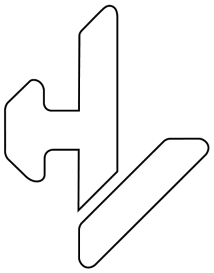

Título: Proyecto FINAL

Subtitulo Training Center

Grupo de Reporte Partners

ESCRITO POR	
REVISADO POR	
ÚLTIMA MODIFICACIÓN	30/03/2024

Tabla de Contenido:

1. Fechas

Marzo						
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25 Proy. Final	26	27	28	29	30	31

Abril						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Mayo						
		1	2	3	4	5
6	7	8	9	10	11	12
13 Entrega Proyecto Final	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

2. Proyecto Final

Introducción.

Se pretende que el proyecto final sea el diseño completo de un POD que cumpla con los requisitos básicos técnicos de propulsarse y poder levitar.

Para ello deberán de cohesionar todos los subsistemas, por lo que vamos a establecer unos objetivos mínimos por cada uno para que todo pueda funcionar.

Dimensiones máximas: 60cm x 100cm x 40cm máximo 100kg de peso

Presupuesto máximo: 37.500€

Equipo A: 11 Personas

Mechanics: David / Maria / Nadia / Pedro / Vizu
Electromagnetics: Sofia
Firmware: Jose
Software: Qiang
Outreach: Raquel / Mateo
Partners & Economics: Bruno

Equipo B: 10 Personas

Mechanics: Alejandro / Pablo / Bego / Tudela
Electromagnetics: Anderson
Firmware:Alex
Software: Aitana
Outreach: Irene / María
Partners & Economics: Maria

Mechanics:

Para el proyecto final los alumnos de mechanics deberán de diseñar y tener en cuenta para fabricar el mini pod como mínimo:

- Chassis.
- Anclajes para el motor y la levitación (Comunicación con los alumnos de electromagnetics)
- Espacio para integración de diseño de interior.
- Espacio para la integración de las baterías y la electrónica (Para saber el espacio de la batería por ejemplo, electromagnetics ha de marcar los requerimientos que necesita, luego se ha de buscar una batería real que los cumpliría y con sus medidas dejar espacio para ella y su integración (paso de cables, etc))
- Integración del “mini pod” con el track que diseñaron anteriormente.

Outreach:

De cara el diseño final del “min pod”, cada equipo de outreach será de 2 alumnos, que deberán de presentar los siguientes objetivos.

1. Diseño de interior del “mini pod” podrán utilizar el material que hayan realizado en el proyecto de subsistemas. Deberá de ser una cabina de pasajeros en un pod hyperloop.
    a. Butacas
    b. Mesas
    c. Portamaletas
    d. Luminaria
2. Se deberá de realizar en blender un video del explosionado de la propuesta de pod completa.
3. Deberán de realizar una cartelería de su propuesta de pod. Con los colores corporativos .

Partners & Economics:

Su función se centrará en controlar la viabilidad del proyecto en cuanto a los compañeros de ingeniería. Deberán asegurarse de que los materiales, piezas, componentes que utilicen el resto de compañeros sean viables.

Para ello se pondrá un presupuesto al que se deberán de ajustar todos los alumnos. Se llevará un control por excel de este presupuesto, dividiéndolo en subsistemas y dentro de cada subsistema las diferentes piezas.

En cuanto a la tarea de partners, tendrá relación con buscar la empresa de la que se obtendrían los materiales, todos los materiales del proyecto deberán de conseguirse de diferentes empresas.

Electromagnetics:

Se les pedirá que hagan un sistema de levitación que tenga como mínimo la fuerza suficiente para levantar el vehículo.

Deberán de comunicarse con principalmente mechanics para saber peso, espacio, anclajes, track y demás.

Firmware:

Implementar un control en bucle abierto basado en una máquina de estados con 3 estados, IDLE, OPERATIONAL, FAULT.

La placa deberá empezar en el estado IDLE, y solamente cambiar a OPERATIONAL tras el usuario pulsar un botón externo, se proporcionara un circuito en una protoboard con dos botones, uno para transicionar de IDLE a OPERATIONAL y otro para ir en cualquier momento a FAULT. (*los pines de la placa a los que se conectarán los botones son de libre elección*). En el estado de OPERATIONAL se deberá leer un sensor analogico, revisar documentación sobre ADCs. Se deberá implementar un pequeño algoritmo que a través del duty cycle de una PWM de salida regule la lectura del sensor.

El valor deseado de lectura será de 1.6V, si el duty de salida es mayor que el anterior la lectura del sensor será aleatoriamente mayor , y viceversa.

En cada Estado deberá parpadear un led de estado de la NUCLEO a una frecuencia de 5 Hz, es decir en OPERATIONAL el Led Verde, en IDLE el naranja y en FAULT el rojo.

Adicionalmente, emplear CAN/FDCAN en modo loopback para enviar las lecturas del sensor periodicamente, (20 Hz)

Como medida de seguridad en el momento en el que se pulse el botón de transición a FAULT la placa deberá dejar de generar la PWM, dejar de leer el sensor, y de enviar mensajes por CAN.

Software:

Deberán diseñar y programar que se comunique con el vehículo y una interfaz que muestre de manera gráfica la telemetría del vehículo. Tendrán que asumir que el vehículo les enviará

datos de la siguiente manera: El vehículo tendrá un servidor UDP al que se conectará la aplicación. Una vez conectada, el vehículo mandará datos con formato JSON y los siguientes campos: nombre, valor y timestamp. El nombre será un identificador para el valor enviado (uno entre “airgap”, “speed”, “current” y “voltage”) , el valor será en si el valor que tiene el vehículo y el timestamp el momento en el que se generó ese dato. El timestamp será el número de microsegundos desde la época Unix.

Los datos recibidos se deberán enviar al frontend mediante WebSockets. El frontend tendrá que mostrar en tiempo real los últimos valores recibidos en forma de número y de alguna manera gráfica (gráficas, indicadores u otra representación que se entienda).

Para resumir, la aplicación como minimo debera, para el backend:

- Conectarse por UDP a una dirección y puerto específicos
- Recibir por UDP mensajes con formato JSON y comprenderlos
- Enviar por WebSocket los datos recibidos

Y para el frontend:

- Recibir por websocket los mensajes JSON y comprenderlos
- Mostrar el valor numérico que lleva cada mensaje
- Mostrar gráficamente el valor que lleva cada mensaje

Adicionalmente, se valorará muy positivamente lo siguiente:

- Los datos se guarden a un fichero en el disco para poder usarlos posteriormente
- Mas de un frontend pueda recibir los datos
- Enviar desde el frontend un mensaje JSON y que este se envía por la misma conexión UDP que los datos para el vehículo
- La visualización gráfica de los valores no muestre solo el último valor
- La interfaz tenga una estructura intuitiva
- Hya un diseño de la interfaz previo a su programación
- El código se suba poco a poco a GitHub