

13/03/2016



# VEX- U COMPETITION

*SPAIN - Barcelona*



Bilal Hassan, Cristian Tudela, Albert Olivé, Mohammed Ayoub

# VEX- U COMPETITION

*SPAIN - Barcelona*

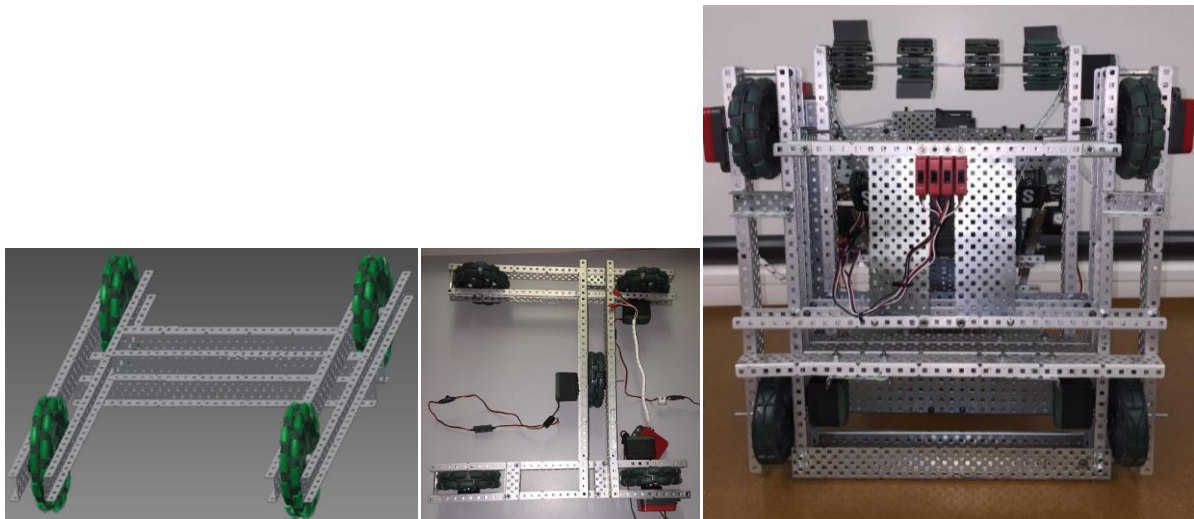
## Introducción

---

El robot que presentamos a la competición nacional ha sido diseñado desde cero. Hemos usado la herramienta Autodesk Inventor 2014.

### La base

En primer lugar se diseñó la base y se probaron los diferentes tipos de ruedas, en nuestro diseño se ha optado por dos ruedas motrices y dos omnidireccionales de 4 pulgadas. Usamos para el desplazamiento dos motores colocados en las ruedas traseras. El chasis escogido es en H mayoritariamente de aluminio para hacerlo más ligero.



Se probaron varias configuraciones, todas las ruedas omnidireccionales. Incluso una quinta rueda omnidireccional en posición perpendicular para realizar movimientos laterales, pero la falta de precisión en el movimiento de la fase autónoma nos hizo plantear el uso de unas ruedas motrices no omnidireccionales y extraer la quinta rueda.

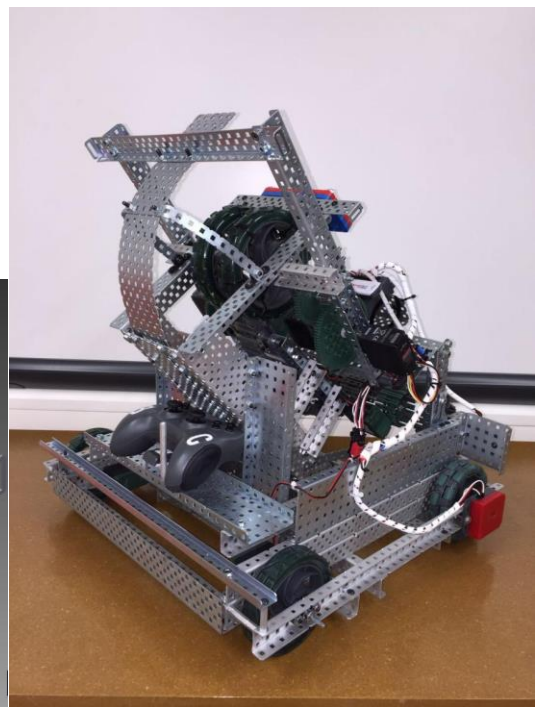
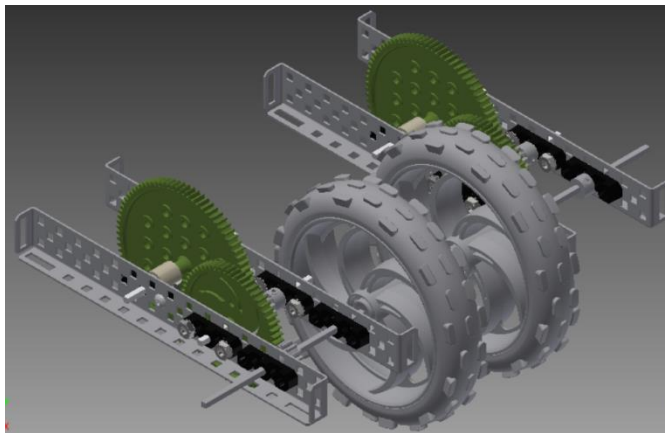
Para controlar el desplazamiento se han usado dos Encoders colocados en las ruedas delanteras para evitar errores de desplazamiento.

## El lanzador

El lanzador ha sido uno de los procesos más complejos, se han probado diferentes configuraciones para conseguir tirar desde base.

Se observaron los mejores resultados al usar una secuencia de engranajes 7:1 \* 5:1 proporcionando en teoría hasta 3500 revoluciones por minuto. Impulsados por 4 motores de 100 RPM y el uso de dos ruedas de 5 pulgadas hace que las pelotas lleguen con facilidad y ángulo adecuado para encestar.

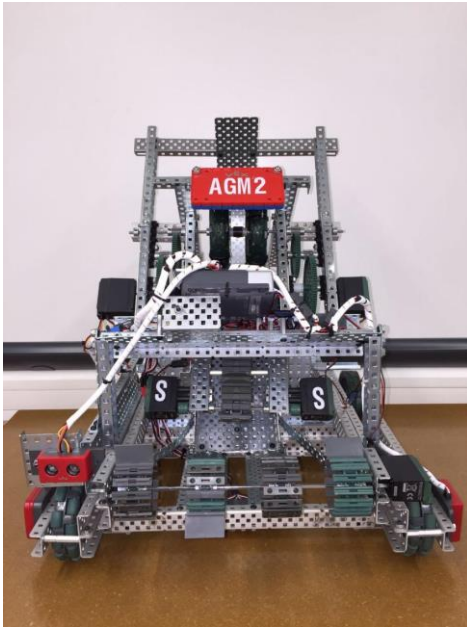
Se realizó el diseño con el software Autodesk Inventor y se ajustó en la realidad para minimizar el rozamiento.



Para controlar la velocidad del lanzador usamos un motor con encoder integrado. También se han colocado durante las pruebas encoders en las ruedas de 5 pulgadas que lanzan (las velocidades necesarias para encestar en el campo son de 22 a 29 vueltas por segundo).

## El recogedor de pelotas delantero

Se ha realizado un recogedor de pelotas delantero que funciona con un motor a 100 RPM. Permite coger las pelotas que están debajo a mucha velocidad e impide que una vez dentro salgan.



## Alineador de pelotas ascendente

Se ha realizado un alineador de pelotas ascendente que permiten hacer llegar un máximo de 3 pelotas al lanzador, es muy rápido ya que se han introducido los engranajes rápidos que permiten una velocidad de 160 RMP, logrando lanzar una pelota por segundo. Se han colocado unas etiquetas con S en los motores para diferenciarlos.

Se colocan dos para tener la fuerza suficiente al mover las tres pelotas y evitar que se sature.

## Sensórica

### Encoders

Se han colocado 3 encoders 2 en las ruedas delanteras para saber que distancia estamos recorriendo y ser más precisos. Con dos canales de salida del codificador se puede medir tanto la posición y la dirección de rotación de un eje Esto nos permitirá calcular la velocidad del eje , así como la distancia que ha recorrido estas son mediciones muy útiles para la programación autónoma.

Tenemos un 3er encoder integrado en uno de los motores del lanzador para tener una velocidad exacta en el lanzador y asegurarnos que entran la mayoría de las pelotas

- Medida angular del viaje.
- Determinar sentido de giro.
- Calcular velocidad del eje.
- Calcular distancia recorrida.
- Aumentar el control de navegación.

## Sonar

Para detectar obstáculos en su camino mediante la utilización de la propagación de ondas sonoras de alta frecuencia. El sensor emite una onda de sonido de 40 kHz, que rebota en una superficie reflectante y vuelve al sensor. Luego, utilizando la cantidad de tiempo que tarda la onda para volver al sensor, la distancia al objeto se puede calcular

Tenemos un sonar para evitar chocar con otros oponentes y acercarnos a la canasta en modo autónomo y quedarnos a la distancia exacta.

## Seguidor de línea

El robot dispone de 4 sensores de línea que nos son muy útiles para poder avanzar encarados a la canasta y poder así, ir a recoger pelotas a la vez que lanzamos las que ya habíamos recogido.

## Programación

A medida que perfeccionábamos el robot fuimos más rigurosos con la programación.

Empezamos programando utilizando las funciones de tiempo y cambiando la potencia de los motores. Obtuvimos resultados no muy precisos y decidimos programar los sensores de los que disponíamos. Primero programamos los encoders para poder ser precisos en los giros y las distancias recorridas. Nos encontramos con algún problema como la inercia del robot en los movimientos, pero lo solucionamos con un poco de ingenio:

```
// funcion para realizar un giro a la izquierda de X grados

void giro_izquierda(int grados)
{
    SensorValue[rightEncoder] = 0;
    while(SensorValue[rightEncoder] < grados)
    {
        motor[backRightMotor] = 50;
        motor[backLeftMotor] = -50;
    }
    motor[backRightMotor] = -10;
    motor[backLeftMotor] = 10;
    wait1Msec(200);
}
```

```
motor[backRightMotor] = 0;
motor[backLeftMotor] = 0;
}
```

También disponíamos de sensores de línea, los programamos para poder evitar cualquier error de dirección al avanzar encarados hacia la canasta.

Llegados a este punto éramos capaces de coger 3 montones de pelotas más las de precarga y encestar un gran porcentaje. Aún así, podíamos optimizar mucho más el tiempo si supiésemos los momentos o sitios exactos en los que maniobrar y ser más precisos al ajustar la velocidad del lanzador.

Al montar y programar el sonar éramos capaces de parar donde nos convenía para no fallar en los lanzamientos a la par de evitar posibles colisiones con los contrincantes al movernos por el campo.

Con todo esto, el robot era capaz de encestar la mayoría de las pelotas que recogía. Aunque en algunas pruebas fallaba debido a la carga de las baterías. Si las baterías estaban al 100% las inercias y potencias de los motores aumentaban considerablemente y si perdían demasiada carga los motores no rendían suficiente. Encargamos un motor con encoder integrado para montarlo en el lanzador y creamos la siguiente función para poder ajustar estos excesos y carencias de potencia.

```
// funcion que nos da el error del lanzador
int ajustlanzador()
{
    // Ajustamos la velocidad para iniciar lanzamientos
    int gradossegs=0;
    int error=0;

    SensorValue[L4MEncoder] = 0;

    wait1Msec(500);
    gradossegs=SensorValue[L4MEncoder];

    if(340<gradossegs) error=-4;
    else if(320<gradossegs<341) error=-2;
    else if(310<gradossegs<321) error=0;
    else if(gradossegs<311) error=2;

    return error;
}
```

Hicimos muchas pruebas con tal de ver las vueltas que hacía el encoder integrado en medio segundo y poder definir los intervalos de error.

Así aumentamos el rendimiento de las baterías pudiendo usarlas durante más tiempo y aprovecharlas mejor.



## Conclusiones

Montar y programar un robot desde cero ha sido un proyecto complejo en el que la dedicación y empeño han sido cruciales para poder avanzar y mejorar. Poder trabajar en equipo ha aumentado la capacidad de cooperación y adaptación a la hora de desarrollar el proyecto, así como aprender de los compañeros. Al tener total libertad a la hora de diseñar el robot nos hemos encontrado con varios problemas (ajustar y calibrar todos los motores y sensores, desmontar y cambiar el lanzador porque no nos gustaba, etc) que entre todos hemos sido capaces de superarlos y obtener cierta experiencia de cara al futuro.

Quiero agradecer a todo el equipo de VEX por el apoyo, energía y constancia que han dado desde que decidimos participar en la competición. También quiero agradecer a la escuela A.G. Mundet por la gran inversión que nos ha proporcionado dotándonos del material y las instalaciones necesarias para poder participar con tres equipos en la competición.

Estoy muy satisfecho con los resultados obtenidos, espero que nuestro esfuerzo e ingenio se vea reflejado en la competición de España.