Mini Sumo-Bot

Andrés Felipe González – Álvaro Esteban Jiménez – Alain Solano

# INTRODUCCION

Se requiere hacer un mini sumo-bot, con la tarjeta de desarrollo STM32F103C8T6 (Blue Pill), con el fin de participar en una competencia de mini sumos de la feria de proyectos organizada por la escuela de ingeniería. Para ello se tuvo que diseñar cada pieza de chasis e imprimirla en 3D, así mismo diseñar el circuito necesario para el funcionamiento deseado, adicionalmente se debe programar dicho comportamiento con máquinas de estado para cumplir con el objetivo de ganar el torneo.

# OBJETIVOS

Objetivo general:

Construir un mini sumo-bot para participar en la feria de proyectos organizada por la escuela de ingeniería y conseguir la victoria en esta.

Objetivos específicos:

•Manejo de máquinas de estado para el control y funcionamiento del sumo.

• Manejo de sensores de proximidad y detección de color.

• Uso del módulo Wifi para transmisión de datos en tiempo real a través de Ubidots.

• Uso del USART para el control del módulo Wifi.

• Uso de ADC.

• Uso de PWM

# DESCRIPCION

Se diseñó y construyó un mini sumo-bot, el cual debe participar en la feria de proyectos organizada por la escuela de ingeniería, con el objetivo de ganar en la competencia de sumos. Para ello se diseñaron las piezas del chasis en un programa de CAD llamado FreeCAD, para ser impresas posteriormente en 3D.

Para el circuito se diseñó en altium el prototipo de este, para posteriormente ser impreso por el método de planchado y ácidos con el fin de tener un circuito confiable y fácil de soldar.

Luego de tener las dos partes más importantes como son el chasis y el circuito, se comenzó el montaje de todos los componentes en el PCB impreso, así mismo encajar los motores y sensores en el chasis para tener el prototipo completo para empezar con la programación, la cual debe estar hecha con base en máquinas de estado las cuales deben dar el comportamiento deseado del sumo para dicha competencia y debe tener también la capacidad de transmitir el estado de la batería a través de internet y la plataforma Ubidots, para ser monitoreada y controlada. Por ultimo posee un switch de encendido y apagado, así mismo un pulsador que permite intercambiar los programas a correr, esto con el fin de identificar que comportamiento es más óptimo con respecto al oponente, con el fin de cumplir el objetivo y ganar el torneo.

# DESARROLLO

Para el desarrollo se tuvieron en cuenta varios parámetros, por ejemplo las reglas de la competencia dictan que para este mini sumo-bot las medidas de este no deben superar el 10 \* 10 cm lo que nos deja un margen de 10cm por cada lado para el diseño y acomodamiento del circuito, sensores y motores, la única medida que está sin definir es la altura que puede ser de las dimensiones que se deseen. Teniendo lo anterior en cuenta se procedió a diseñar el chasis final del sumo el cual cuenta con un ancho de 10 cm, un largo de 10cm contando las ruedas, y un alto de aproximadamente 20 cm ya que es en la estructura superior donde reposan el sistema de alimentación, el circuito y los cables de este, dejando el primer nivel únicamente para los sensores y conexiones del PCB a estos. Las piezas del primer nivel se imprimieron en una impresora 3D tipo prusa con un PLA rojo semitransparente el cual hace de nivel inferior y soporte para motores y sensores, el segundo nivel fue impreso en una impresora Ultimaker 2+ con un PLA de color verde este sirviendo de caja contenedora de la batería, cableado y el PCB.

Todas las piezas del chasis fueron diseñadas en un software de CAD llamado FreeCAD, luego exportadas en formato .STL para ser cargadas en el programa controlador de la impresora llamado Repetier-Host, el cual se encarga de convertir el CAD en una serie de coordenadas y control de expulsión del PLA para que este pueda ser controlado por la impresora.

Para el circuito se diseñó un esquemático con las conexiones que finalmente serian exportadas a PCB para ser impresas (ver carpeta de anexos, archivo Schematic Prints.pdf), luego de tener el diseño del esquemático, se procedió a probar esta implementación en una protoboard, para verificar si todo lo realizado hasta el momento era funcional y así poder proceder al siguiente paso para imprimir el circuito. Luego de terminar las pruebas de funcionamiento y verificar que todo estaba conforme con su correcto funcionamiento, se procedió a imprimir el PCB por el método de planchado y ácidos, para finalizar con el montaje de las piezas. Al principio se decidió implementar en una baquela universal ya que el proceso salió fallido y se perdió el proceso realizado, luego de implementarlo en la baquela y tratar de hacer la primera prueba, surgió un problema de conexiones por lo que el microcontrolador se quemó, haciendo que to9cara volver a empezar todo el montaje en una nueva superficie, por lo que se optó por repetir la impresión del PCB como inicialmente se tenía previsto. Luego de realizar el segundo intento con la impresión del PCB, esta resulto tal como se tenía previsto inicialmente por lo que se procedió a colocar los componentes en este PCB y realizar pruebas de cada uno al ser colocado para evitar errores por corto circuito o algún otro problema que pueda llegar a presentarse y nuevamente sea el micro el que se dañe o algún otro componente esencial para su funcionamiento.

Finalizando con el montaje de los componentes, se tuvo dificultades con el diodo zener ya que su comportamiento era anormal, lo cual luego de tratar con el profesor y plantear una solución se llegó a la conclusión de que este comportamiento inusual se debía a que la impedancia del divisor de voltaje al cual estaba ligado era superior a la impedancia del diodo haciendo que este se comportara de manera anormal, para solucionar esto se bajó el factor de las resistencias manteniendo intacto el valor de estas y adicionalmente aumentando la capacidad de vatios que estas soportan. Finalizando y corrigiendo estos problemas, se procedió con la parte del puente h para el control de los motores, así que luego de ser soldado y probado y constatar que este funciona de manera correcta, luego de tener el puente h montado y comprobar que no existan cortos ni malos montajes se procedió a probar con una serie de voltaje se prueba generados desde una fuente para ver cómo se comporta dentro del PCB. Al ver que este estaba funcionando de manera correcta, se procedió a terminar de montar el resto de componentes y poner un programa de prueba para el sentido de giro de los motores, el cual funciono de manera correcta, para así poder proseguir con el siguiente proceso.

En cuanto al montaje de los sensores infrarrojos Sharp, estos cuentan con un conector el cual facilita el uso de estos, por lo cual fue fácil conectarlos al PCB, bastando con una simple soldadura de cables. En cuanto a los sensores ubicados en la parte inferior al contar con 4 pines independientes, hubo más complicación a la hora de soldarlos ya que se debió agregar un cable extra desde el pin hasta la PCB, ya que por sí solos estos no alcanzan a llegar para ser soldados. Tanto los sensores Sharp como los anteriores funcionan de manera análoga por lo cual en la programación se debió utilizar múltiples canales de lectura de estos para aprovechar su funcionamiento para el sistema.

Para la programación, siendo esta en base a máquinas de estado primero se cargó un programa piloto la el control de los motores, de esta manera verificando su funcionamiento, dicho programa piloto mediante máquinas de est5ado, genera lo necesario para poder ir hacia adelante, izquierda, derecha y atrás, así mismo el estado de avanzar y de frenado, para el control básico de los motores así verificando que el circuito y todos sus componentes funcionan de manera apropiada. Luego de esto se procedió a terminar las máquinas de estado para los sensores e incluir las ya programas para los estados de giro del motor.

Por último, se programó el módulo wifi para la trasmisión de datos del estado de la batería, a través de Ubidots, plataforma usada para proyectos de internet de las cosas, la cual nos permite tener control y transmisión de esto.

# CONCLUSIONES

# Anexos