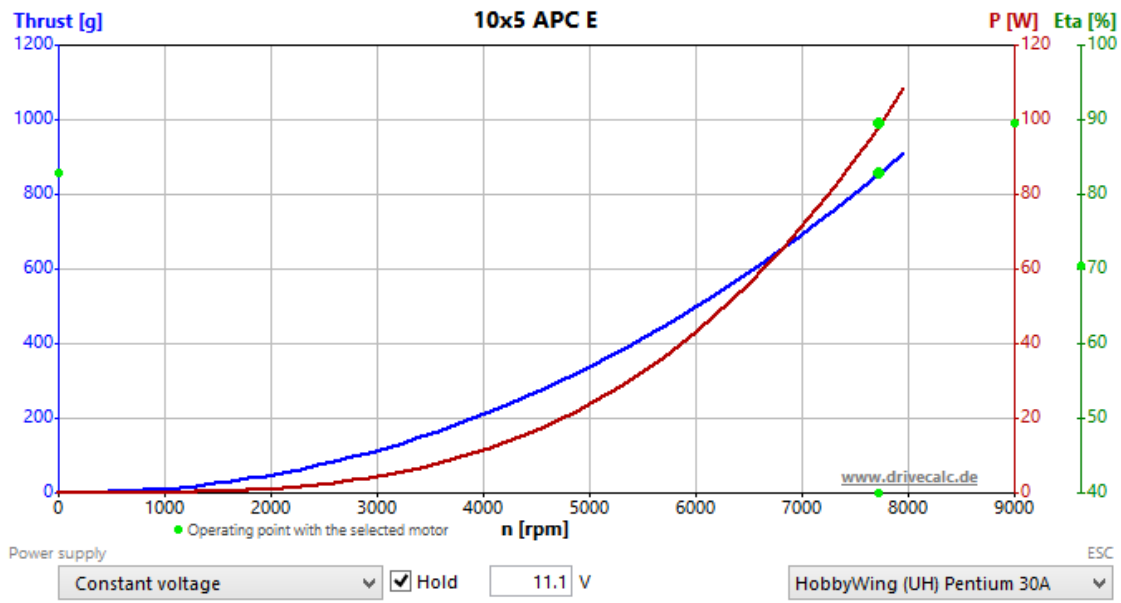


Figura4.9: Grafica de empuje y potencia del motor

Fuente: Propia

4.2.2.6.- GRAFICAS OBTENIDAS PARA LA SELCCION DE HELICES

Para la obtencion de graficas se empleo el software PropCalc y tambien trabajamos con uno de los perfiles mas usados CLARK-Y .

El clark-y es uno de los perfiles mas estudiados en el diseño aeronautico.

Tiene un espesor de 11.7%.

• PRUEBA CON HELICES 1045

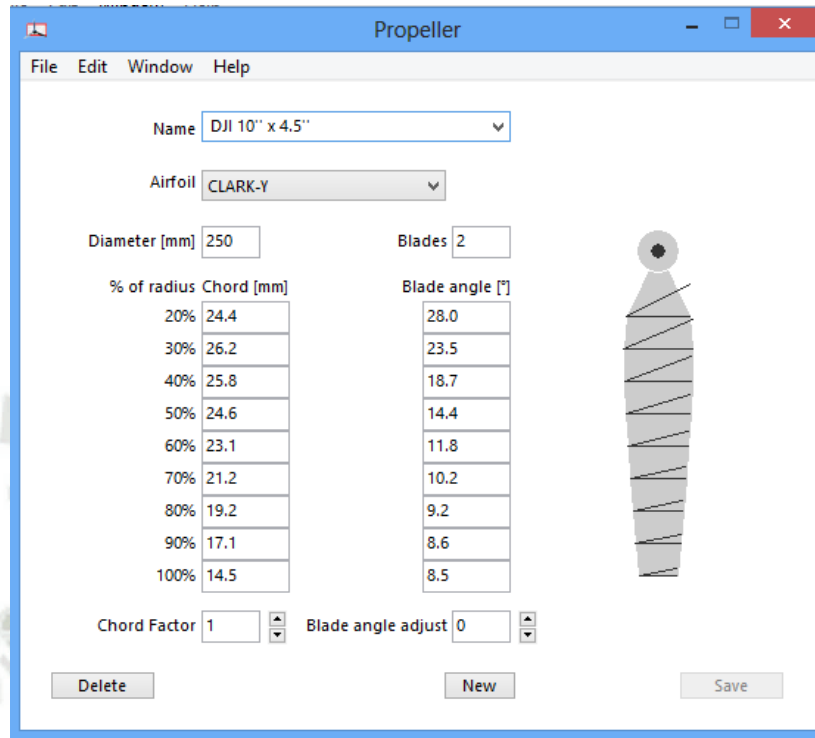


Figura4.10: Configuración de perfil CLARK-Y

Fuente: Propia

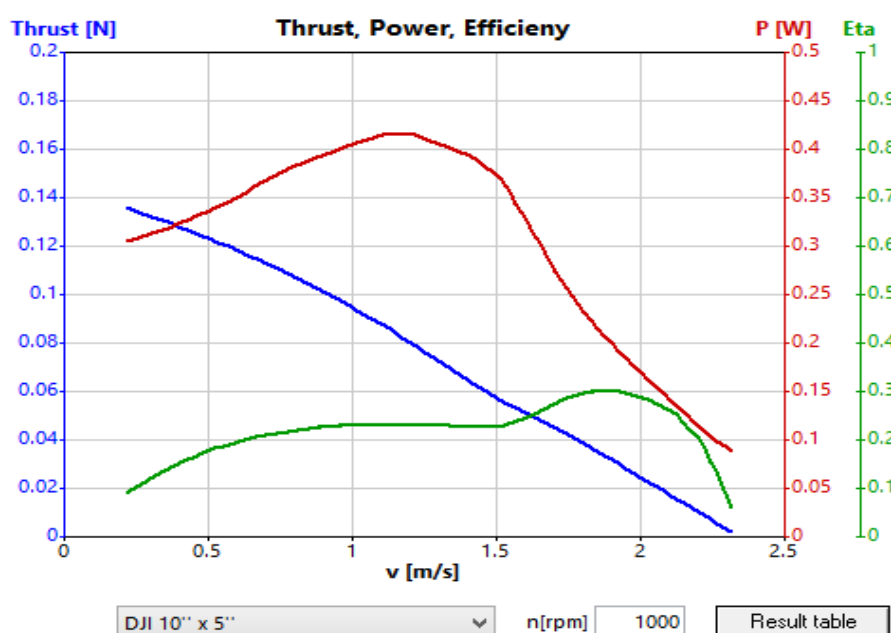


Figura411: Gráfica de potencia ,empuje y eficiencia

Fuente: Propia

4.2.2.5.-CALCULO DE DURACION DE BATERIA HOBBYLINE 30C 4000mAh

Considerando:

$$\text{Capacidad} = 4000\text{mAh}$$

$$\text{Corriente demanda por motores} = 12700 \text{ mA} * 4 = 50800\text{mA}(4.45)$$

$$\text{duracion} = \frac{\text{capacidad}}{\text{corriente demanda por motores}}(4.46)$$

$$\text{duracion} = \frac{4000\text{mAh}}{50800\text{mA}}(4.47)$$

$$\text{duracion} = 0.0787 \text{ h}(4.48)$$

Cambio de horas a minutos:

$$\text{duracion en min} = 0.0787\text{h} * \left(\frac{60\text{min}}{1\text{h}}\right)(4.49)$$

$$\text{duracion en min} = 4.72 \text{ minutos}(4.50)$$

El tiempo de vuelo aproximado de nuestro prototipo sera de 4.7 minutos.

Una vez obtenida la duracion de tiempo de vuelo con la bateria, es necesario comprobar que la bateria es capaz de suministrar la corriente maxima demandada, esto se hace comparando el valor de descarga normal de la bateria 30C contra dicho valor maximo de corriente.

$$\text{Tasa de descarga constante 30C} = 30 * 4000\text{mA} = 120000\text{mA}(4.51)$$

$$\text{corriente maxima demandada} = 12700\text{mA} * 4 = 50800\text{mA}(4.52)$$

$$120000\text{mA} > 50800\text{mA}(4.53)$$

Por lo tanto la bateria puede suministrar la corriente maxima.

4.2.2.6.-CALCULO DE DURACION DE BATERIA TIGER POWER 5400mAh

Considerando:

$$\text{Capacidad} = 5400\text{mAh}$$

$$\text{Corriente demanda por motores} = 12700 \text{ mA} * 4 = 50800\text{mA}(4.54)$$

$$\text{duracion} = \frac{\text{capacidad}}{\text{corriente demanda por motores}}(4.55)$$

$$\text{duracion} = \frac{5400\text{mAh}}{50800\text{mA}}(4.56)$$

$$\text{duracion} = 0.1063 \text{ h}(4.57)$$

Cambio de horas a minutos:

$$\text{duracion en min} = 0.1063\text{h} * \left(\frac{60\text{min}}{1\text{h}}\right)(4.58)$$

$$\text{duracion en min} = 6.4 \text{ minutos}(4.59)$$

Seguidamente como se hizo en el paso anterior comprobaremos si la batería es capaz de suministrar la corriente máxima demandada.

$$\text{Tasa de descarga constante } 45\text{C} = 30 * 5400\text{mA} = 243000\text{mA}(4.60)$$

$$\text{corriente máxima demandada} = 12700\text{mA} * 4 = 50800\text{mA}(4.61)$$

$$243000\text{mA} > 50800\text{mA}(4.62)$$

Por lo tanto la batería es capaz de suministrar la corriente máxima.

4.2.3.-PROCESO DE EMSAMBLADO (PARTE HARDWARE)

4.2.3.1.-ARMADO DE CHASIS

Se empezara por organizar las bases y brazos que conformaran la estructura del cuadricoptero,determinando que existen un par de brazos de distinto color, a eleccion los brazos negros señalaran la parte delantera del cuadricoptero

4.2.3.2.-CONEXIÓN DE VARIADORES A PLACA DISTRIBUIDORA DE VOLTAJE

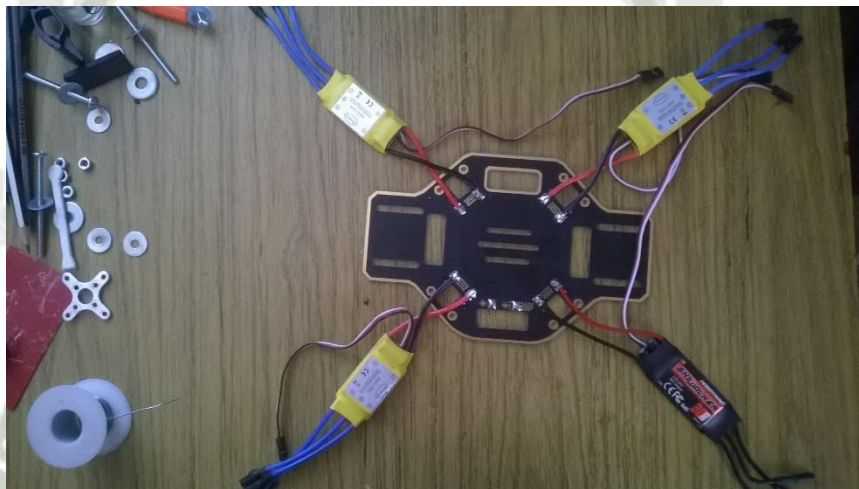


Figura4.12: Variadores soldados a tarjeta distribuidora de voltaje

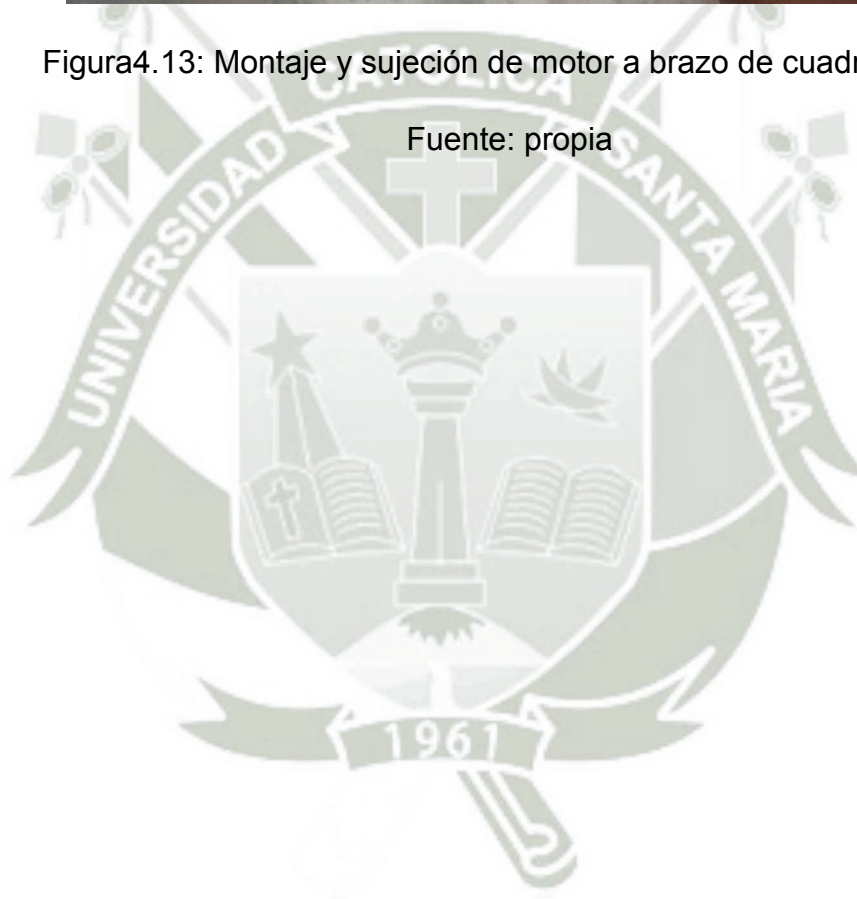
Fuente: propia

4.2.3.3.-MONTAJE DE LOS MOTORES A2212/13T



Figura4.13: Montaje y sujeción de motor a brazo de cuadricoptero

Fuente: propia



4.2.3.4.-MONTAJE DE VARIADORES ESC 30 AMPERIOS



Figura4.14: conexionado de variadores a motores y sujeción a brazo de cuadricoptero

Fuente: propia

4.2.3.5.-MONTAJE DE BRAZOS A PLACA DISTRIBUIDORA DE VOLTAJE

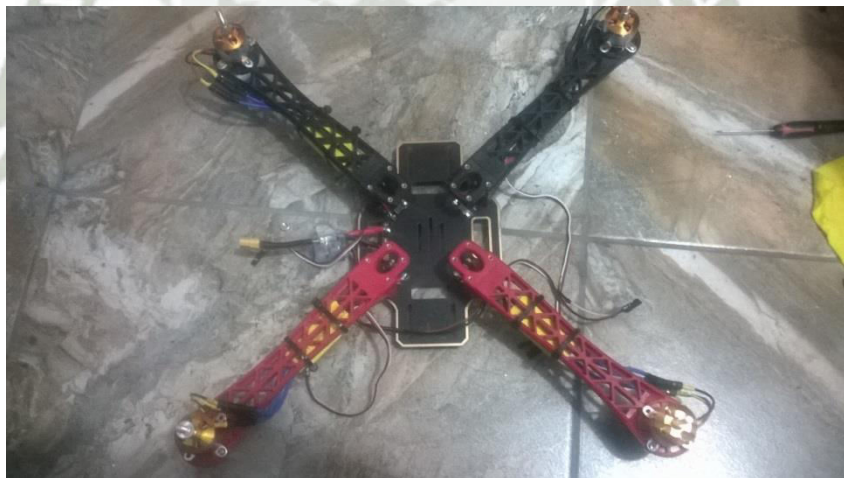


Figura4.15: Montaje de brazos a tarjeta distribuidora de voltaje

Fuente:propia

4.2.3.6.-CONEXIONADO DE LOS MOTORES A LA TARJETA

Debido a que en el programa misión planner se configuro el tipo de configuración de motores estos irán según se muestra en la figura:



Figura4.16: sentido de conexión de variadores a tarjeta apm

Fuente: <http://ardupilot.org/>

Otro aspecto que se tiene que tener en cuenta en el sentido de giro de los motores para poder instalar correctamente las hélices, el cambio de giro se hace cambiando los cables de motor.

Otra alternativa es haciéndolo desde Mission Planner, pero dada la versión instalada en nuestra controladora no se podrá entonces solo se hizo un cambio en las líneas de salida del motor.

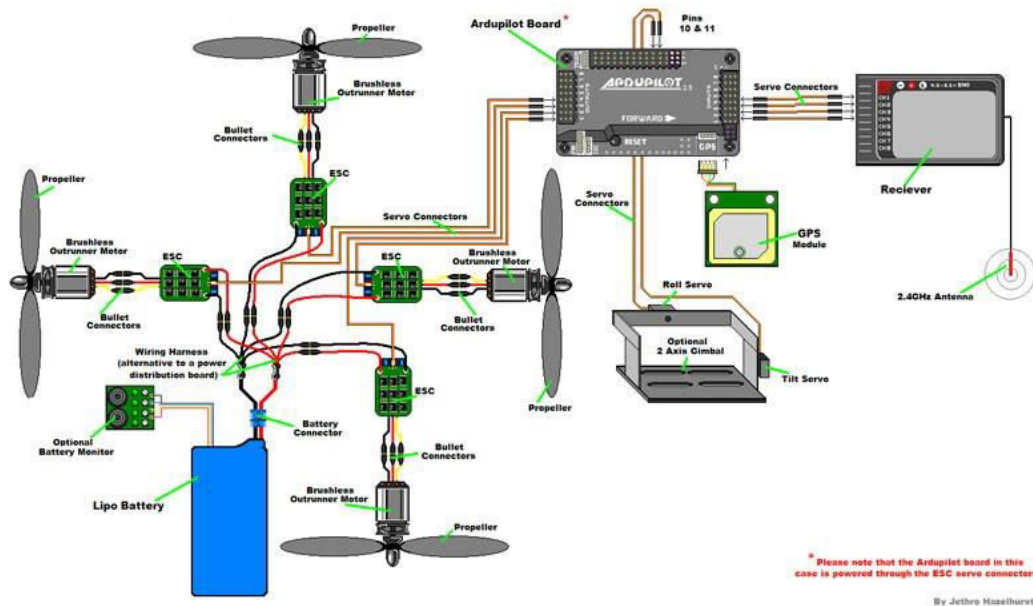


Figura4.17: Diagrama de conexión de motores, variadores a tarjeta apm

Fuente: <http://ardupilot.org/>

4.2.3.7.-CONEXIONADO DE RECEPTOR FLYSKY A LA TARJETA

Del receptor se sacaran los cables de señal que iran conectados a la parte de entrada de la tarjeta apm 2.8, como se ve en la figura4.17.

El primer canal del receptor es correspondiente al alerón y se conecta en la entrada 1 de la tarjeta, el elevador que corresponde al canal 2 en el receptor se conecta a la entrada 2 de la tarjeta, el canal 3 correspondiente al acelerador en el receptor se conecta a la entrada 3 en la tarjeta, el canal 4 correspondiente al timón de cola del receptor es conectado a la entrada 4 de la tarjeta, el canal 5 es un auxiliar en el control se puede programar con cualquiera de los botones del control, que en este caso es un switch de varias posiciones el cual va a controlar los modos de vuelo y va conectado a la entrada 5 de la tarjeta.

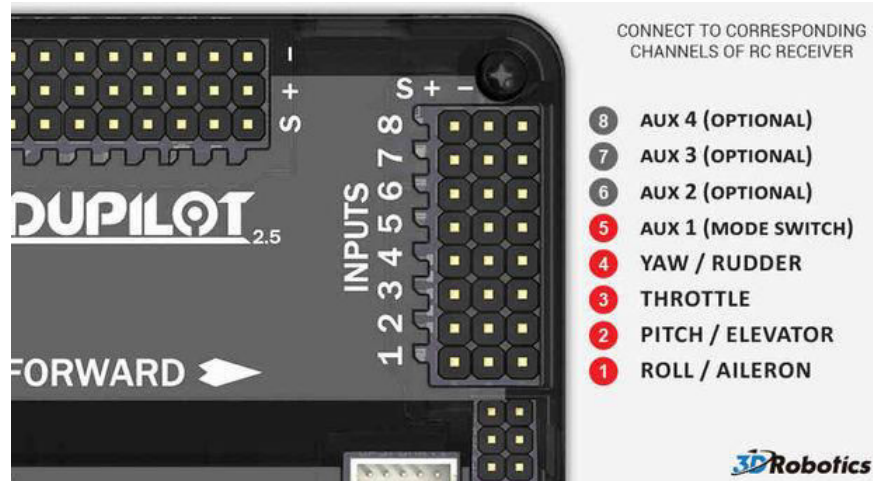


Figura4.18: conexión de receptor a tarjeta apm

Fuente: <http://ardupilot.org/>

4.2.3.8.-MONTAJE DE LA PLACA APM 2.8 Y RECEPTOR Y MODULO GPS

Para poder obtener un sistema completamente funcional tendremos que ubicar de forma correcta la tarjeta controladora, internamente esta cuenta con el acelerómetro/giroscopio MPU6000 y una flecha apuntando a lo que vendría siendo el norte.

Tendremos que ubicar el sensor de forma correcta en el centro de la tarjeta distribuidora de voltaje apuntando a lo que vendrá siendo la cabeza del cuadricóptero, el módulo de gps debe de ir en la misma dirección del MPU 6000.

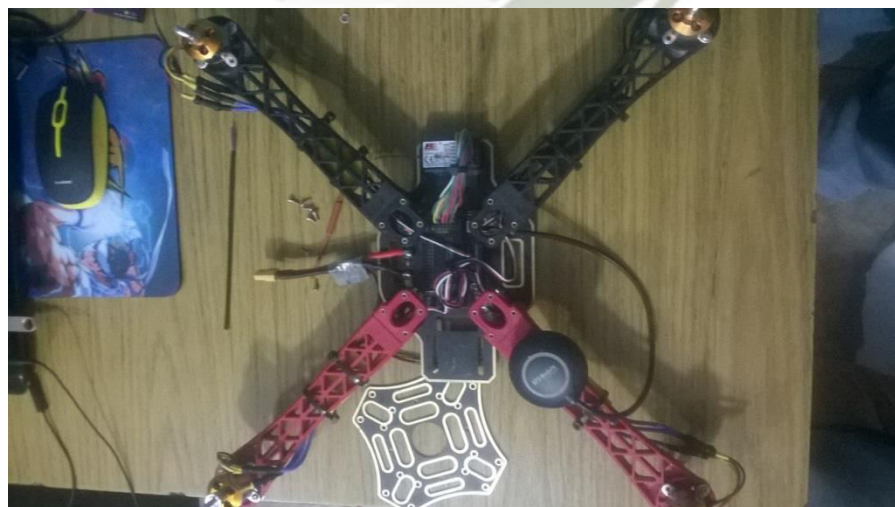


Figura4.19: Incorporación de tarjeta y módulo GPS

Fuente: propia

CAPITULO V

CONTROL DE CUADRICOPTERO

5.1. TARJETA APM 2.8

Es una controladora de vuelo de altas prestaciones para varios modelos de copters, aviones, rovers.



Figura5.1:apm2.8

Fuente: Propia

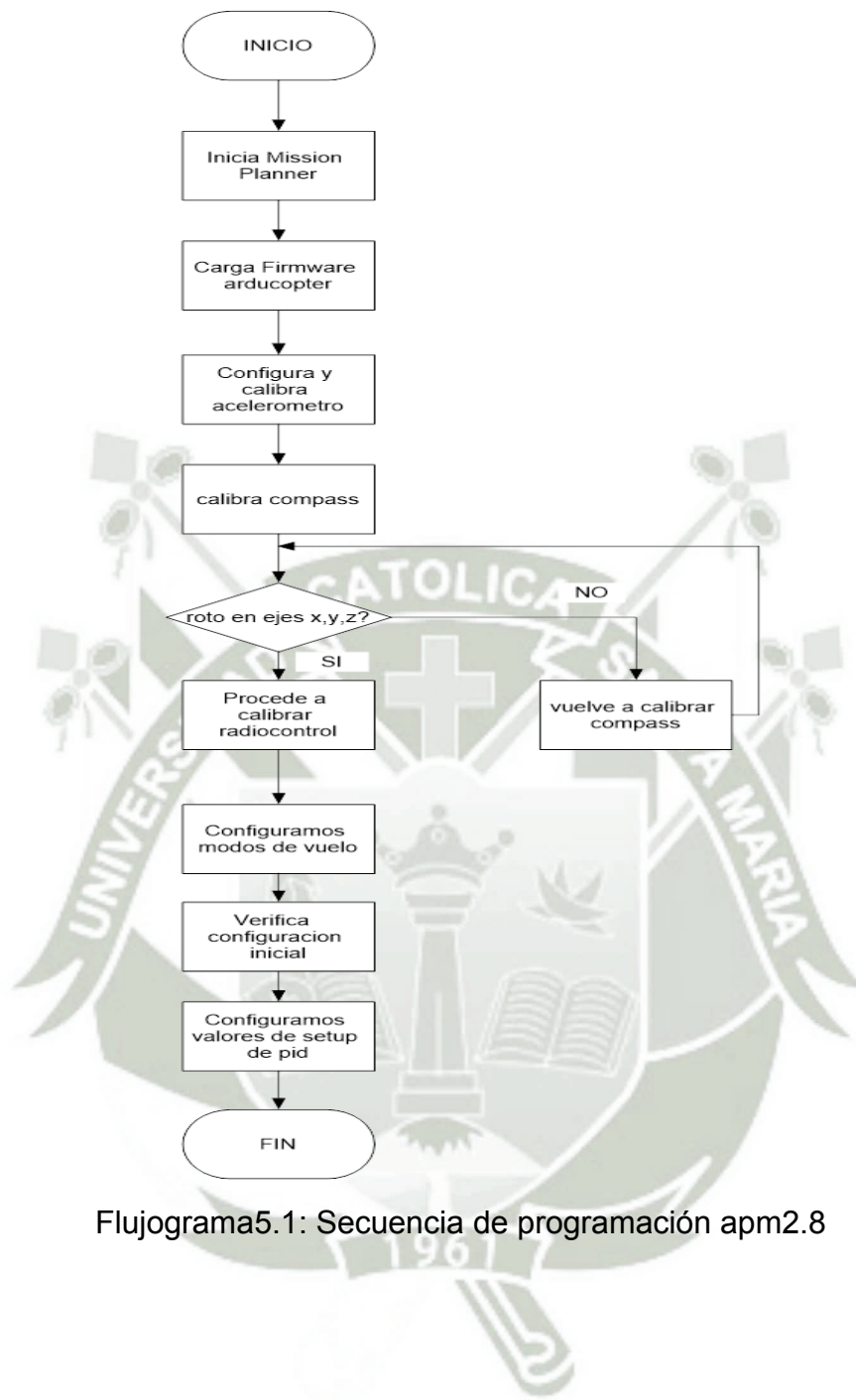
EL APM 2.8 es un sistema autopiloto completamente Open Source que permite convertir a cualquier vehículo aéreo, avión, helicóptero, multicoptero e incluso carros robots (rovers) y botes en un vehículo completamente autónomo, capaz de realizar misiones programadas con posicionamiento GPS.

5.1.1. CARACTERISTICAS

CARACTERISTICAS APM 2.8	
Procesadores	ATMEGA2560 y ATMEGA32U-2
Modulo GPS	Neo-7M
Sensor Barométrico	MS5611-01BA03.
Giroscopio/Acelerómetro	MPU-6000.
Controlador de energía	RC150.
Sistema de telemetría	915MHz
Salidas de comunicación	UART1, UART2, I2C y OSD

Tabla5.1: Características apm2.8

5.2. DIAGRAMA DE FLUJO DE PROGRAMACION DE APM



Flujograma5.1: Secuencia de programación apm2.8

5.3 CONFIGURACION DE TARJETA APM2.8 EN MISSION PLANNER

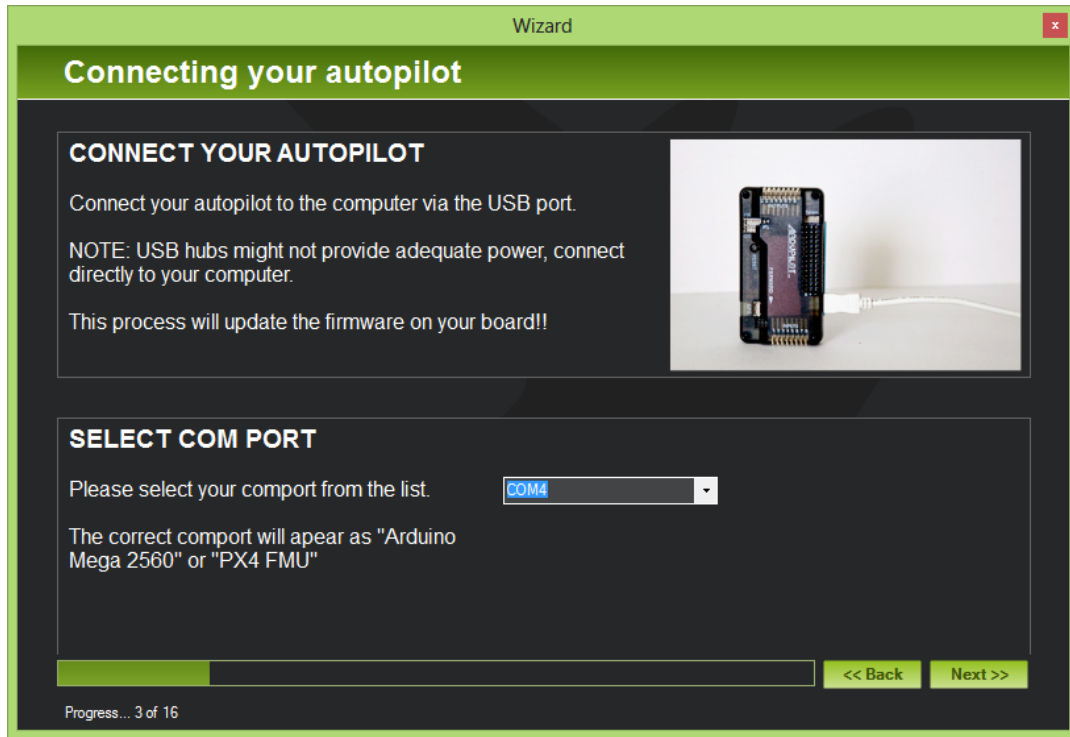


Figura5.2: Conexión de apm 2.8 con misión planner

Fuente: propia

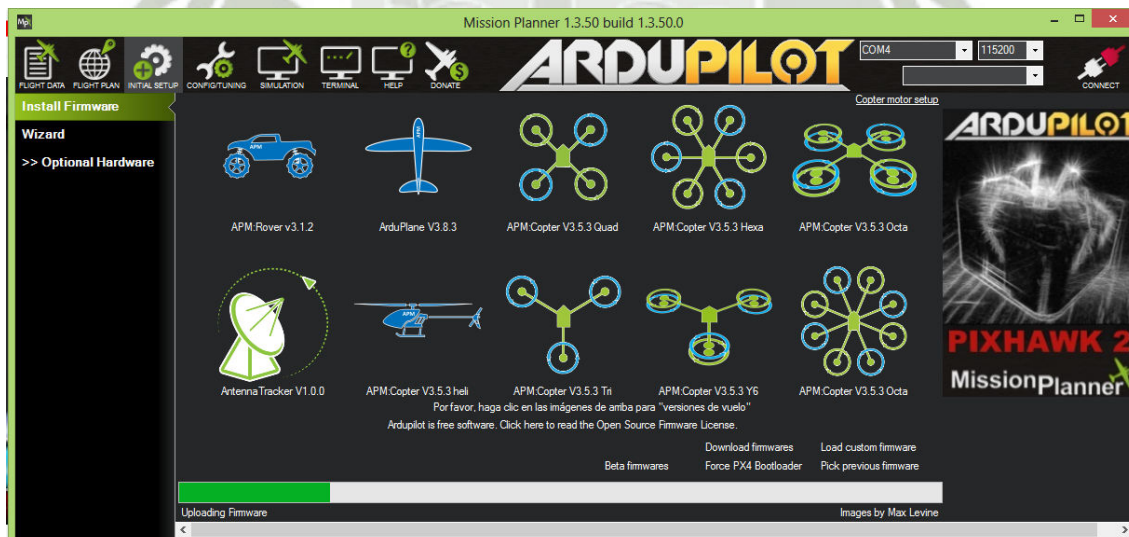


Figura5.3: Descarga del firmware mission planner

Fuente: propia



Figura5.4: Wizard mission planner

Fuente: propia

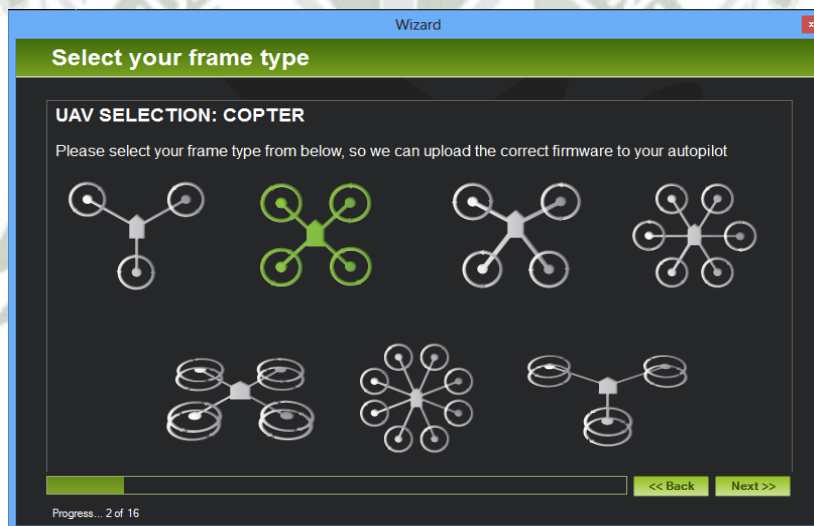


Figura5.5: Selección de frame

Fuente: Propia

Después de haber realizado la selección de la forma del frame que tendrá nuestro cuadricoptero, se instalara y se cargara el firmware a la tarjeta apm.

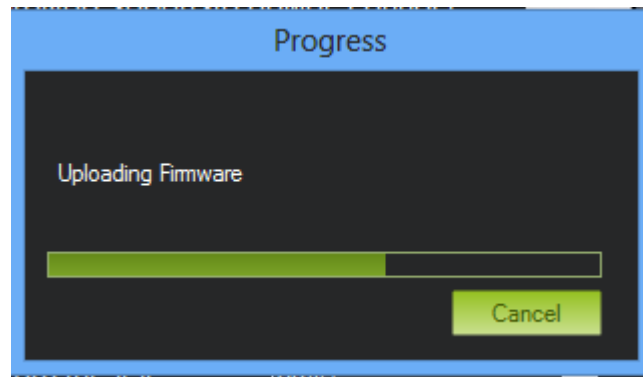


Figura5.6: Cargando firmware a apm 2.8

Fuente: propia

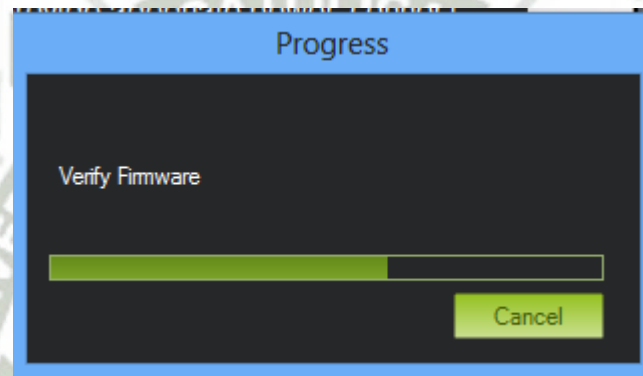


Figura5.7: Verificación de firmware instalado

Fuente: Propia

5.4. CALIBRACION DEL ACELEROMETRO

Para poder calibrar adecuadamente el acelerómetro de la tarjeta apm2.8, necesitamos seguir una secuencia de pasos que nos presenta el asistente de configuración de misión planner.

Este proceso requiere que usted coloque el cuadricoptero en las posiciones que se muestran a continuación en la secuencia. La posición de nivel es el más importante para hacerlo bien ya que esta será la actitud de que su controlador considera nivel durante el vuelo. Es importante que no se mueva el cuadricoptero inmediatamente después de pulsar la tecla para cada paso.

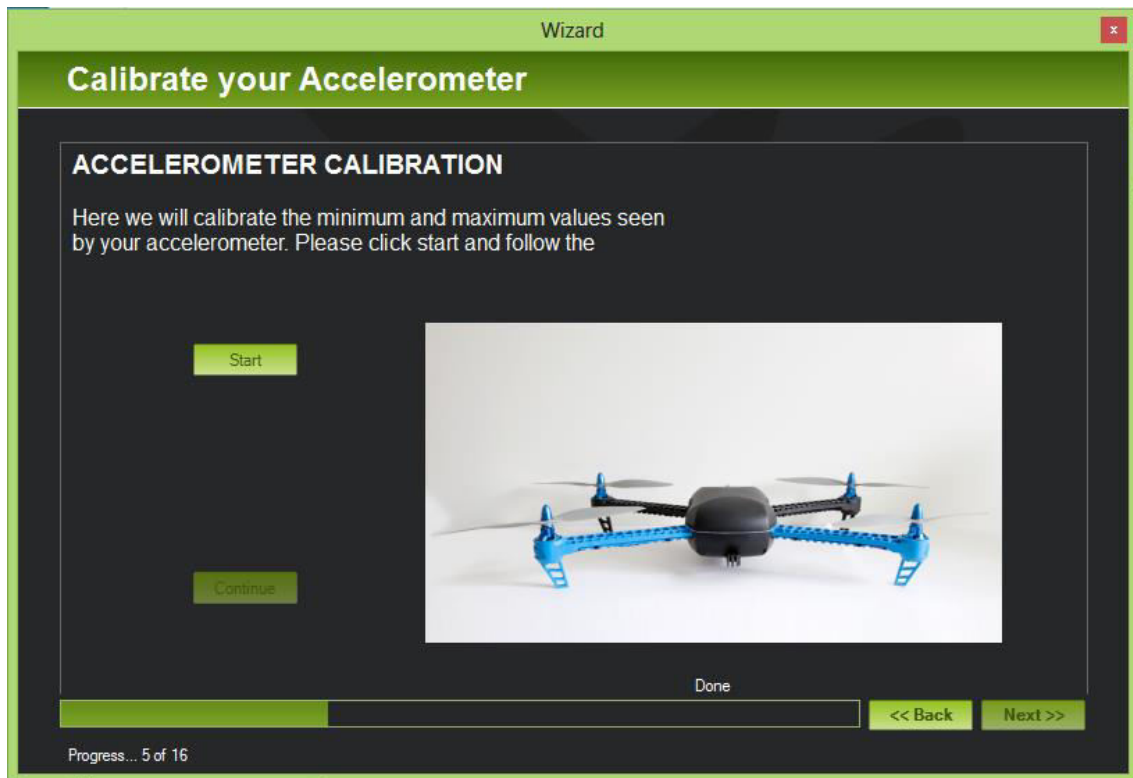


Figura5.8: Procedimiento de calibración de acelerometro

Fuente: propia



Figura5.9: Secuencia de calibración

Fuente: <http://ardupilot.org/copter/docs/common-accelerometer-calibration.html>

5.5. CALIBRACION DE COMPASS

Para la configuración del compás, es necesario mover al cuadricoptero sobre todos sus ejes (x, y, z) de forma circular, obteniendo así el mayor número de puntos que se irán mostrando en el misión planner. Al llegar a un número de puntos determinado automáticamente el programa da por finalizado la calibración del compás magnético.

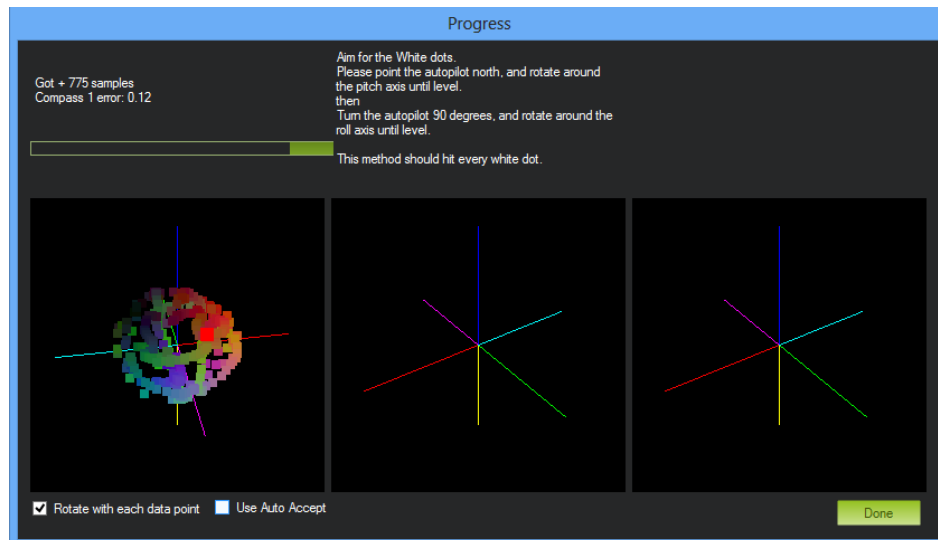


Figura5.10: Calibración de Compas

Fuente: Propia

Después de haber realizado la calibración del compás obtendremos los valores compensados. La suma de estos valores no debe exceder a 500, sino la calibración del compás será errada.

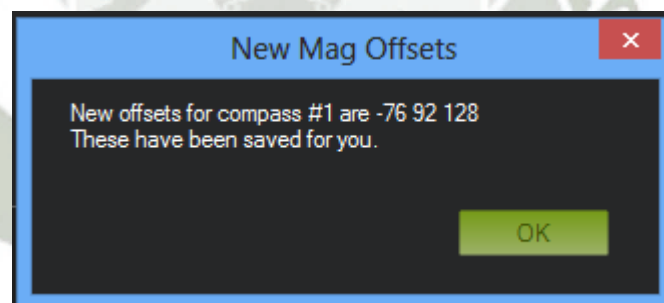


Figura5.11: Compensaciones obtenidas después de la calibración

Fuente: Propia

5.6. CONFIGURACION DE MODOS DE VUELO APM2.8

Stabilize: El modo 'Estabilizador' te permite volar el vehículo manualmente, pero auto nivela los niveles de 'roll' y 'pitch'.

Altitude hold mode: El modo de mantenimiento de altitud, el cóptero mantiene una altitud constante, permitiendo al piloto maniobrar el 'roll', 'pitch' y 'yaw'.

Loiter: El modo 'Merodear' automáticamente intenta mantener la misma posición, los grados y la altitud. El piloto debe volar en el modo 'Loiter' como si fuera 'manual'.

RTL Mode: En el modo 'Vuelta al lugar de lanzamiento (RTL)' el cóptero navega desde su actual posición, para volar alrededor de su zona de lanzamiento. El comportamiento del modo RTL se puede ajustar por varios parámetros reajustables.

Auto Mode: En el modo automático, el cóptero seguirá una misión de vuelo programada, almacenada en el autopiloto, que está hecha de comandos de navegación (ej. caminos) y 'hace' comandos (ej. comandos que no afectan la localización del cóptero, incluyendo el disparador de fotos de la cámara).

5.6.1. CONFIGURACION DE MODOS DE VUELO

Ahora el programa pide configurar los canales auxiliares, para este caso se configuro en el canal 5, un switch de 2 posiciones, 2 modos de vuelo.

Los modos de vuelo se definen como la forma en que el Drone se va a comportar cuando este en el aire, los modos de vuelo son 2, estabilizar (Stabilize) y mantener altitud (Alt Hold).

En el primero estabilizar (Stabilize), en este modo la tarjeta nos ayuda en vuelo con los acelerómetros y los giróscopos a mantener siempre el horizonte en el Drone después de tener un movimiento, auto corrige. En el segundo modo es mantener altitud (Alt Hold), en este modo la tarjeta toma de referencia por medio del barómetro la presión atmosférica a la cual fue activada este modo, en este modo se desactiva el acelerador en el mando y se vuelve autónoma la altitud.

En la figura 5.9 configuraremos los canales auxiliares para trabajar con los modos de vuelo stabilize y alt hold.

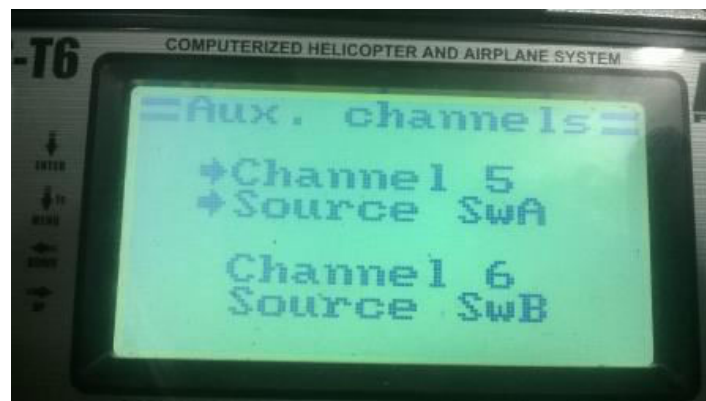


Figura5.12: Setup de canales auxiliares

Fuente: Propia

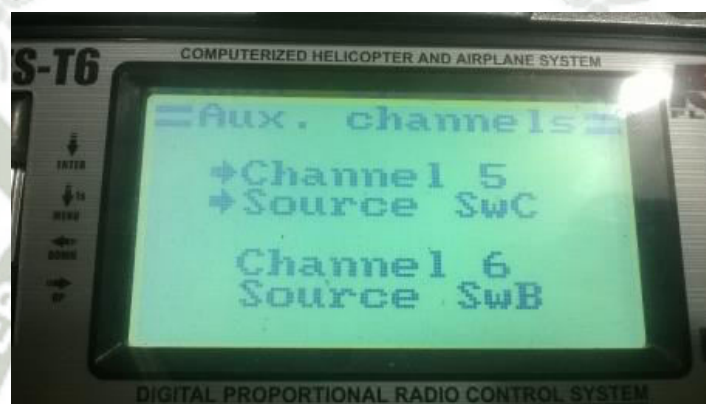


Figura5.13: Cambio de canales auxiliares swC=canal5 swD=canal6

Fuente: Propia

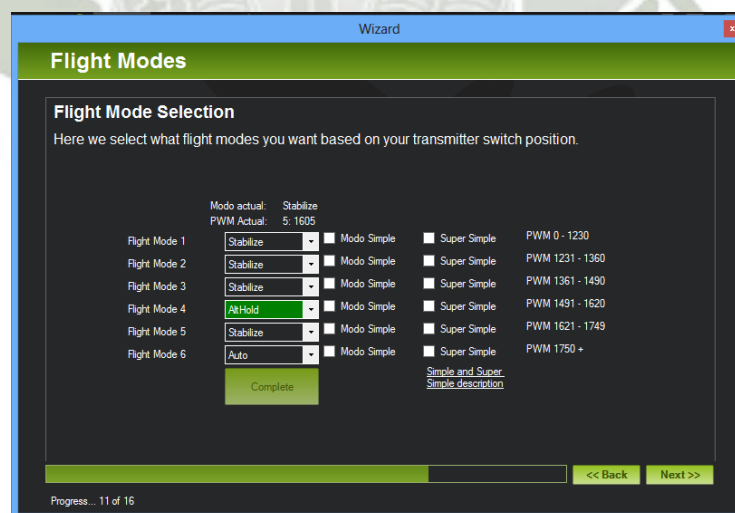


Figura5.14: Verificación de modos de vuelo

Fuente: Propia

5.7. CONFIGURACION DE EMISORA FLYSKY 6 CANALES

Los transmisores RC se usan para controlar el movimiento y la orientación del vehículo. El helicóptero y el avión controlan mínimamente el acelerador, cabeceo, balanceo y guiñada, mientras que en los Rover simplemente controlamos el acelerador y el balanceo. Cada una de estas señales de control se asigna a los stick/switch(s) del transmisor y, a su vez, a los canales de piloto automático desde el receptor conectado.

Calibrar cada uno de los controles y canales del transmisor es un proceso sencillo: simplemente mueva cada uno de los sticks/switches habilitados a través de su rango completo y registre las posiciones máxima y mínima.



Figura5.15: Calibración de emisora flysky

Fuente: Propia

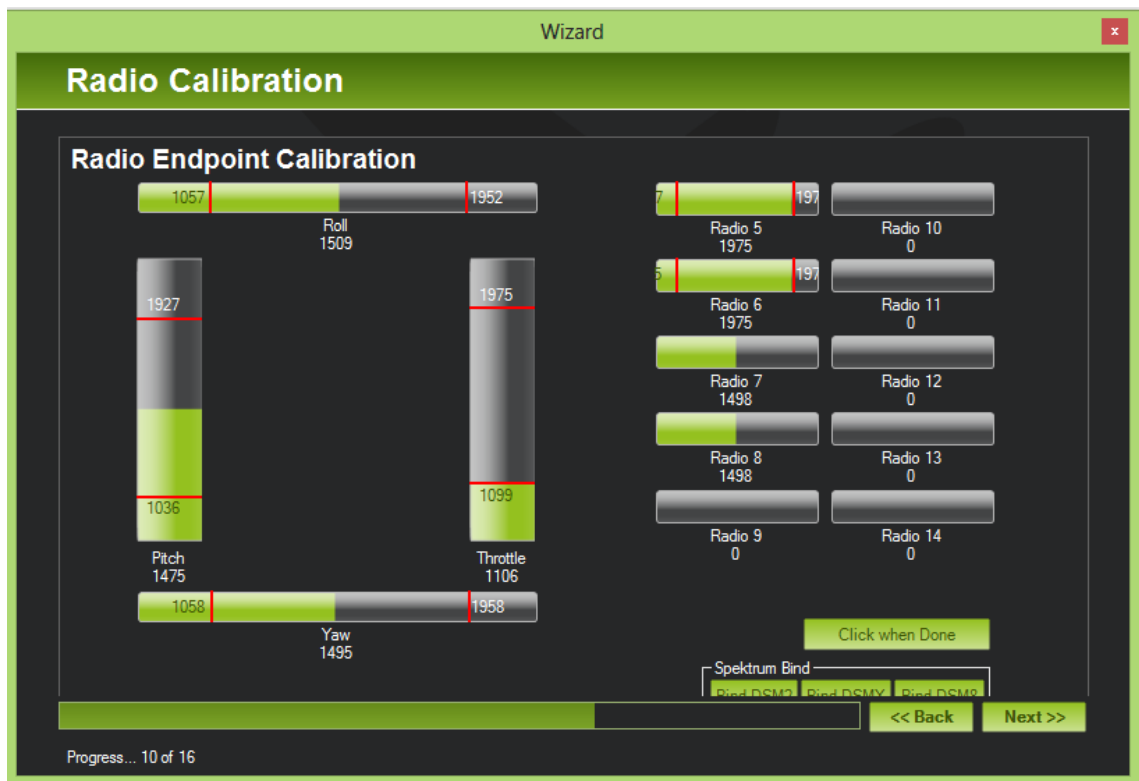


Figura 5.16: Valores Max y min de emisora

Fuente: Propia

5.8. MAPEO DE CANALES

Por defecto en Mission Planner la asignación de canales básicamente viene determinada de la siguiente manera:

Canal 1: Roll

Canal 2: Pitch

Canal 3: Throttle

Canal 4: Yaw

Canal 5 a 8 (default): Dependiendo de los canales que tenga el receptor.
Configura modos de vuelo

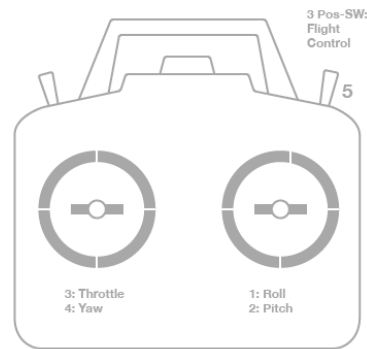


Figura5.17: Mapeo de canales emisora

Fuente: <http://ardupilot.org>

Mission Planner mostrará un resumen de los datos de calibración. Los valores normales son alrededor de 1100 para mínimos y 1900 para máximos.

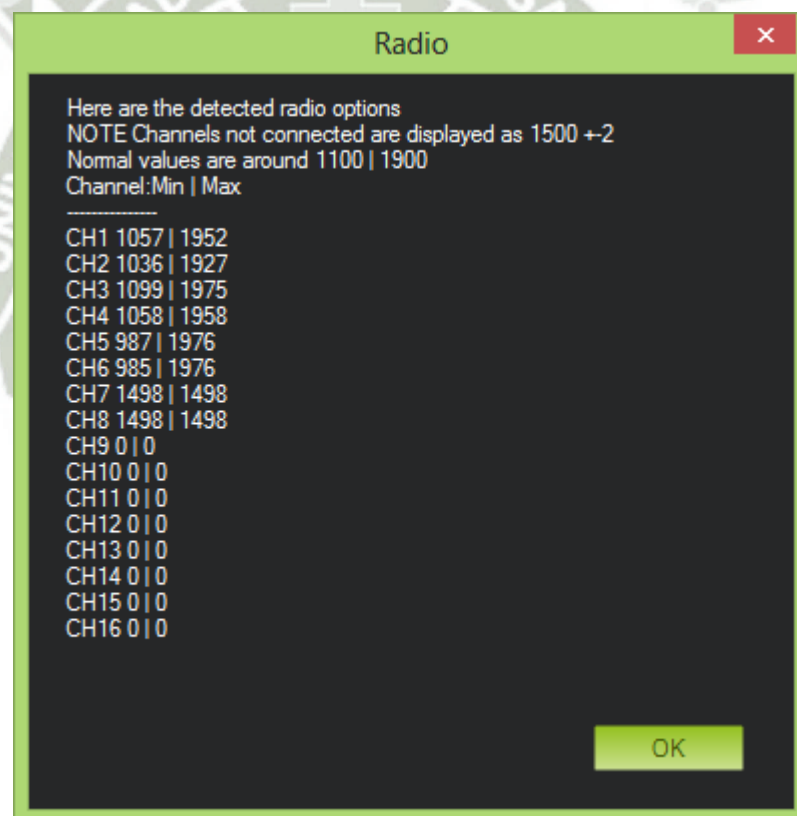


Figura5.18: Valores obtenidos después de la calibración

Fuente: Propia

5.9. CONFIGURACION DEL GEOFENCE DE ARDUPILOT

En este apartado se configurara lo que viene siendo el geofence, el cual básicamente es una malla virtual de la cual al momento de volar, nuestro equipo no podrá salir de esos límites.

La máxima altitud a la que configuramos la malla es a 100 metros, con un radio de 300 metros y con la posibilidad de configurar un modo de vuelo, el cual es el rtl(retorno a casa) el cual está configurado a una máxima altitud de 15 metros.

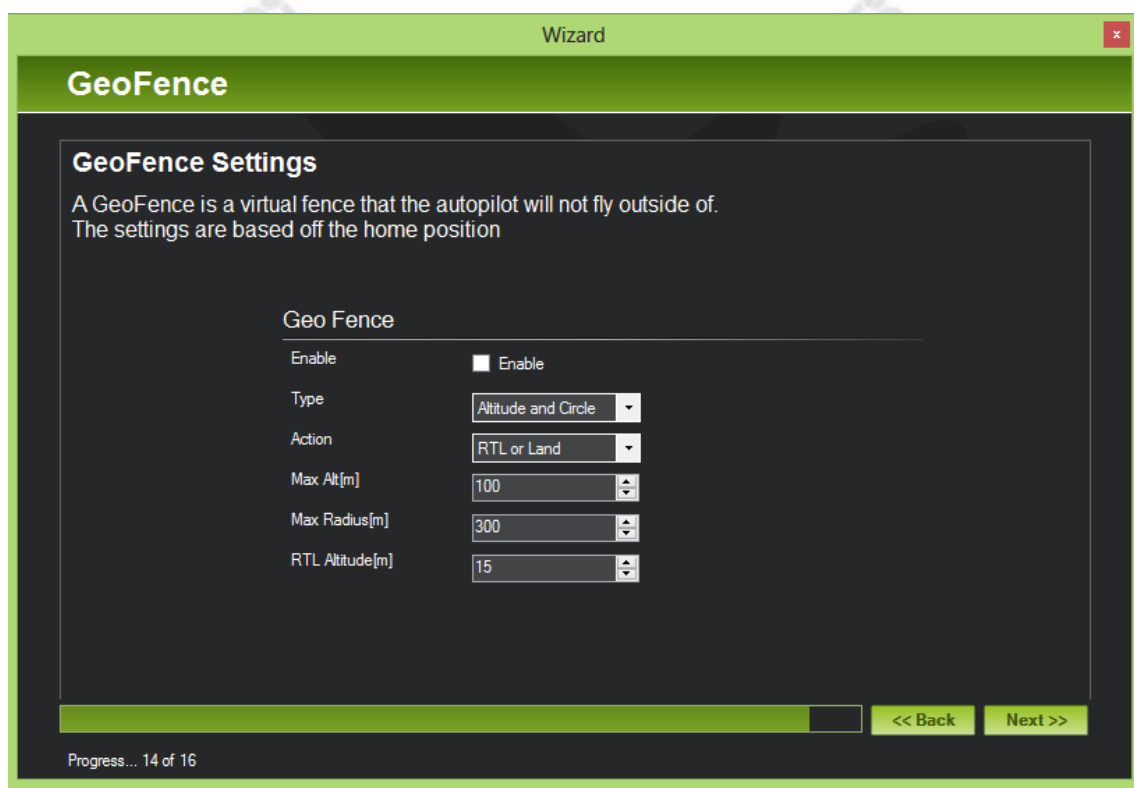


Figura5.19: Configuración del Geofence

Fuente: Propia

5.10. VERIFICACION DE CONFIGURACION DE APM 2.8

Esta es la última parte de la configuración del software de la tarjeta ardupilot 2.8, en la imagen se muestra los procedimientos que se hicieron y se desarrollaron empezando desde la verificación del GPS, la calibración del acelerómetro, calibración del compás o brújula, calibración de la emisora flysky y la prueba de pre-armado de los motores.

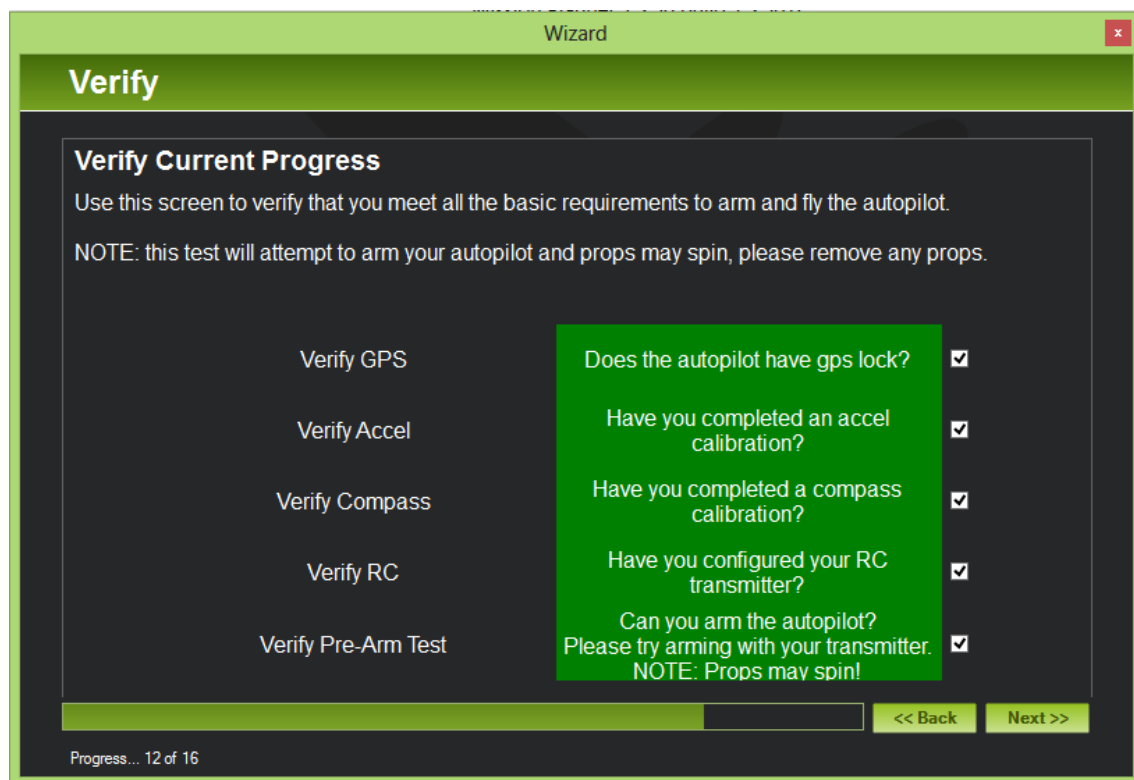
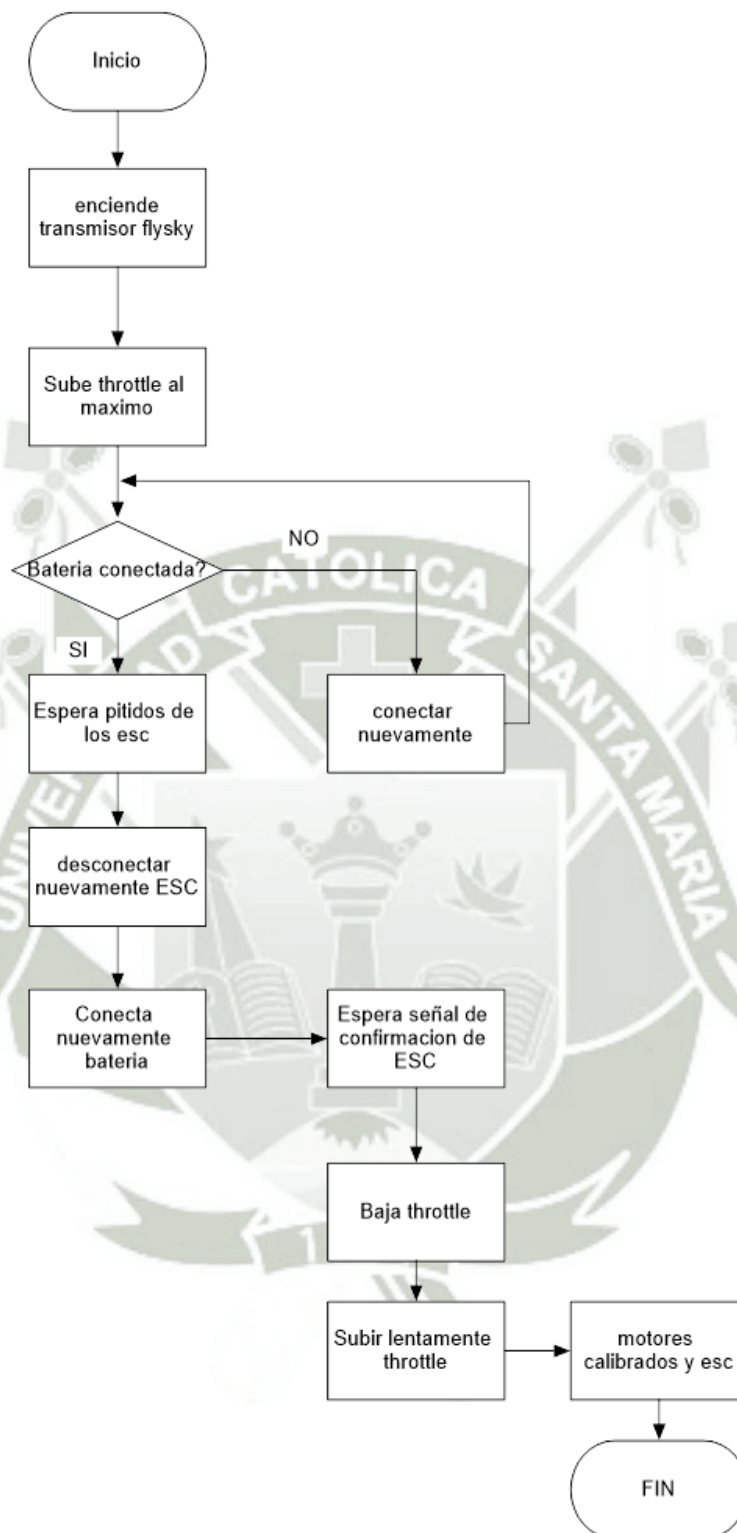


Figura5.20: Verificación de proceso de instalación culminado

Fuente: Propia

5.11. DIAGRAMA DE FLUJOS CALIBRACION DE VARIADORES ESC



Flujograma5.2: Calibración de ESC

5.12. VERIFICACION DE IMU DE APM2.8

PITCH

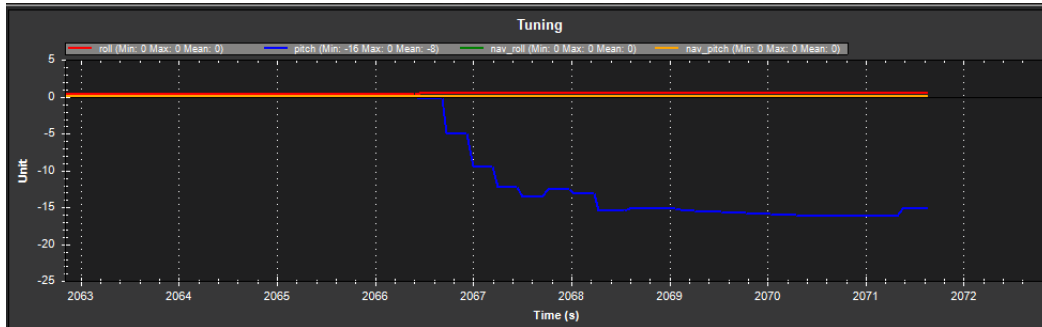


Figura5.21: Verificación de pitch

ROLL

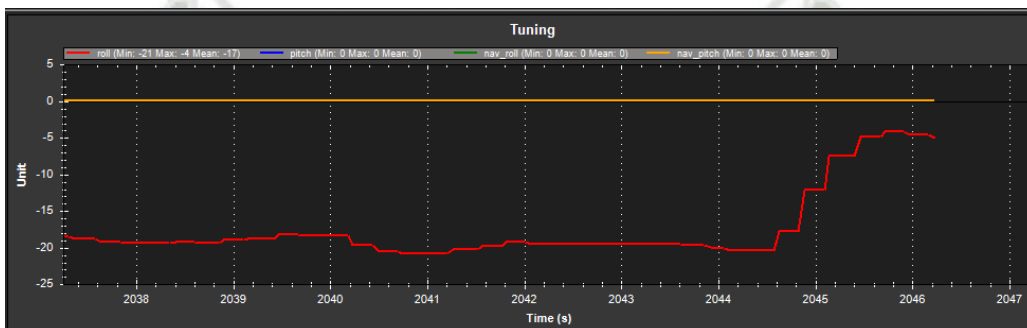


Figura5.22: Verificación de roll

PITCH AND ROLL

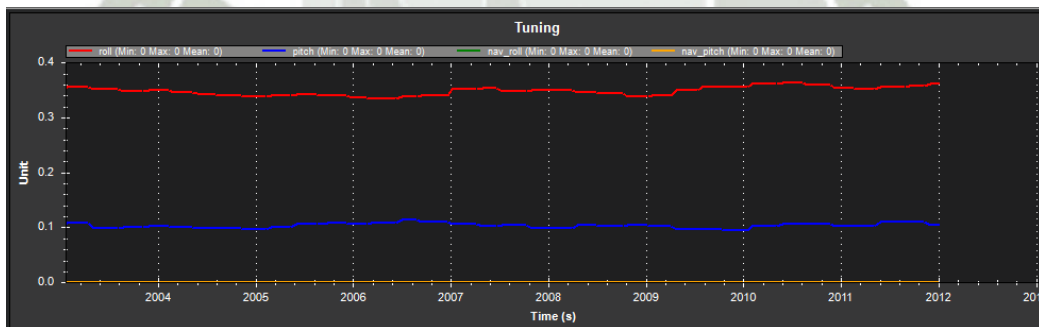


Figura5.23: tuning pitch and roll simultaneamente

5.13. CONFIGURACION DE MONITOREO DE BATERIA EN ESTACION BASE

En este punto se configura el monitoreo de batería de nuestro dron, esto se hace con la intención de que si el voltaje bajase, nuestro equipo tenga la opción de avisarnos y este pueda aterrizar en un punto cercano.

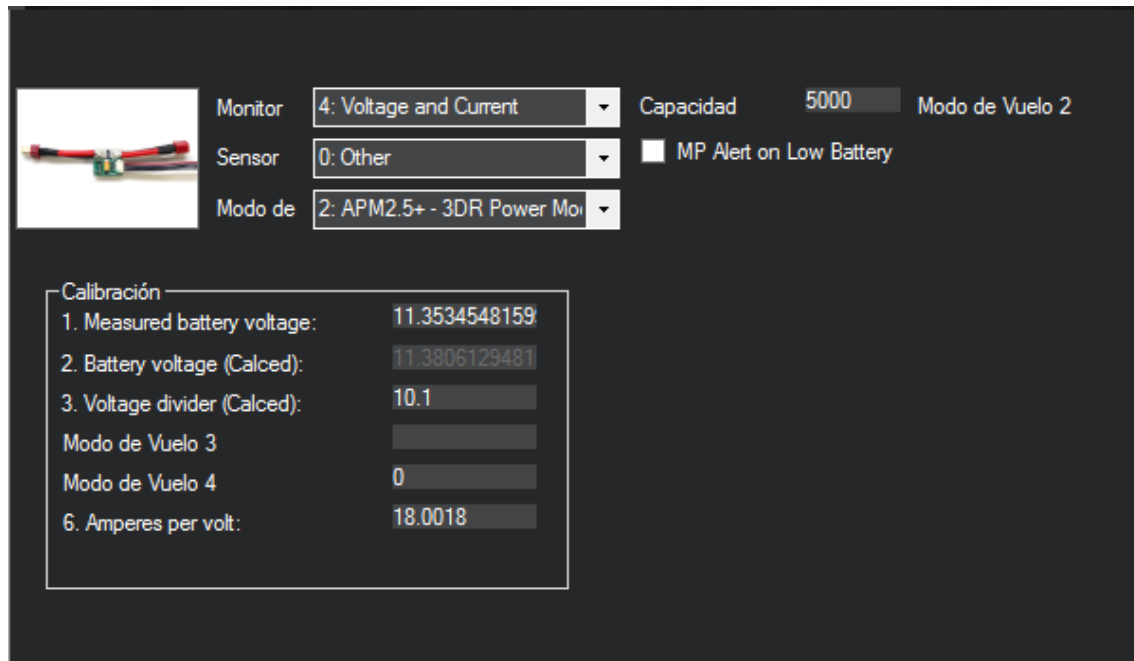


Figura5.24: Configuración de monitoreo de batería

Fuente: Propia

Seguidamente se muestra nuestro equipo ya habilitado con el sistema de posicionamiento global como se muestra en la figura.



Figura 5.25: configuración de GPS

Fuente: Propia

5.12. PID MISSION PLANNER

Los PID son el método utilizado por nuestro firmware para estabilizar el vehículo de forma continua.

Proporcional = Corrección inmediata: Entre más lejos este al valor medido del valor indicado más rápidamente será la corrección.

Integral = Con el tiempo o la corrección de estado estacionario: Si se está fallando añade corrección adicional.

Derivado = Es la corrección que va a ayudar si el caso se trata de hacerlo más lento (amortiguar) de a un poco para evitar el rebasamiento.

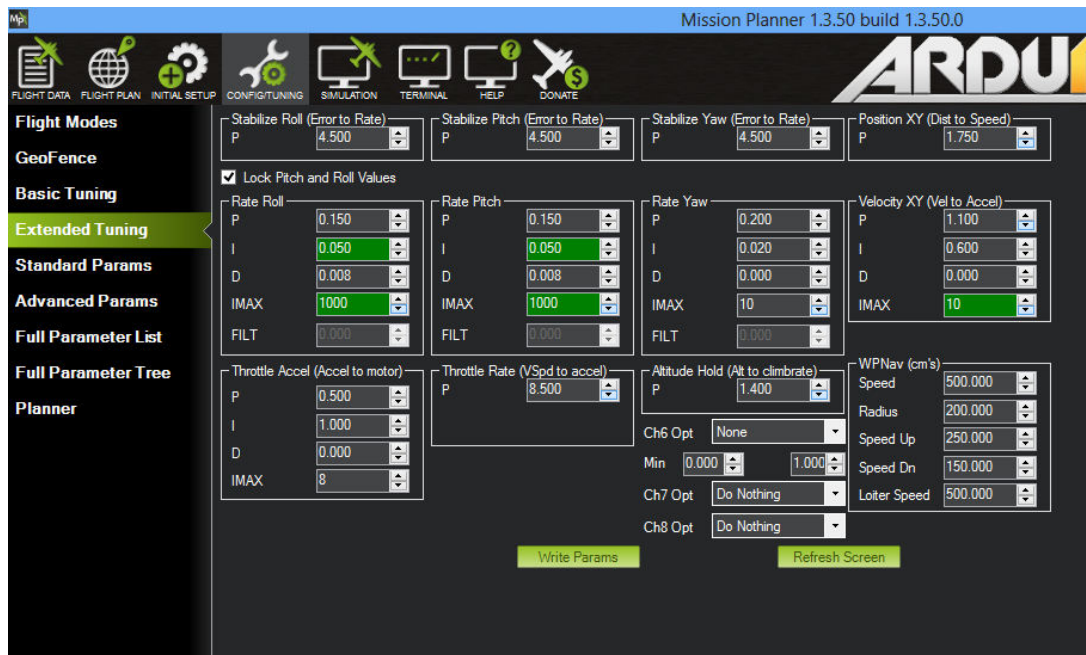


Figura5.26: Setup de PID

Fuente: <http://ardupilot.org>

El parámetro más importante para hacerlo bien es la velocidad de balanceo P (y por defecto la velocidad de cabeceo P está bloqueada en el mismo valor). Normalmente lo mejor es comenzar sintonizando la Tasa de balanceo / inclinación P en el modo Estabilizar luego pasar a la sintonización de mantenimiento de altitud en el modo Alt Hold, a continuación, Loiter (que a menudo no necesita sintonización) y, finalmente, la performance de navegación waypoint en el modo automático.



6.1. RED DE NODOS FINALIZADO

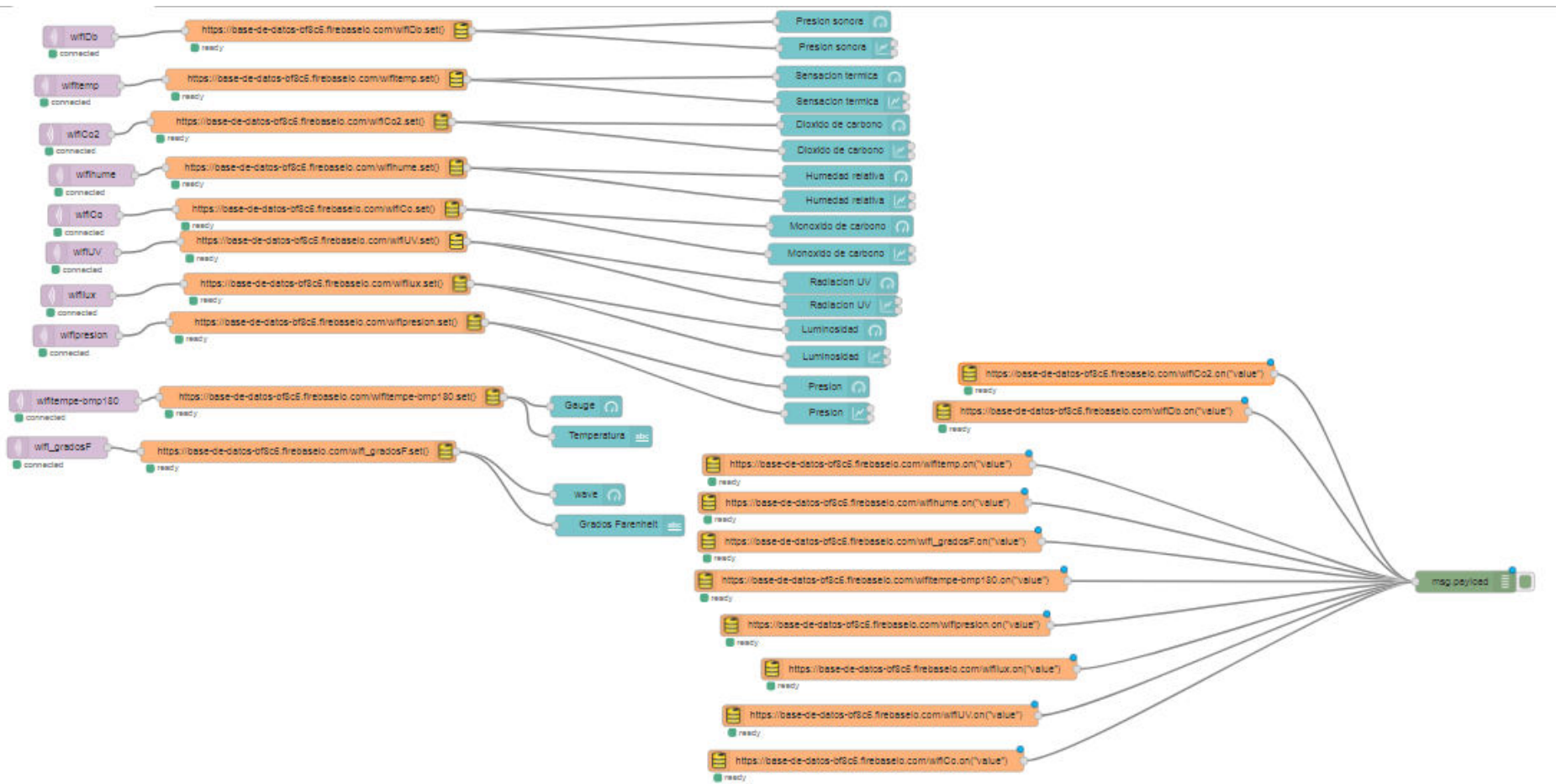


Figura6.1: Red de nodos para dashboard

Fuente: propia

6.2. SISTEMA DE MONITOREO EN NODE-RED FINALIZADO



Figura6.2: Dashboard en node red

Fuente: propia

6.3. MUESTRA DE FUNCIONAMIENTO DE BASE DE DATOS CON FIREBASE

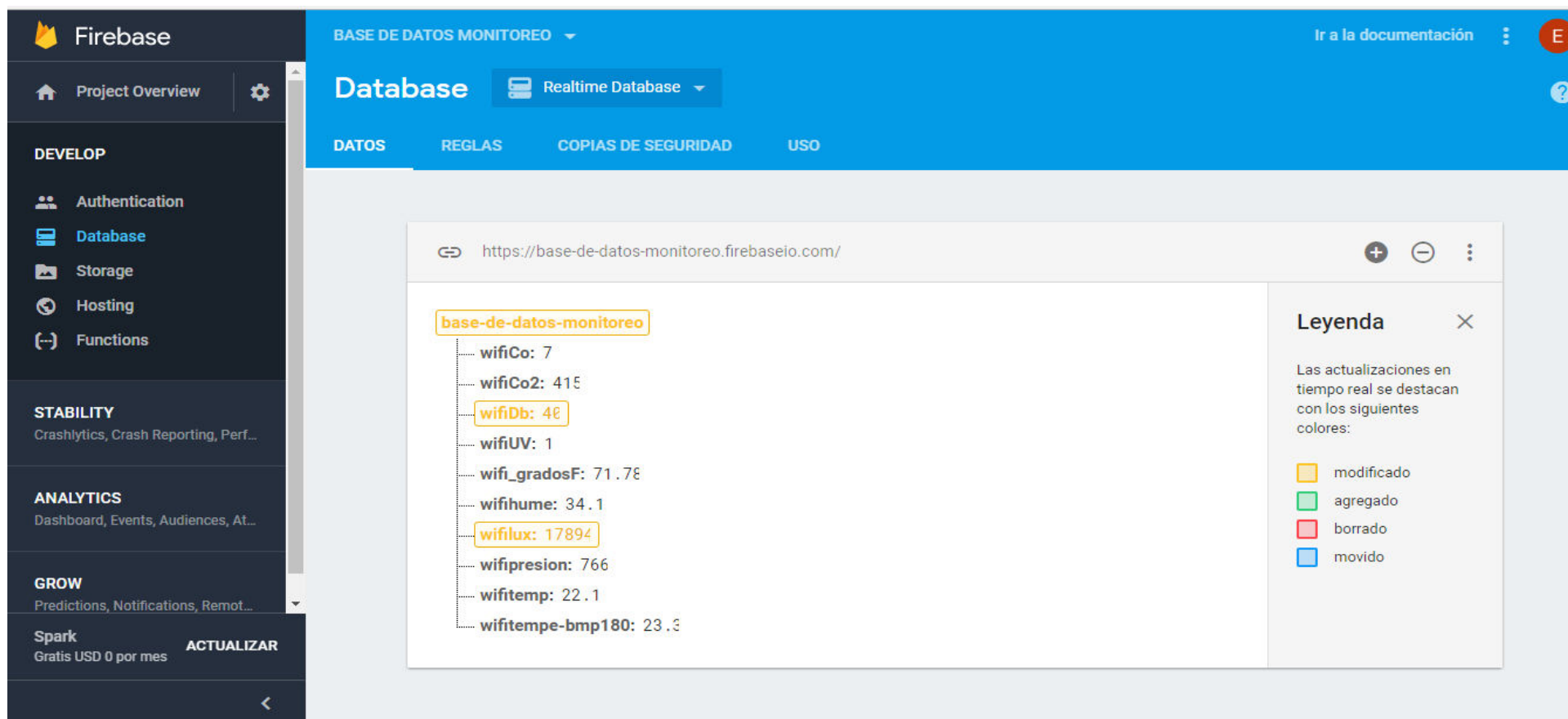


Figura6.3: base de datos elaborada en firebase

Fuente: propia

6.4. PRIMER PROTOTIPO DE VUELO



Figura 6.4: primer prototipo elaborado

Fuente: propia

Debido a su peso no se trabajó con el prototipo mostrado en la figura 6.4

6.5. PROTOTIPO DE VUELO



Figura6.5: Unidad de vuelo

Fuente: propia

6.6 UNIDAD DE MONITOREO

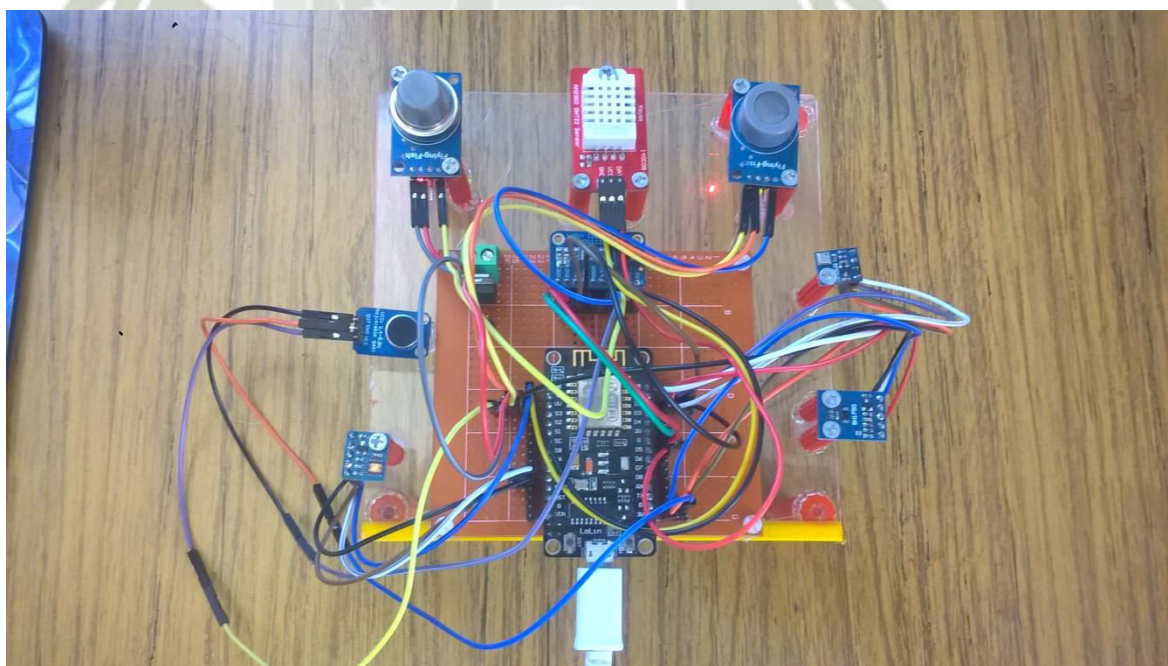


Figura6.6: Unidad de monitoreo

Fuente: propia

6.7 PRUEBAS DE VUELO



Figura6.7:Equipo listo para ser probado

Fuente: propia



Figura6.8: Despegue del equipo

Fuente: propia



Figura6.9: Estabilización del equipo

Fuente: propia





Figura6.10: Equipo estabilizado

Fuente: propia



Figura6.11: modo de vuelo: position hold

Fuente: propia



Figura6.12: Position hold

Fuente: propia



Figura6.13: equipo estabilizado

Fuente: propia



Figura6.14: Decremento en el throttle

Fuente: propia

6.8. PRUEBAS DE MONITOREO

Como se muestra en la imagen, la primera de nuestras pruebas se realizó por la clínica Arequipa en horas de la tarde.



Figura6.15: Alrededores de la clínica Arequipa

Fuente: propia

Se obtuvieron los siguientes datos:

wifiCo	wifiCo2	wifiDb	wifiUV	wifi_gradosF	wifihume	wifilux	wifipresion	wifitemp	wifitempe-bmp180
11	423	41	1.28	76.1	40.6	646	769	24.5	23.9
11	423	43	1.28	76.28	40.7	647	769	24.6	23.9
15	423	52	1.28	76.1	40.6	648	769	24.5	23.9
20	423	85	1.28	76.28	40.7	649	769	24.6	23.9
11	423	69	1.28	76.28	40.6	650	769	24.6	23.9
47	425	44	1.28	76.64	41	634	769	24.8	24.4
11	423	46	1.28	76.64	40.8	624	769	24.8	24.1
13	428	56	1.28	76.46	40.8	624	769	24.7	24.2
44	437	48	1.28	76.46	40.7	606	769	24.7	24.2
87	423	47	1.28	76.46	41	586	769	24.7	24
11	421	60	1.28	71.06	40.3	79	770	21.7	18
90	421	46	1.33	69.44	42.5	35	770	20.8	17.2
10	422	55	1.33	65.84	44.8	32	770	18.8	16.2
14	419	50	0.74	71.24	52.3	281	771	21.8	22.1
77	420	44	3.28	71.6	47.5	34054	768	22	19.3
11	421	60	1.28	71.06	40.3	79	770	21.7	18
22	420	56	4.16	69.98	47.2	54612	768	21.1	17.8
11	420	67	3.28	71.24	46.7	34015	768	21.8	18.7
33	420	45	3.28	71.6	47.8	34126	768	22	19.7

Tabla6.1: Datos obtenidos en la clínica Arequipa

Fuente: propia

Se tomaron datos en una parte de la avenida ejército, en horas de la mañana en la que se venían ejecutando obras de cambio de tuberías de desagüe y agua.



Figura6.16: Alrededores de la Avenida Ejercito

Fuente: propia

Se obtuvieron los siguientes datos:

wifiCo	wifiCo2	wifiDb	wifiUV	wifi_gradosF	wifihume	wifilux	wifipresion	wifitemp	wifitempe-bmp180
12	431	43	0.79	78.98	55.8	9	770	21.5	25.6
9	422	39	7.29	85.82	36.3	54612	766	29.9	29.2
14	425	47	0.83	78.8	40	352	769	26	24.5
10	423	42	7.84	78.8	34.8	54612	764	26	21.3
10	422	40	7.87	78.8	34.8	54612	764	26	21.1
10	422	52	7.87	78.26	35	54612	764	25.7	21.5
11	423	46	15.35	77.54	36.1	54612	763	25.3	20.6
12	424	46	15.07	76.64	35.5	54612	763	24.8	19.9
11	422	51	10.1	79.16	35.6	54612	764	26.2	20.9
10	415	61	12.48	78.44	35	54612	764	25.8	21.1
12	424	48	14.92	76.82	35.4	54612	763	24.9	20.7
20	438	57	14.83	77.9	36	54612	763	25.5	21.1
21	433	62	13.3	74.48	35.4	54612	764	23.6	19.8
38	440	80	13.01	74.3	35.5	54612	764	23.5	20
87	418	85	14.78	76.82	35	54612	763	24.9	20.6
45	450	52	7.84	71.24	38.6	54612	767	21.8	19.5
77	430	55	10.02	79.52	35.3	54612	764	26.4	20.8
10	423	39	7.84	71.24	38.6	54612	767	21.8	19.5
9	422	40	7.32	85.1	37.2	54612	766	29.5	29.6

Tabla6.2: Datos obtenidos por la avenida ejercito

Fuente: propia

Se realizó la adquisición de datos en una parte del puente Grau en horas de la tarde

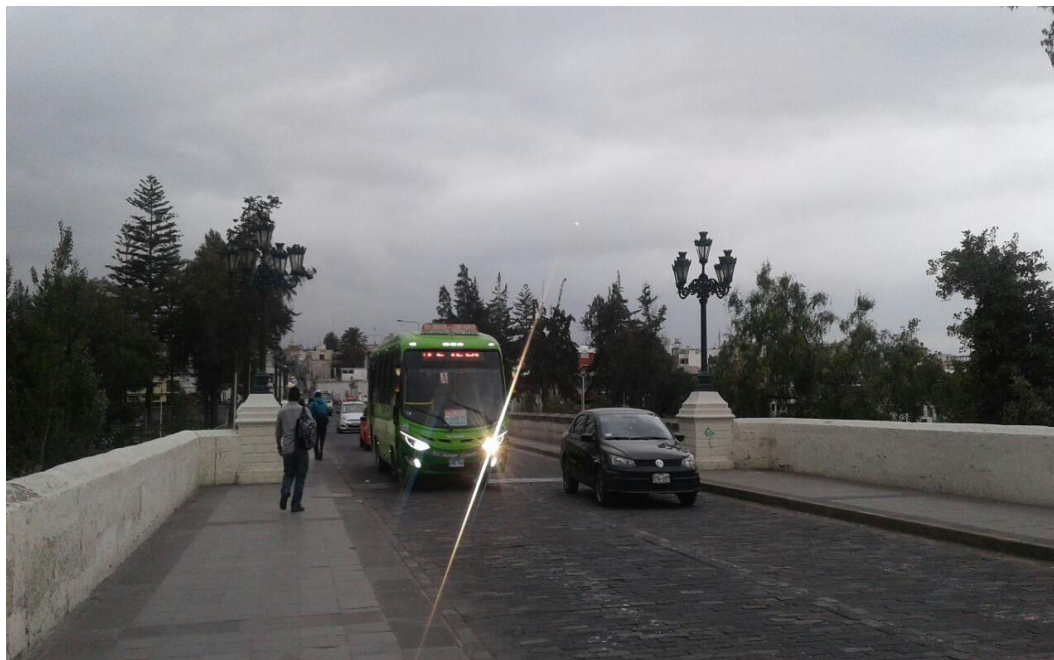


Figura6.17: Puente Grau

Fuente: propia

wifiCo	wifiCo2	wifiDb	wifiUV	wifi_gradosF	wifihume	wifilux	wifipresion	wifitemp	wifitempe-bmp180
15	419	44	4.99	69.98	47.1	54612	766	21.1	17.9
21	420	40	3.28	71.06	46.6	34348	768	21.7	18.5
35	420	42	3.49	70.88	46.6	36712	768	21.6	17.6
46	420	71	3.28	71.24	46.6	34088	768	21.8	18.6
28	419	41	3.47	70.7	46.5	36132	768	21.5	17.6
15	419	69	3.52	70.52	46.6	38067	768	21.4	17.5
23	420	61	3.55	70.16	46.5	38834	768	21.2	17.7
10	419	44	4.99	69.98	47.1	54612	766	21.1	17.9
9	420	43	6.64	74.3	43.2	54612	764	23.5	20.2
13	420	40	0.77	71.24	51.4	281	771	21.8	22.1
50	435	61	6.72	75.2	42.1	54612	764	24	20.3
12	440	51	6.75	75.74	41.5	44612	764	24.3	21.1
37	418	82	6.78	76.46	41.6	54612	764	24.7	21.3
12	410	85	6.78	76.82	41.6	54612	764	24.9	21.5
12	419	51	6.75	77.54	40.2	54612	764	25.3	21.4
24	421	42	6.72	77.54	39.7	54612	764	25.3	20.8
30	420	47	6.7	78.08	39.7	54612	764	25.6	21.2
12	419	66	6.64	78.62	39.8	32117	764	25.9	20.9
12	420	60	6.64	77.36	39.1	54410	764	25.2	20.7

Tabla 6.3: Datos obtenidos por el puente Grau

Fuente: propia

*Se realizo la toma de datos por los alrededores del mercado El Avelino y se obtvieron los siguientes datos.



Figura6.18: Alrededores del mercado el Avelino

Fuente: propia



Figura6.19: Mercado el Avelino

Fuente: propia

wifiCo	wifiCo2	wifiDb	wifiUV	wifi_gradosF	wifihume	wifilux	wifipresion	wifitemp	wifitempe-bmp180
10	421	55	6.5	85.1	34.2	46670	768	29.5	25.4
11	421	41	5.78	84.02	34	40438	768	28.9	25.2
10	421	56	5.39	83.48	34.4	37079	768	28.6	24.6
10	422	76	5.27	82.58	34.7	35078	768	28.1	24.2
10	422	45	5.03	81.68	35.8	32873	768	27.6	23.5
11	491	56	5.19	80.96	36.4	33191	769	27.2	23.9
174	439	63	5.7	78.62	36.7	38751	768	25.9	22.5
73	428	95	5.94	78.8	37.1	40420	768	26	22.7
28	441	42	9.38	78.62	37.9	54612	768	25.9	22.8
13	422	61	4.63	84.56	35.7	29220	769	29.2	27
13	434	46	3.83	82.22	36.2	21594	768	27.9	24.6
101	425	52	3.79	81.32	37	21463	769	27.4	23.9
14	422	78	3.91	76.28	39.9	22907	768	24.6	21.8
72	422	49	5.98	75.02	41.2	41031	768	23.9	21.8
13	422	51	4.23	75.56	40.9	25615	768	24.2	21.7
17	424	71	1.07	73.4	47.8	680	771	23	22.6
230	426	63	3.79	81.5	36.8	20891	769	27.5	24.6
83	422	80	3.95	76.1	40	23198	769	24.5	21.7
51	422	53	4.63	75.2	41.1	28059	768	24	21.5

Tabla6.4: Datos obtenidos por el Mercado el Avelino

Fuente: propia



*Se realizaron pruebas por alrededores del seguro social



Figura6.20: Seguro social

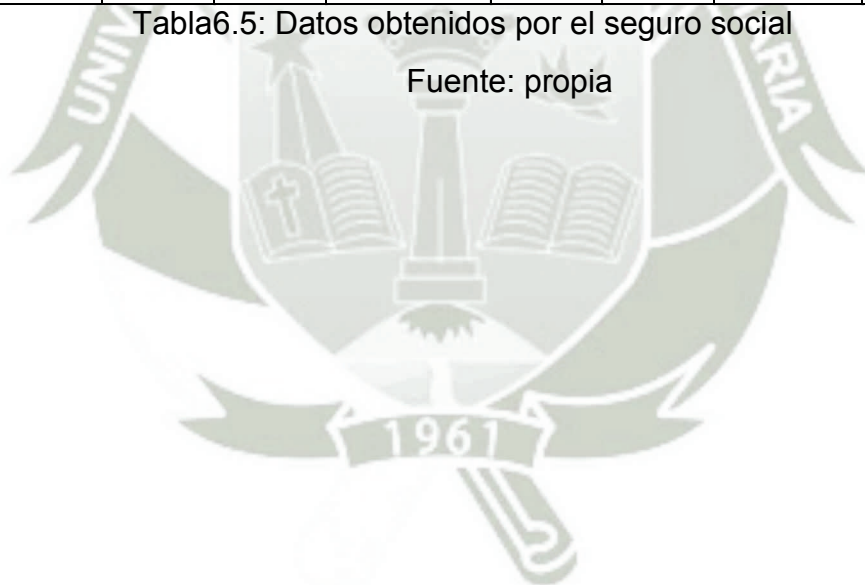
Fuente: propia

Tenemos los siguientes resultados, estos fueron obtenidos en horas de la mañana.

wifiCo	wifiCo2	wifiDb	wifiUV	wifi_gradosF	wifihume	wifilux	wifipresion	wifitemp	wifitempe-bmp180
17	424	43	1.11	73.4	47.4	638	771	23	22.6
11	424	41	1.07	73.4	47.4	624	770	23	22.6
17	424	44	1.11	73.4	47.4	603	771	23	22.6
22	424	73	1.07	73.4	47.5	570	771	23	22.6
17	424	71	1.07	73.4	47.5	544	770	23	22.6
17	424	55	1.07	73.4	47.5	536	770	23	22.6
17	424	61	1.07	73.4	47.4	533	770	23	22.6
38	420	43	6	79.7	37.3	54612	766	26.5	22.9
26	420	85	6.22	82.22	37.5	54612	765	27.9	23.6
10	422	48	13.77	85.82	34	54612	764	29.9	24.9
11	422	50	13.66	91.22	34	54612	765	32.9	28.2
125	428	40	1.07	86.72	55.2	373	771	31.9	32.3
55	481	48	1.03	101.12	53.2	353	771	38.4	30.3
23	422	46	1.07	81.5	39	455	771	27.5	25
17	423	44	1.03	83.66	39.4	488	771	28.7	25.2
10	418	60	10.86	71.78	42.3	54612	764	22.1	19
24	424	45	1.03	84.2	41.1	481	771	29	25.4
23	422	46	1.07	81.5	39	455	771	27.5	25
14	423	64	12.74	91.04	34.6	54612	765	32.8	25.1

Tabla6.5: Datos obtenidos por el seguro social

Fuente: propia



Se hicieron capturas de pantalla, en las que se muestra los cambios que se produjeron al realizar las mediciones, tanto en el dashboard que se elaboró como también en nuestra base de datos

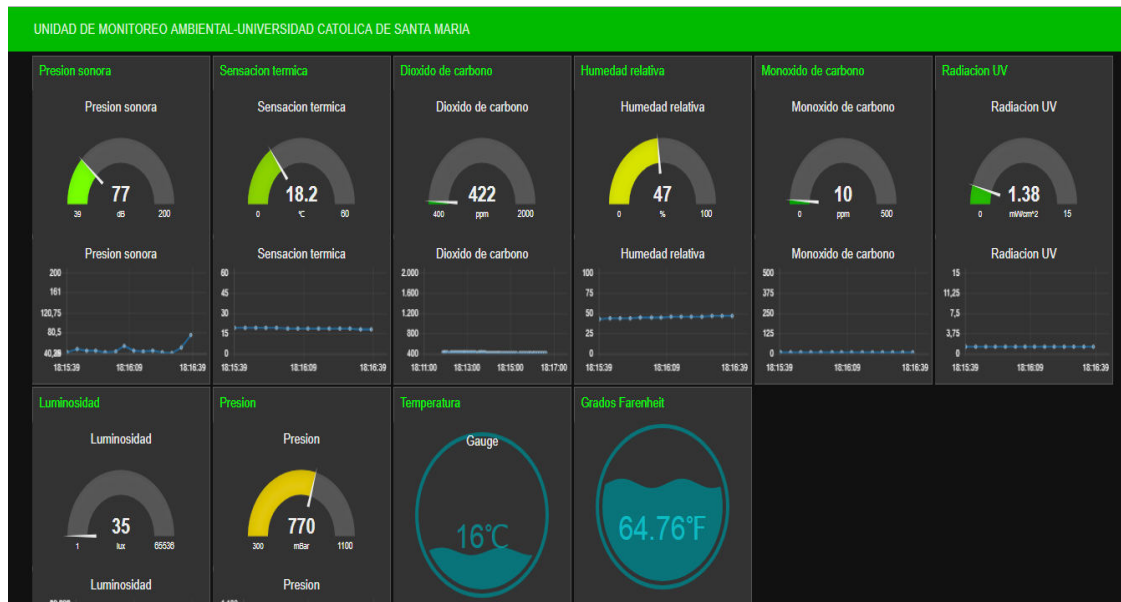


Figura6.21: Cambios en el dashboard

Fuente: propia

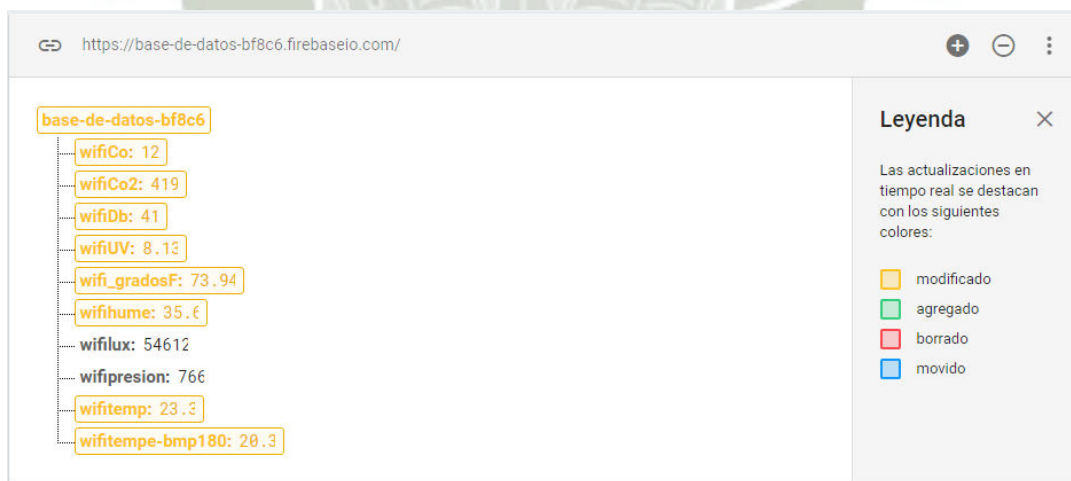


Figura6.22: Base de datos con valores modificados

Fuente: propia



Figura6.23: Variación de datos en Dashboard

Fuente: propia

NOTA: EN EL CD DE TESIS SE ADJUNTAN LOS DATOS ALMACENADOS POR NUESTRA BASE DE DATOS EN FORMATO JSON, ASI COMO TAMBIEN LOS VALORES GENERADOS POR EL SOFTWARE PROP CALC DADO A QUE SE HICIERON PRUEBAS CON HELICES 10X4.5" Y 9X4.5".



CONCLUSIONES

- PRIMERO:** El presente trabajo tenía como objetivo el diseño y construcción de un prototipo de cuadricoptero con la capacidad de llevar un sistema de monitoreo. El presente objetivo llegó a buen fin, sin embargo, debido a diferentes problemáticas nos vimos en la necesidad de hacer un cambio en la estructura.
- SEGUNDO:** Se decidió por trabajar con la estructura del dron F450 de DJI, ya que terminó siendo una estructura muy liviana y compacta.
- TERCERO:** Se optó por el uso de una tarjeta de vuelo comercial, resultando en la culminación del vehículo dada la facilidad y versatilidad que ofrecen tanto la tarjeta que usamos, como las tarjetas existentes en el mercado.
- CUARTO:** El uso de este equipo combinado con el sistema de monitoreo que se construyó nos permitirá conocer cuáles son los puntos existentes en la ciudad de Arequipa con un mayor índice de contaminación acústica, contaminación del aire y el incremento de parámetros ambientales debido a diversos factores.
- QUINTO:** Se implementó un sistema de monitoreo, basado en el internet de las cosas.
- SEXTO:** Optamos por trabajar con node-red debido a que es una herramienta de programación visual, con una curva de aprendizaje muy alta, siendo muy intuitiva a la hora de programación y al igual que arduino posee muy activa en cuanto a actualizaciones y plugins se refiere.
- SEPTIMO:** Se trabajó con la tarjeta NodeMCU dada a su fácil conectividad a redes wifi, su tamaño también fue favorable dado que no ocupa mucho espacio.
- OCTAVO:** Uno de los principales problemas al trabajar con esta tarjeta fue que solo poseía un puerto analógico, debido a eso usamos un módulo expensor analógico de 16 bits.

NOVENO: Trabajamos con el protocolo MQTT dada su ligereza, requiere poco ancho de banda y su consumo es realmente bajo así como también precisa de pocos recursos para su funcionamiento, viene siendo muy empleado en la comunicación de sensores y, consecuentemente en el internet de las cosas



RECOMENDACIONES

1. Se puede integrar un sistema de vuelo con geo-localización para programar rutas de vuelo para nuestro sistema y probar otros modos de vuelo que integra la tarjeta ardupilot apm 2.8.
2. Implementar y hacer uso de técnicas de manufactura modernas mediante el uso de materiales compuestos como fibra de carbono o el empleo de impresiones 3D para poder desarrollar un modelo más óptimo y más liviano a la hora de vuelo.
3. En cuanto a la medición de temperatura y otros parámetros descritos anteriormente, se puede hacer de uso de sensores que posean un tiempo de respuesta mucho más rápido, mejor resolución y una mayor precisión a la hora de la medición.
4. Se recomienda trabajar con baterías de mayor capacidad, ya que de acuerdo a los cálculos obtenidos el tiempo de vuelo es aproximadamente siete minutos, se tiene que tomar en cuenta el rendimiento de los motores o buscar mejores alternativas que se encuentren en el mercado.
5. Para poder realizar pruebas en determinadas zonas es preferible contar con alguien experimentado en el manejo de estos equipos, todas las pruebas que se realizaron fueron en espacios controlados.
6. Se puede elaborar una red de sensores ubicados en determinadas zonas de la ciudad, con un sistema de almacenamiento individual, estos equipos podrán ser transportados por el cuadricoptero.
7. Dado que actualmente la comunidad open source crece exponencialmente, podremos trabajar con otras tarjetas que tengan un mayor alcance a la hora de adquirir datos y poder conectarse a internet.
8. Poder trabajar con otras aplicaciones orientadas al internet de las cosas, y bases de datos más robustas.
9. Trabajar con transmisores de mayor alcance y mayor capacidad como son los transmisores FUTABA o TURNIGY.

BIBLIOGRAFIA

- ArduPilot Mega 2 Quadcopter Basic Stabilize Mode Flying & PIDs Obtained from:
http://download.ardupilot.org/downloads/wiki/pdf_guides/APM2_Quad_Stabilize_1.pdf
- Bawek Dean(2012)Design and implementation of a control system for a quadrotor MAV.Department of Aerospace Engineering.University of Maryland.
- Costa de Oliveira Marcelo de Lellis. (2011)Prague.Modeling,Identification and control of a Quadrotor Aircraft.Department of Control Engineering.Faculty of Electrical Engineering. Czech Technical University in Prague.
- Estandares de Calidad Ambiental (ECA) para aire.Obtenido de:
<http://sinia.minam.gob.pe/normas/aprueban-estandares-calidad-ambiental-eca-aire-establecen-disposiciones-1>
- Ferreras Astorqui, Ignacio María. (2016)Madrid. SENSOR IoT PARA MONITORIZACION DE CONSUMO DE ENERGIA EN CONTINUA. Facultad de Informatica.Universidad Complutense de Madrid.
- Gel'fand, Israel (1963). Calculus of Variations
- Gonzalez Daza Enrique (2014/2015) Valencia-España. Red de sensores-Internet de las cosas. Escola Tècnica Superior d'Enginyeria Informàtica. Universitat Politècnica de València
- Guia para la elaboracion de planes de accion para la prevencion y control de ruido urbano.Obtenido de: <http://sinia.minam.gob.pe/documentos/gua-elaboracin-planes-accin-prevencin-control-ruido-urbano>
- <http://ardupilot.org/copter/docs/common-apm25-and-26-overview.html>
- <https://www.arduino.cc/>
- Informe: Registro de emisiones y transferencia de contaminantes-RETC Peru.Obtenido de: <http://sinia.minam.gob.pe/documentos/registro-emisiones-transferencia-contaminantes-retc-peru>

- Install and Setup the ArduPilot Mega 2 in a Quadcopter. Obtained from: http://download.ardupilot.org/downloads/wiki/pdf_guides/APM2_Quad_Setup_3.pdf
- Jirapatnakul C. Artit. (2005) A multi-sensor embedded microcontroller system for condition monitoring of rc helicopters. Department of Electrical Engineering. Pennsylvania State University.
- KK2.1 & KK2.1.5 Instruction Manual. Obtained from: <https://hobbyking.com/media/file/330522838X248012X8.pdf>
- Límites máximos permisibles de emisiones contaminantes para vehículos automotores que circulan en la red vial (Decreto Supremo). Obtenido de: <http://sinia.minam.gob.pe/documentos/lmites-mximos-permisibles-emisiones-contaminantes-vehiculos>
- López de Paz Renzo Gustavo. (2016) Lima. Diseño del sistema de control de temperatura de un invernadero. Pontificia Universidad Católica del Perú
- Mak Christopher. (2015) Urbana, Illinois. Modeling and simulation of comercial aircraft electrical systems. University of Illinois at Urbana-Champaign.
- Manual de usuario y especificaciones técnicas para Inbeebox modelo Inbeebox33 Versión 1.0 y plataforma node-RED. Obtenido de www.amnetpro.com
- Nadales Real Christian. (2009) Catalunya. Control de un Quadrotor mediante la plataforma Arduino. Escola Politècnica Superior de Castelldefels. Universitat Politècnica de Catalunya.
- Parada Pardo Eduardo. (2012) Leganes-Madrid. Quadcopter: Construcción, control de vuelo y navegación GPS. Universidad Carlos III de Madrid.
- Protocolo Nacional de Monitoreo de Ruido Ambiental. Obtenido de: <http://sinia.minam.gob.pe/documentos/protocolo-nacional-monitoreo-ruido-ambiental>
- Rabanal Carretero Danna Zuleyka. (2011) Lima. Integración de un sistema UAV con control autónomo en un equipo aéreo para agricultura de precisión. Pontificia Universidad Católica del Perú.

- Reed Siefert Christiansen. (2004). Design of an autopilot for small unmaned aerial vehicles. Department of Electrical and Computer Engineering. Brigham Young University.
- Reyna Huaman Claudia Evelia. (2015) Lima. SISTEMA AUTOMATIZADO PARA EL MONITOREO Y CONTROL DE HUMEDAD EN UN INVERNADERO. Pontificia Universidad Católica del Perú.
- Romero Serna Edward Dagoberto. (2015) DESARROLLO DE UN PROTOTIPO PARA EL GEOPOSICIONAMIENTO REFERENCIADO POR GPS DE UN AEROMODELO TIPO QUADCOPTERO. Universidad Pedagógica y Tecnológica de Colombia.
- Salcedo Torres. (2014) Lima. DISEÑO DE UN SISTEMA AUTOMATIZADO PARA RIEGO POR GOTEJO PARA PALTA HASS. Pontificia Universidad Católica del Perú.
- Schmidt, Michael David (2011). SIMULATION AND CONTROL OF A QUADROTOR UNMANNED AERIAL VEHICLE. University of Kentucky.
- Setting up a DJI Flamewheel F450 Quadcopter with the APM2. Obtained from:
http://download.ardupilot.org/downloads/wiki/pdf_guides/APM2_Flamewheel_Quad_Setup1.pdf
- Tabuchi Fukuhara Rubén Toshiharu. (2015) Lima. Diseño de un Vehículo Aéreo no Tripulado de cuatro rotores para una carga útil de 1kg. Pontificia Universidad Católica del Perú.



ANEXO A

**CODIGO DE MONITOREO DE PARAMETROS FISICOS AMBIENTALES Y CONTAMINACION
ACUSTICA**

```

////////////////////////////////////
////////////////////////////////////
// PROGRAMA ELABORADO PARA LA MEDICION DE PARAMETROS AMBIENTALES Y
// CONTAMINACION ACUSTICA
////////////////////////////////////
////////////////////////////////////

//-----
//
// LIBRERIAS
//-----

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_BMP085.h>
#include <Wire.h>
#include <BH1750.h>
#include <Adafruit_ADS1015.h>

//-----
// CONFIGURACION DE LA RED WIFI A CONECTARSE
//-----

const char* ssid = "-----";
const char* password = "-----";
const char* mqtt_server = "-----";
//-----

// CREACION DEL CLIENTE MQTT
//-----

WiFiClient espClient;
PubSubClient client(espClient);
//-----

// CREACION DE VARIABLES GLOBALES
//-----

#define DHTPIN 14
#define DHTTYPE DHT22
#define coZero 50 //nivel de co calibrado
#define co2Zero 100 //nivel de co2 calibrado
DHT dht(DHTPIN, DHTTYPE);
Adafruit_BMP085 bmp;
BH1750 lightMeter;
int UVsensor = A0; //salida del sensor
const int sampleWindow = 50; //Ancho de ventana de muestreo en ms (50 ms
= 20Hz)
Adafruit_ADS1115 ads;
//-----

// FUNCION DE CONFIGURACION DE WIFI

```

```
//-----  
-----  
void setup_wifi() {  
  
    delay(10);  
    Serial.println();  
    Serial.print("Conectando a ");  
    Serial.println(ssid);  
  
    WiFi.begin(ssid, password);  
  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
  
    randomSeed(micros());  
  
    Serial.println("");  
    Serial.println("WiFi connected");  
    Serial.println("IP address: ");  
    Serial.println(WiFi.localIP());  
}  
//-----end setup wifi-----  
  
//-----  
//                               FUNCION RECONNECT  
//-----  
  
void reconnect() {  
    // Loop esperando reconexion  
    while (!client.connected()) {  
        Serial.print("Intentando conexión con MQTT...");  
        // Crea un ID de cliente aleatoria  
        String clientId = "ESP8266Client-";  
        clientId += String(random(0xffff), HEX);  
        // Intento de conexion  
        if (client.connect(clientId.c_str())) {  
            Serial.println("connected");  
            // Una vez conectado publica el siguiente anuncio  
            client.publish("outTopic", "hello world");  
            // ... y vuelve a suscribir  
            client.subscribe("inTopic");  
        } else {  
            Serial.print("fallo, rc=");  
            Serial.print(client.state());  
            Serial.println(" Prueba otra vez después de 5 segundos");  
            // Espera 5 segundos y vuelve a intentar  
            delay(5000);  
        }  
    }  
}  
//-----end reconnect-----  
-----  
  
//-----  
//                               SETUP  
//-----  
-----  
void setup()
```

```

{
  Serial.begin(115200); //inicia transmision serial
  setup_wifi(); //inicializa wifi
  client.setServer(mqtt_server, 1883); //inicializa servidor mqtt
  dht.begin(); //inicializa sensor dht22
  Wire.begin(); //inicia i2c
  bmp.begin(); //inicia sensor de presion
  lightMeter.begin(); //inicia sensor de luminosidad
  ads.setGain(GAIN_ONE); // +/- 4.096V 1 bit = 0.125mV
  ads.begin(); //inicia modulo expansor ads1115
  pinMode(UVsensor, INPUT);
}
//-----end setup-----
//-----
//
//                                LOOP INFINITO
//-----
void loop(){
  if (!client.connected()) //pregunta si se conecto con mqtt
  {
    reconnect();
  }

//-----DHT22-----
float h = dht.readHumidity(); //hace lectura de humedad relativa
float t = dht.readTemperature(); //lectura de temperatura en grados
centigrados
float f = dht.readTemperature(true); //
//-----
//-----BMP180-----
float tempe = bmp.readTemperature(); //lee la temperatura del sensor de
presion
double presion = (bmp.readPressure())/100; //realiza la lectura de
presion barometrica en hPa
//-----
uint16_t lux = lightMeter.readLightLevel(); //lectura de datos del sensor
bh1750
//-----
//-----UV-----
int nivel_UV = averageAnalogRead(UVsensor);

float voltaje_out = 3.3 * nivel_UV/1024;
float intensidad_UV = mapfloat(voltaje_out, 0.99, 2.8, 0.0, 15.0);

//-----
//-----DECIBELIOS-----
int16_t adc2;
unsigned long startMillis= millis(); //empieza ventana de muestreo
unsigned int peakToPeak = 0; // nivel pico-pico
unsigned int signalMax = 0;

```

```
unsigned int signalMin = 65536;

// recoleta datos durante 50 mS
while (millis() - startMillis < sampleWindow)
{
  adc2 = ads.readADC_SingleEnded(2);
  if (adc2 < 65536) // elimina muestras erróneas
  {
    if (adc2 > signalMax)
    {
      signalMax = adc2; // salva solo los niveles máximos
    }
    else if (adc2 < signalMin)
    {
      signalMin = adc2; // salva solo los mínimos niveles
    }
  }
}
peakToPeak = (signalMax - signalMin)/10;//Max - min = amplitud pico-pico
float db = map( peakToPeak,20,900,40.50,130.50);
//-----
//-----CO-----
//-----
int16_t adc0[5];
int co_x = 0; //int para valor sin procesar de co
int co_y = 0; //int para valor compensado co
int coppm = 0; //int para ppm calculado
int prom = 0; //int for promedio
for (int x = 0 ; x<5 ; x++)
{
  adc0[x] = ads.readADC_SingleEnded(0);
  delay(200);
}
for (int x = 0 ; x<5 ; x++){ //sumamos las muestras
//obtenidas con el promedio y alacena en promedio
  prom = prom + adc0[x];
}
co_x = prom/5; //dividimos las muestras entre 5
co_y = co_x - coZero; //obtenemos un valor ponderado
coppm = map(co_y,0,65536,0,500); //hace un mapeo de valores de co en
la atmosfera
//-----
//-----CO2-----
//-----
int16_t adc1[10]; //int array para las lecturas de co2
int co2raw = 0; //int para valor sin procesar de co2
int co2comp = 0; //int para valor compensado de co2
int co2ppm = 0; //int para ppm calculado
int zzz = 0; //int for promedio
for (int x = 0 ; x<10 ; x++)
{
  adc1[x] = ads.readADC_SingleEnded(1);
  delay(200);
}
for (int x = 0 ; x<10 ; x++){ //sumamos las muestras
//obtenidas con el promedio y alacena en promedio
```

```

    zzz=zzz + adc1[x];

}
co2raw = zzz/10;          //dividimos las muestras entre 10
co2comp = co2raw - co2Zero; //obtenemos un valor ponderado
co2ppm = map(co2comp,0,65536,400,2000); //hace un mapeo de valores de co2
en la atmosfera
//-----
//-----IMPRIME VALORES DE TEMPERATURA Y HUMEDAD-----
//-----
Serial.print("Humedad: ");
Serial.print(h);
Serial.println(" %");
Serial.print("Temperatura: ");
Serial.print(t);
Serial.println(" *C ");
Serial.print("Temp_GradosF: ");
Serial.print(f);
Serial.println(" *F ");
//-----
//-----IMPRIME VALORES DE PRESION, ALTITUD Y TEMPERATURA-----
//-----
Serial.print("Temperatura.BMP180 = ");
Serial.print(tempe);
Serial.println(" *C");
Serial.print("Presión = ");
Serial.print(presion);
Serial.println(" mBar");
//-----
//-----IMPRIME VALORES DE LUMINOSIDAD-----
//-----
Serial.print("Luminosidad: ");
Serial.print(lux);
Serial.println(" lx");
//-----
//-----IMPRIME INDICE DE RADIACION UV-----
//-----
Serial.print(" intensidad uv: ");
Serial.print(intensidad_UV);
Serial.println(" mW/cm^2");
//-----
//-----IMPRIME VALORES DE MONOXIDO DE CARBONO-----
//-----
Serial.print("Nivel de CO: "); //imprime etiqueta
Serial.print(coppm); //imprimimos el valor de co
Serial.println(" ppm"); //imprimimos unidades
//-----
//-----IMPRIME VALORES DE DIOXIDO DE CARBONO-----
//-----
Serial.print("Nivel de CO2: "); //imprime etiqueta

```

```

Serial.print(co2ppm);          //imprimimos el valor de co
Serial.println(" ppm");        //imprimimos unidades
//-----
//-----IMPRIME VALORES DE PRESION SONORA EN DB-----
//-----
Serial.print("SPL: ");         //imprime etiqueta
Serial.print(db);              //imprimimos el valor de decibelios
Serial.println(" db");         //imprimimos unidades
//-----
//-----
//          CONVERSION DE DATOS A ENTERO
//-----
Serial.print((int)h);Serial.println(" %");
Serial.print((int)t);Serial.println(" *C");
Serial.print((int)f);Serial.println(" *F");
Serial.print((int)tempe);Serial.println(" *C ");
Serial.print((int)presion);Serial.println(" hPa");
Serial.print((int)lux);Serial.println(" lx");
Serial.print((int)intensidad_UV);Serial.println(" mW/cm^2");
Serial.print((int)coppm);Serial.println(" ppm");
Serial.print((int)co2ppm);Serial.println(" ppm");
Serial.print((int)db);Serial.println(" db");
//-----
//-----
//          CONVERSION A STRING Y PUBLICA A MQTT
//-----
client.publish("wifitemp",String(t).c_str());
client.publish("wifihume",String(h).c_str());
client.publish("wifi_gradosF",String(f).c_str());
client.publish("wifitempe-bmp180",String(tempe).c_str());
client.publish("wifipresion",String(presion).c_str());
client.publish("wifilux",String(lux).c_str());
client.publish("wifiUV",String(intensidad_UV).c_str());
client.publish("wifiCo",String(coppm).c_str());
client.publish("wifiCo2",String(co2ppm).c_str());
client.publish("wifiDb",String(db).c_str());
delay(1000);
}
////////////////////////////////////
////////////////////////////////////
// toma un promedio de lecturas obtenidas por el pin
//nos devuelve el promedio
int averageAnalogRead(int pinToRead)
{
    byte numerolecturas = 8;
    unsigned int valorejec = 0;

    for(int x = 0 ; x < numerolecturas ; x++)
        valorejec += analogRead(pinToRead);
    valorejec /= numerolecturas;

    return(valorejec);
}

```

```
float mapfloat(float x, float in_min, float in_max, float out_min, float  
out_max)  
{  
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) +  
    out_min;  
}
```



ANEXO B

CODIGO DE ADQUISICION Y VISUALIZACION DE DATOS EN NODE RED

```
[{"id":"fc5755b5.243188","type":"tab","label":"Flow
1","disabled":false,"info":""},{id":"6bd92e98.491e6","type":"mqtt-
broker","z":"","name":"","broker":"192.168.0.10","port":"1883","clientId
":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true
,"willTopic":"","willQos":"0","willPayload":"","birthTopic":"","birthQos
":"0","birthPayload":""},{id":"9b3e3c4f.dd703","type":"ui_base","theme":{"
"name":"theme-
custom","lightTheme":{"default":"#0094CE","baseColor":"#0094CE","baseFont
":"-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-
serif","edited":true,"reset":false},"darkTheme":{"default":"#097479","bas
eColor":"#097479","baseFont":"-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-
serif","edited":true,"reset":false},"customTheme":{"name":"Untitled Theme
1","default":"#4B7930","baseColor":"#4B7930","baseFont":"-apple-
system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-
serif","reset":false},"themeState":{"base-
color":{"default":"#4B7930","value":"#4B7930","edited":false},"page-
titlebar-backgroundColor":{"value":"#4B7930","edited":false},"page-
backgroundColor":{"value":"#111111","edited":false},"page-sidebar-
backgroundColor":{"value":"#000000","edited":false},"group-
textColor":{"value":"#6db046","edited":false},"group-
borderColor":{"value":"#555555","edited":false},"group-
backgroundColor":{"value":"#333333","edited":false},"widget-
textColor":{"value":"#eeeeee","edited":false},"widget-
backgroundColor":{"value":"#4b7930","edited":false},"widget-
borderColor":{"value":"#333333","edited":false},"base-font":{"value":"-
apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-
Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"}}},"site":{"name":"Node-
RED
Dashboard","hideToolbar":"false","allowSwipe":"false","dateFormat":"DD/MM
/YYYY","sizes":{"sx":48,"sy":48,"gx":6,"gy":6,"cx":6,"cy":6,"px":0,"py":0
}}},{id":"755f2d87.674344","type":"firebase
config","z":"","firebaseurl":"base-de-datos-
bf8c6","loginType":"none"},{"id":"c19cle46.98f13","type":"ui_tab","z":"","
"name":"UNIDAD DE MONITOREO
AMBIENTAL","icon":"dashboard"},{"id":"6ef36b6f.4b9ea4","type":"ui_group",
"z":"","name":"Sensacion
termica","tab":"c19cle46.98f13","disp":true,"width":"6","collapse":false}
,{"id":"ba4114e7.1b2ba8","type":"mqtt-
broker","z":"","name":"","broker":"192.168.0.11","port":"1883","clientId
":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true
,"willTopic":"","willQos":"0","willPayload":"","birthTopic":"","birthQos
":"0","birthPayload":""},{id":"e4b85b87.978c78","type":"ui_group","z":"","
"name":"Presion","tab":"c19cle46.98f13","disp":true,"width":"6","collapse
":false}, {"id":"40db8c03.6a0e04","type":"ui_group","z":"","name":"Luminos
idad","tab":"c19cle46.98f13","disp":true,"width":"6","collapse":false}, {"
id":"f5725154.4586e","type":"ui_group","z":"","name":"Radiacion
UV","tab":"c19cle46.98f13","disp":true,"width":"6","collapse":false}, {"id
":"ca5d0b23.624478","type":"ui_group","z":"","name":"Monoxido de
carbono","tab":"c19cle46.98f13","disp":true,"width":"6","collapse":false}
,{"id":"5d289d5e.33b434","type":"ui_group","z":"","name":"Humedad
relativa","tab":"c19cle46.98f13","disp":true,"width":"6","collapse":false}
,{"id":"c1dfa114.4ac15","type":"ui_group","z":"","name":"Dioxido de
carbono","tab":"c19cle46.98f13","disp":true,"width":"6","collapse":false}
,{"id":"b18fe24.1ea442","type":"ui_group","z":"","name":"Presion
```

```
sonora", "tab": "c19c1e46.98f13", "disp": true, "width": "6", "collapse": false},
{"id": "1505ac7c.30c264", "type": "ui_group", "z": "", "name": "Temperatura", "tab": "c19c1e46.98f13", "disp": true, "width": "6", "collapse": false}, {"id": "5d6635c1.1c966c", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifiDb", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 117.89453125, "y": 41.3515625, "wires": [{"830a6d6b.4bc8c"}]}, {"id": "1c7fd694.d16c29", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifitemp", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 115.8828125, "y": 109.359375, "wires": [{"dce0335a.44196"}]}, {"id": "377690d9.291e2", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifiCo2", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 112.88671875, "y": 177.3593807220459, "wires": [{"d13b80a6.2de91"}]}, {"id": "15d8d6e9.f70909", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifiCo", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 124.8828125, "y": 290.3632879257202, "wires": [{"9d2b5bdf.bd2098"}]}, {"id": "8ae5425e.1f089", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifiUV", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 121.88671875, "y": 342.36719512939453, "wires": [{"71a8d637.8fc958"}]}, {"id": "f187ca9.0628538", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifilux", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 114.88671875, "y": 402.3671875, "wires": [{"747eae7e.34dlb"}]}, {"id": "a56f81bc.e4b8", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifipresion", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 113.8828125, "y": 458.3672161102295, "wires": [{"bb151baf.1b7f88"}]}, {"id": "3252b83f.39c448", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifitempe-
bmp180", "qos": "2", "broker": "ba4114e7.1b2ba8", "x": 110.8984375, "y": 549.3554992675781, "wires": [{"2ed8a759.330208"}]}, {"id": "8ca4f201.a3712", "type": "m
qtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifi_gradosF", "qos": "2", "bro
ker": "ba4114e7.1b2ba8", "x": 87.89062881469727, "y": 614.3515701293945, "wires
": [{"7d472857.000908"}]}, {"id": "18d8d585.29489a", "type": "mqtt
in", "z": "fc5755b5.243188", "name": "", "topic": "wifihume", "qos": "2", "broker":
"ba4114e7.1b2ba8", "x": 135.89453125, "y": 234.3554973602295, "wires": [{"e33d625b.6189c"}]}, {"id": "830a6d6b.4bc8c", "type": "firebase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifiDb", "method": "set", "value": "msg.payload", "priority": "m
sg.priority", "x": 465.8828125, "y": 32.875, "wires": [{"32321670.ac869a", "900c8443.fff5b8"}]}, {"id": "dce0335a.44196", "type": "firebase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifitemp", "method": "set", "value": "msg.payload", "priority":
"msg.priority", "x": 448.88282012939453, "y": 100.87500476837158, "wires": [{"d
b7c9e69.adf39", "94d31e92.f6bad"}]}, {"id": "d13b80a6.2de91", "type": "firebas
e
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifiCo2", "method": "set", "value": "msg.payload", "priority": "
msg.priority", "x": 427.89844512939453, "y": 159.87109851837158, "wires": [{"2951b7c3.b73dc8", "9248be2b.e493a"}]}, {"id": "e33d625b.6189c", "type": "firebas
e
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifihume", "method": "set", "value": "msg.payload", "priority":
"msg.priority", "x": 448.88672637939453, "y": 223.86719369888306, "wires": [{"a
ee29e57.dc35b", "b04848d1.ed1cb8"}]}, {"id": "9d2b5bdf.bd2098", "type": "fireb
ase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifiCo", "method": "set", "value": "msg.payload", "priority": "m
sg.priority", "x": 452.89453125, "y": 281.87890625, "wires": [{"49c2809f.98ae7", "c5b9419a.16e65"}]}, {"id": "71a8d637.8fc958", "type": "firebase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifiUV", "method": "set", "value": "msg.payload", "priority": "m
sg.priority", "x": 458.890625, "y": 326.87890625, "wires": [{"a819d0b6.aebbf", "}
```

```

elbcf8f0.c48478"]]}, {"id": "747eae7e.34dlb", "type": "firebase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifilux", "method": "set", "value": "msg.payload", "priority": "
msg.priority", "x": 463.890625, "y": 385.8828125, "wires": [{"4c62e550.744bbc",
"80efd894.0a5498"}]}, {"id": "bb151baf.1b7f88", "type": "firebase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifipresion", "method": "set", "value": "msg.payload", "priorit
y": "msg.priority", "x": 463.88282775878906, "y": 439.8711223602295, "wires": [{"
"298560d5.bdf6a", "c7a607f2.f3bc98"}]}, {"id": "2ed8a759.330208", "type": "fir
ebase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifitempe-
bmp180", "method": "set", "value": "msg.payload", "priority": "msg.priority", "x
": 469.8906555175781, "y": 543.8829288482666, "wires": [{"7745025a.5eac7c", "75
0c7b4b.0a2114"}]}, {"id": "7d472857.000908", "type": "firebase
modify", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344
", "childpath": "wifi_gradosF", "method": "set", "value": "msg.payload", "priori
ty": "msg.priority", "x": 423.88282775878906, "y": 619.8672180175781, "wires": [{"
"6b2f4e88.9e2a2", "a37db05f.e84b"}]}, {"id": "32321670.ac869a", "type": "ui_g
auge", "z": "fc5755b5.243188", "name": "", "group": "b18fe24.1ea442", "order": 0,
"width": 0, "height": 0, "gtype": "gage", "title": "Presion
sonora", "label": "dB", "format": "{value}", "min": "39", "max": "130", "colors"
: ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1169.89075851440
43, "y": 20, "wires": []}, {"id": "900c8443.fff5b8", "type": "ui_chart", "z": "fc57
55b5.243188", "name": "", "group": "b18fe24.1ea442", "order": 0, "width": 0, "heig
ht": 0, "label": "Presion
sonora", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpo
late": "linear", "nodata": "", "dot": true, "ymin": "39", "ymax": "130", "removeOld
er": 1, "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneCol
or": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d
62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1173.894664
7644043, "y": 56.9921875, "wires": [{"], [{"], {"id": "db7c9e69.adf39", "type": "ui
_gauge", "z": "fc5755b5.243188", "name": "", "group": "6ef36b6f.4b9ea4", "order"
: 0, "width": 0, "height": 0, "gtype": "gage", "title": "Sensacion
termica", "label": "°C", "format": "{value}", "min": 0, "max": "60", "colors": [{"
#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1179.8907585144043,
"y": 97.01171875, "wires": [{"], {"id": "2951b7c3.b73dc8", "type": "ui_gauge", "z"
: "fc5755b5.243188", "name": "", "group": "cldfa114.4ac15", "order": 0, "width": 0
, "height": 0, "gtype": "gage", "title": "Dioxido de
carbono", "label": "ppm", "format": "{value}", "min": "400", "max": "2000", "col
ors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1185.8985061
645508, "y": 164.45314407348633, "wires": [{"], {"id": "94d31e92.f6bad", "type": "
ui_chart", "z": "fc5755b5.243188", "name": "", "group": "6ef36b6f.4b9ea4", "orde
r": 0, "width": 0, "height": 0, "label": "Sensacion
termica", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interp
olate": "linear", "nodata": "", "dot": true, "ymin": "0", "ymax": "60", "removeOlde
r": 1, "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColo
r": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d6
2728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1180.8868522
644043, "y": 131.9921875, "wires": [{"], [{"], {"id": "9248be2b.e493a", "type": "ui
_chart", "z": "fc5755b5.243188", "name": "", "group": "cldfa114.4ac15", "order":
0, "width": 0, "height": 0, "label": "Dioxido de
carbono", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interp
olate": "linear", "nodata": "", "dot": true, "ymin": "400", "ymax": "2000", "remove
Older": 1, "removeOlderPoints": "", "removeOlderUnit": "3600", "cutout": 0, "useO
neColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a
", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1187.8
906784057617, "y": 200.4375228881836, "wires": [{"], [{"], {"id": "aee29e57.dc35b
", "type": "ui_gauge", "z": "fc5755b5.243188", "name": "", "group": "5d289d5e.33b
434", "order": 0, "width": 0, "height": 0, "gtype": "gage", "title": "Humedad
relativa", "label": "%", "format": "{value}", "min": 0, "max": "100", "colors": [{"

```

```
"#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1188.8906784057617, "y": 235.45314979553223, "wires": [], {"id": "b04848d1.ed1cb8", "type": "ui_chart", "z": "fc5755b5.243188", "name": "", "group": "5d289d5e.33b434", "order": 0, "width": 0, "height": 0, "label": "Humedad relativa", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": true, "ymin": "0", "ymax": "100", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1188.8828659057617, "y": 270.4258050918579, "wires": [[], []], {"id": "49c2809f.98ae7", "type": "ui_gauge", "z": "fc5755b5.243188", "name": "", "group": "ca5d0b23.624478", "order": 0, "width": 0, "height": 0, "gtype": "gage", "title": "Monoxido de carbono", "label": "ppm", "format": "{{value}}", "min": 0, "max": "500", "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1180.8985061645508, "y": 305.4570903778076, "wires": [], {"id": "c5b9419a.16e65", "type": "ui_chart", "z": "fc5755b5.243188", "name": "", "group": "ca5d0b23.624478", "order": 0, "width": 0, "height": 0, "label": "Monoxido de carbono", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": true, "ymin": "", "ymax": "", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1178.8906936645508, "y": 344.42584228515625, "wires": [[], []], {"id": "a819d0b6.aebbf", "type": "ui_gauge", "z": "fc5755b5.243188", "name": "", "group": "f5725154.4586e", "order": 0, "width": 0, "height": 0, "gtype": "gage", "title": "Radiacion UV", "label": "mW/cm^2", "format": "{{value}}", "min": 0, "max": "15", "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1179.8829307556152, "y": 384.449294090271, "wires": [], {"id": "elbcf8f0.c48478", "type": "ui_chart", "z": "fc5755b5.243188", "name": "", "group": "f5725154.4586e", "order": 0, "width": 0, "height": 0, "label": "Radiacion UV", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": true, "ymin": "0", "ymax": "15", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1179.8867721557617, "y": 416.4258270263672, "wires": [[], []], {"id": "4c62e550.744bbc", "type": "ui_gauge", "z": "fc5755b5.243188", "name": "", "group": "40db8c03.6a0e04", "order": 0, "width": 0, "height": 0, "gtype": "gage", "title": "Luminosidad", "label": "lux", "format": "{{value}}", "min": "1", "max": "65536", "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1172.8945999145508, "y": 450.44537925720215, "wires": [], {"id": "80efd894.0a5498", "type": "ui_chart", "z": "fc5755b5.243188", "name": "", "group": "40db8c03.6a0e04", "order": 0, "width": 0, "height": 0, "label": "Luminosidad", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": true, "ymin": "1", "ymax": "20000", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1173.890682220459, "y": 487.4375171661377, "wires": [[], []], {"id": "298560d5.bdf6a", "type": "ui_gauge", "z": "fc5755b5.243188", "name": "", "group": "e4b85b87.978c78", "order": 0, "width": 0, "height": 0, "gtype": "gage", "title": "Presion", "label": "mBar", "format": "{{value}}", "min": "300", "max": "1100", "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 1159.886775970459, "y": 529.4492359161377, "wires": [], {"id": "7745025a.5eac7c", "type": "ui_gauge", "z": "fc5755b5.243188", "name": "", "group": "1505ac7c.30c264", "order": 0, "width": 0, "height": 0, "gtype": "wave", "title": "Gauge", "label": "°C", "format": "{{value}}", "min": 0, "max": "60", "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 825.1666870117188, "y": 557.0000152587891, "wires": [], {"id": "6b2f4e88.9e2a2", "type": "ui_gauge", "z": "fc5755b5.243188", "name": "", "group": "6ef36b6f.4b9ea4", "order": 0, "width": 0, "height": 0, "gtype": "wave", "title": "Gauge", "label": "°F", "format": "{{value}}", "min": 0, "max": "100", "colors": ["#00b500", "#e6e600",
```

```
"#ca3838"], "seg1": "", "seg2": "", "x": 829.1666870117188, "y": 680.000015258789
1, "wires": []}, {"id": "750c7b4b.0a2114", "type": "ui_text", "z": "fc5755b5.2431
88", "group": "1505ac7c.30c264", "order": 0, "width": 0, "height": 0, "name": "", "l
abel": "Temperatura", "format": "{msg.payload}°C", "layout": "row-
center", "x": 847.1666870117188, "y": 599.0000152587891, "wires": []}, {"id": "a3
7db05f.e84b", "type": "ui_text", "z": "fc5755b5.243188", "group": "6ef36b6f.4b9
ea4", "order": 0, "width": 0, "height": 0, "name": "", "label": "Grados
Fahrenheit", "format": "{msg.payload}°F", "layout": "row-
center", "x": 872.1666870117188, "y": 723.0000152587891, "wires": []}, {"id": "c7
a607f2.f3bc98", "type": "ui_chart", "z": "fc5755b5.243188", "name": "", "group":
"e4b85b87.978c78", "order": 0, "width": 0, "height": 0, "label": "Presion", "chart
Type": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear"
, "nodata": "", "dot": true, "ymin": "300", "ymax": "1100", "removeOlder": 1, "remov
eOlderPoints": "", "removeOlderUnit": "60", "cutout": 0, "useOneColor": false, "c
olors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9
896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1161.166748046875, "y": 5
66.6666870117188, "wires": [[], []]}, {"id": "847946f1.bfed18", "type": "firebas
e.on", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344",
"childpath": "wifitemp", "atStart": true, "eventType": "value", "queries": [], "x
": 261.16668701171875, "y": 762.333381652832, "wires": [{"72eb3285.9df14c"}]},
{"id": "72eb3285.9df14c", "type": "debug", "z": "fc5755b5.243188", "name": "", "a
ctive": true, "tosidebar": true, "console": false, "tostatus": false, "complete":
"false", "x": 1095.1667098999023, "y": 925.3333811759949, "wires": []}, {"id": "a
a62e7c5.421da8", "type": "firebase.on", "z": "fc5755b5.243188", "name": "", "fir
eabaseconfig": "755f2d87.674344", "childpath": "wifihume", "atStart": true, "eve
ntType": "value", "queries": [], "x": 255.16668701171875, "y": 815.3333840370178
, "wires": [{"72eb3285.9df14c"}]}, {"id": "b64d3e5b.2ba97", "type": "firebase.o
n", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344", "ch
ildpath": "wifi_gradosF", "atStart": true, "eventType": "value", "queries": [], "
x": 264.16668701171875, "y": 869.3333864212036, "wires": [{"72eb3285.9df14c"}]
}, {"id": "f3cb806.3230a8", "type": "firebase.on", "z": "fc5755b5.243188", "name
": "", "firebaseconfig": "755f2d87.674344", "childpath": "wifitempe-
bmp180", "atStart": true, "eventType": "value", "queries": [], "x": 280.166687011
71875, "y": 924.3333902359009, "wires": [{"72eb3285.9df14c"}]}, {"id": "43b303d
8.87b0ac", "type": "firebase.on", "z": "fc5755b5.243188", "name": "", "firebasec
onfig": "755f2d87.674344", "childpath": "wifipresion", "atStart": true, "eventT
ype": "value", "queries": [], "x": 286.16668701171875, "y": 986.3333930969238, "w
ires": [{"72eb3285.9df14c"}]}, {"id": "df098f8a.3fec4", "type": "firebase.on",
"z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344", "child
path": "wifilux", "atStart": true, "eventType": "value", "queries": [], "x": 333.1
6668701171875, "y": 1047.333396911621, "wires": [{"72eb3285.9df14c"}]}, {"id":
"51fb8822.e70668", "type": "firebase.on", "z": "fc5755b5.243188", "name": "", "f
irebaseconfig": "755f2d87.674344", "childpath": "wifiUV", "atStart": true, "eve
ntType": "value", "queries": [], "x": 279.16668701171875, "y": 1112.333398818969
7, "wires": [{"72eb3285.9df14c"}]}, {"id": "59d306d4.814d08", "type": "firebase
.on", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.674344", "
childpath": "wifiCo2", "atStart": true, "eventType": "value", "queries": [], "x":
282.16668701171875, "y": 1223.3334064483643, "wires": [{"72eb3285.9df14c"}]},
{"id": "6731bba3.6c8824", "type": "firebase.on", "z": "fc5755b5.243188", "name":
"", "firebaseconfig": "755f2d87.674344", "childpath": "wifiDb", "atStart": tru
e, "eventType": "value", "queries": [], "x": 282.1666793823242, "y": 1282.3334102
630615, "wires": [{"72eb3285.9df14c"}]}, {"id": "1a3d4a5c.37ddc6", "type": "fir
ebase.on", "z": "fc5755b5.243188", "name": "", "firebaseconfig": "755f2d87.6743
44", "childpath": "wifiCo", "atStart": true, "eventType": "value", "queries": [],
"x": 259.1666717529297, "y": 1175.3333740234375, "wires": [{"72eb3285.9df14c"}
]]]
```

ANEXO C

ESPECIFICACIONES TECNICAS DEL FUNCIONAMIENTO DEL EQUIPO

➤ ESPECIFICACIONES ELECTRICAS DEL CUADRICOPTERO

- Tipo de batería: LiPo
- Baterías utilizadas: Tiger Power & ChinaHobbyLine
- Voltaje de alimentación: 11.1 voltios
- Numero de celdas: 3 celdas
- Tipo de Alimentación: Continua
- Motores: A2212/ 13T
- ESC: 30 A
- Capacidad de baterías: 5400 mAh & 4000mAh

➤ ESPECIFICACIONES DEL CONTROLADOR DE VUELO

- Tarjeta de control: APM 2.8
- Procesador: atmega2560 y atmega32u-2
- Velocidad del Procesador: 16MHz
- Numero de bits: 8 bits

➤ ESPECIFICACIONES DE EMISORA FLYSKY

- Cantidad de canales: 6 canales
- Ancho de banda: 500 Hz.
- Frecuencia de trabajo: 2.4 GHz. a 2.48 GHz.
- Transmisión de energía: 20dBm (100 mW)
- Fuente de alimentación: 8 pilas AA
- Alcance a campo abierto: 100 metros
- Alcance con obstáculos :40 metros

➤ **SISTEMA DE COMUNICACIÓN Y VISUALIZACION**

- Estación en tierra: Mission Planner
- Interfaz : Inalámbrica
- Comunicación inalámbrica: Telemetría 433MHz

➤ **ESPECIFICACIONES DE FUNCIONAMIENTO**

- Tiempo de vuelo : 7 minutos
- Capacidad de carga: 1kg
- Modos de vuelo configurados: altitude hold, position hold, stabilize
- Peso: 1.0498 kilos

➤ **ESPECIFICACIONES ELECTRICAS DEL SISTEMA DE MONITOREO**

- Tipo de Alimentación: DC
- Alimentación: 3.3 voltios
- Regulador interno de voltaje: USB 5 voltios
- Corriente requerida: 1A

➤ **PARAMETROS Y RANGOS DE MEDICION DEL SISTEMA DE MONITOREO**

- Temperatura: 0 a 60 °C
- Humedad relativa : 0 a 100 %
- Radiación uv : 0 a 15 mW/cm²
- Presión barométrica : 300 a 1100 mbar
- Luminosidad : 1 a 65536 lux
- Nivel de CO₂ : 400 a 2000 ppm
- Nivel de CO : 0 a 500 ppm
- Nivel de presión sonora : 40 a 130.50 decibelios

➤ **ESPECIFICACIONES DEL SISTEMA DE CONTROL**

- SoC: ESP8266 (Módulo ESP-12)
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Pin Analógico ADC: 1 de 10 bits
- Módulo expensor adc: ads1115 (16 bits de resolución)

➤ **SISTEMA DE COMUNICACIÓN DEL EQUIPO**

- Interfaz: Inalámbrica
- Velocidad de transmisión: 115200 baudios
- Comunicación inalámbrica: Wifi 802.11 b/g/n
- Alcance:10 metros

➤ **ESPECIFICACIONES DE VISUALIZACION Y ALMACENAMIENTO
DE DATOS**

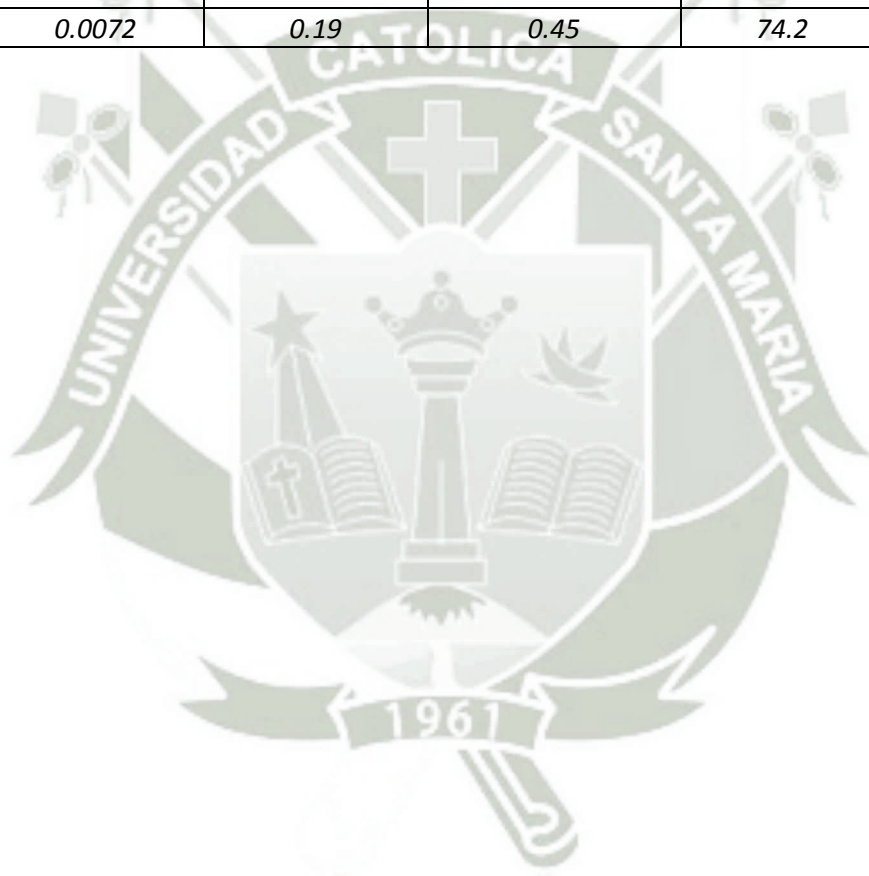
- Interfaz:Node-red
- Tipo de graficas visualizadas: Gauge & chart
- Bróker:Mqtt Mosca
- Base de datos:Firebase
- Formato de almacenamiento: Json

ANEXO C

RESULTADOS OBTENIDOS POR EL SOFTWARE PROP CALC PARA LA SELECCION DE HELICES 1045

J	CT	CP	eta	Thr [N]	P [W]	v [m/s]	Re 0.7
0.05	0.1009	0.0343	0.15	20.285	353	2.6	163944
0.07	0.0989	0.0343	0.19	19.881	353.8	3.5	163979
0.09	0.0967	0.0344	0.24	19.447	354.2	4.4	164025
0.1	0.0944	0.0344	0.28	18.992	354.1	5.3	164080
0.12	0.0921	0.0343	0.32	18.517	353.7	6.2	164147
0.14	0.0897	0.0342	0.36	18.029	352.9	7.1	164223
0.16	0.0872	0.0341	0.4	17.528	351.6	8	164310
0.17	0.0846	0.0339	0.43	17.015	349.8	8.9	164408
0.19	0.082	0.0337	0.46	16.49	347.6	9.8	164516
0.21	0.0793	0.0335	0.49	15.956	344.9	10.7	164634
0.23	0.0766	0.0331	0.52	15.411	341.6	11.6	164762
0.24	0.0739	0.0328	0.55	14.853	337.7	12.5	164901
0.26	0.071	0.0323	0.57	14.282	333	13.4	165050
0.28	0.0681	0.0318	0.6	13.696	327.6	14.3	165209
0.3	0.0651	0.0312	0.62	13.098	321.5	15.2	165378
0.31	0.0621	0.0305	0.64	12.493	314.6	16.1	165558
0.33	0.0591	0.0298	0.66	11.879	307.2	17	165748
0.35	0.0559	0.029	0.67	11.25	299.1	17.9	165948
0.37	0.0528	0.0282	0.69	10.613	290.4	18.8	166157
0.38	0.0496	0.0272	0.7	9.966	280.8	19.7	166377
0.4	0.0461	0.0261	0.71	9.269	269	20.6	166607
0.42	0.0425	0.0248	0.72	8.539	255.6	21.5	166847
0.44	0.0387	0.0233	0.72	7.775	240.3	22.4	167097
0.45	0.0348	0.0218	0.73	7	224.3	23.3	167357
0.47	0.031	0.0202	0.72	6.224	208.1	24.2	167626

0.49	0.0271	0.0186	0.71	5.453	191.5	25.1	167905
0.51	0.0233	0.0169	0.7	4.688	174.4	26	168194
0.52	0.0196	0.0153	0.67	3.932	157.2	26.9	168493
0.54	0.0158	0.0135	0.63	3.173	138.9	27.8	168801
0.56	0.0121	0.0117	0.58	2.423	120.5	28.7	169119
0.58	0.0086	0.01	0.49	1.72	103.3	29.6	169446
0.59	0.005	0.0083	0.36	1.01	85.2	30.5	169783
0.61	0.0022	0.0072	0.19	0.45	74.2	31.4	170129



ANEXO C

RESULTADOS OBTENIDOS POR EL SOFTWARE PROP CALC PARA LA SELECCION DE HELICES 9045

J	CT	CP	eta	Thr [N]	P [W]	v [m/s]	Re 0.7
0.05	0.0986	0.0554	0.09	0.092	0.2	0.2	11434
0.08	0.0955	0.0579	0.13	0.089	0.2	0.3	11437
0.1	0.092	0.0608	0.15	0.086	0.2	0.4	11443
0.13	0.0882	0.0642	0.18	0.083	0.2	0.5	11450
0.15	0.0841	0.0683	0.19	0.079	0.2	0.6	11459
0.18	0.0796	0.0719	0.2	0.074	0.3	0.7	11469
0.21	0.0748	0.0752	0.21	0.07	0.3	0.8	11481
0.23	0.0698	0.0779	0.21	0.065	0.3	0.9	11495
0.26	0.0642	0.0799	0.21	0.06	0.3	1	11510
0.29	0.0584	0.0807	0.21	0.055	0.3	1.1	11527
0.31	0.0522	0.0794	0.2	0.049	0.3	1.2	11545
0.34	0.0457	0.0771	0.2	0.043	0.3	1.3	11565
0.36	0.04	0.0704	0.21	0.037	0.3	1.4	11586
0.39	0.0349	0.0613	0.22	0.033	0.2	1.5	11609
0.42	0.0302	0.0503	0.25	0.028	0.2	1.6	11634
0.44	0.0254	0.0416	0.27	0.024	0.1	1.7	11660
0.47	0.02	0.0343	0.27	0.019	0.1	1.8	11688
0.5	0.0144	0.028	0.26	0.014	0.1	1.9	11717
0.52	0.0091	0.0218	0.22	0.009	0.1	2	11748
0.55	0.0034	0.017	0.11	0.003	0.1	2.1	11780

ANEXO D
PLANOS DE ESTRUCTURA DE DRONE DJI F450

