**“*UNIVERSIDAD CATÓLICA BOLIVIANA SAN PABLO”***

***DEPARTAMENTO DE CIENCIAS EXACTAS E INGENIERIAS***



**NOMBRES:**

Luis Antonio Lanza Tordoya

Carlos Daniel Flores Argote

Kevin Horacio Maita Vargas

**DOCENTE:** Elmer Alan Cornejo Quito

**CARRERA:** Ingeniería Mecatrónica

**MATERIA:** Sistemas Embebidos II (IMT-322)

Tarija – Bolivia

**Plataforma embebida**

NVIDIA JETSON NANO

**Aplicación**

El proyecto consiste en desarrollar un vehículo autónomo para la entrega de paquetes utilizando la placa de desarrollo Nvidia Jetson Nano. La visión artificial se empleará como parte fundamental del sistema para permitir al vehículo detectar y reconocer su entorno, así como identificar obstáculos y rutas de navegación.

La comunicación entre la Jetson Nano y Arduino se establecerá a través de una conexión serial. Arduino será responsable del control de los motores del vehículo y de garantizar la seguridad de la placa principal (Jetson Nano) al evitar daños por sobrecargas eléctricas o interferencias.

Mediante el uso de la Jetson Nano y su capacidad de procesamiento de alto rendimiento, se podrán ejecutar algoritmos de visión artificial en tiempo real para realizar tareas como detección y seguimiento de objetos, reconocimiento de señales o semáforos, y toma de decisiones autónomas para la navegación.

La combinación de la potencia de la Jetson Nano y la flexibilidad de Arduino permitirá construir un sistema de entrega de paquetes autónomo eficiente y seguro, capaz de operar en entornos reales. Además, la conexión serial entre ambas placas facilitará la comunicación y coordinación precisa entre los componentes del vehículo.

Este proyecto ilustra la integración de tecnologías como visión artificial, control de motores y comunicación serial para construir soluciones autónomas y aplicar la inteligencia artificial en la entrega de paquetes de manera eficiente y segura.

**Periféricos (1 ó 2)**

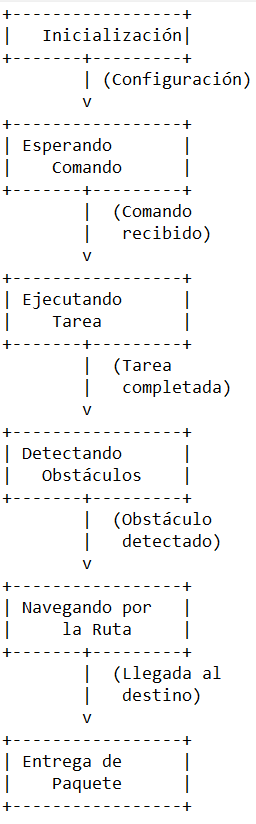
Los periféricos que se utilizarán en este proyecto son GPIO (General Purpose Input/Output) y UART (Universal Asynchronous Receiver/Transmitter) a través del protocolo I2C.

El GPIO permite la comunicación bidireccional de señales digitales entre la Jetson Nano y Arduino. Se utilizará para controlar los motores del vehículo, así como para la detección de eventos externos, como sensores de proximidad o interruptores. Los pines GPIO proporcionan flexibilidad en la conexión y control de dispositivos externos.

Por otro lado, el UART a través del protocolo I2C se utilizará para establecer una comunicación serial entre la Jetson Nano y Arduino. Esta conexión permitirá el intercambio de datos de manera más eficiente y confiable. Se utilizará para enviar comandos y recibir datos desde Arduino, lo cual es fundamental para el control del vehículo y para garantizar la integridad de la comunicación entre ambas placas.

El uso de estos periféricos en conjunto brinda una solución completa para la interconexión y control del vehículo autónomo. El GPIO permite el control directo de los motores y la interacción con los sensores externos, mientras que el UART a través del protocolo I2C facilita una comunicación serial estable y confiable entre la Jetson Nano y Arduino. Esta combinación de periféricos es ideal para aplicaciones de robótica y sistemas embebidos, permitiendo una comunicación efectiva y un control preciso de los componentes del vehículo autónomo.

**Diagrama de estado de MEF con una breve descripción de cada estado.**



**Incluir una breve descripción de cada estado**

Aquí tienes un ejemplo de diagrama de estado de una Máquina de Estados Finita (MEF) para el sistema de entrega de paquetes autónomo utilizando la Jetson Nano y Arduino:

En este diagrama, los estados se representan mediante rectángulos y las transiciones se muestran con flechas. A continuación se describe cada uno de los estados:

**Inicialización:** Estado inicial donde se configuran los componentes y se espera el comando de inicio.

**Esperando Comando:** Estado en el que el sistema está esperando recibir un comando para iniciar la entrega de paquetes.

**Ejecutando Tarea:** Estado en el que se ejecuta una tarea específica, como la detección de obstáculos o la planificación de rutas.

**Detectando Obstáculos:** Estado en el que el sistema utiliza la visión artificial para detectar obstáculos en el entorno.

**Navegando por la Ruta:** Estado en el que el vehículo se desplaza por la ruta planificada hacia el destino.

**Entrega de Paquete:** Estado final en el que se realiza la entrega del paquete en el destino.

**Definir los módulos de software (archivos) que va implementar para cada periférico.**

**- `main.py`:** Este archivo sería el punto de entrada principal del programa. Aquí se llevaría a cabo el procesamiento de la imagen obtenida de la cámara y se realizarían las adaptaciones necesarias según los requisitos del proyecto. Incluye la inicialización de los componentes, como la cámara y la comunicación con Arduino, así como la implementación de la lógica principal del vehículo autónomo. Además, se pueden utilizar las funciones encapsuladas del archivo `funciones.py` para aplicar técnicas como el control PID y el envío de valores a Arduino.

**- `funciones.py`:** Este archivo contiene las funciones encapsuladas necesarias para tareas específicas, como el control PID y el envío de valores a Arduino. Por ejemplo, la funcion pid\_control que tome las entradas necesarias (referencia, medida) y devuelve la señal de control calculada mediante el algoritmo de control PID. También incluye una función con un valor como argumento y lo envíe a Arduino a través del módulo de comunicación.