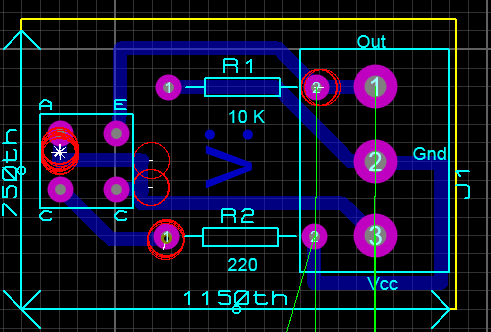
**AVANCES**

como se había pretendido usamos el circuito mostrado en la semana 1, ese circuito constaba de un cny70 y 2 resistencias de 10K y de 220 respectivamente, se diseñó en Ares el siguiente esquema de PCB:



*Fig. 1 esquema de PCB en Ares*

en la fig.1 está el diseño del circuito impreso que será pequeño y más compacto en este se detallan puntos rojos de error, esto es debido a que el cny70 no estaba para ser editado en Ares por lo tanto se hizo a medida y con herramientas propias del programa.

En la fig. 2 el circuito está soldado y probado, se usaron resistencias de 15k en vez de las de 10k con motivo de mermar un poco la sensibilidad del módulo, esta cara del circuito es la que irá abajo al suelo.

En la fig. 3 se demuestra el ruteamiento de las líneas conectoras y un sello personal agregado que le da más originalidad.



*Fig. 2 circuito impreso*



*Fig. 3 Reverso del circuito*



*Fig 4. maqueta del carro*



*Fig 5. maqueta del carro en otra perspectiva*

En la fig.4 y fig. 5 se detalla el marco del carro en donde se puede ver a simple vista el tipo de motor a usar, el controlador que es un puente H ‘L298’ y el arduino que lo cambiamos de un mega a un nano por cuestiones de espacio y peso.

a continuación se presenta el código que estamos implementando y que aún está sujeto a cambios

**// Pines analogos**

**int sensor1 = 0;**

**int sensor2 = 1;**

**int sensor3 = 2;**

**int k\_proporcional = 3;**

**int k\_integral = 4;**

**int k\_derivativa = 5;**

**#define der1 3**

**#define der2 4**

**#define derpwm 5**

**#define iz1 12**

**#define iz2 13**

**#define izpwm 11**

**#define motorPower 8**

**float proporcional = 0;**

**float proporcional\_pasado = 0;**

**int posicion = 0;**

**int lastPosicion = 0;**

**float integral = 0;**

**float derivativo = 0;**

**float valor\_pwm = 0;**

**int maxvelder=230;**

**int maxveliz=230;**

**int basevelder=150;**

**int baseveliz=150;**

**int velder=0;**

**int veliz=0;**

**void setup() {**

**delay(2000);**

**Serial.begin(9600);**

**pinMode(der1, OUTPUT);**

**pinMode(der2, OUTPUT);**

**pinMode(derpwm, OUTPUT);**

**pinMode(iz1, OUTPUT);**

**pinMode(iz2, OUTPUT);**

**pinMode(izpwm, OUTPUT);**

**pinMode(motorPower, OUTPUT);**

**}**

**void loop() {**

**int s1 = analogRead(sensor1);**

**int s2 = analogRead(sensor2);**

**int s3 = analogRead(sensor3);**

**float kp=analogRead(k\_proporcional);**

**float ki=analogRead(k\_integral);**

**float kd=analogRead(k\_derivativa);**

**int \_s1 = 0;**

**int \_s2 = 0;**

**int \_s3 = 0;**

**// Calibracion de los sensores**

**if(s1>270){**

**\_s1=1;**

**}**

**if(s2>360){**

**\_s2=1;**

**}**

**if(s3>350){**

**\_s3=1;**

**}**

**posicion = (1\*\_s1)+(2\*\_s2)+(4\*\_s3); // Convertimos los numeros binarios obtenidos de los sensores en decimales**

**if(posicion <= 6){**

**lastPosicion = posicion;**

**}**

**// Calculamos los valores proporcional, integral y derivativo**

**proporcional = (posicion - 9);**

**integral = (integral + proporcional\_pasado);**

**derivativo = (proporcional - proporcional\_pasado);**

**int error=posicion-9;**

**// Valor obtenidos del PID**

**valor\_pwm = (kp\*proporcional)+(ki\*integral)+(kd\*derivativo);**

**// Limitamos el valor de la integral**

**if(integral>200){**

**integral=200;**

**}**

**if(integral<-200){**

**integral=-200;**

**}**

**// Almacenamos el ultimo valor de proporcional**

**proporcional\_pasado = proporcional;**

**if (error>0)**

**{**

**int velder = basevelder + valor\_pwm;**

**int veliz = baseveliz - valor\_pwm;**

**}**

**if (error<0)**

**{**

**int velder = basevelder - valor\_pwm;**

**int veliz = baseveliz + valor\_pwm;**

**}**

**if (velder > maxvelder ) velder = maxvelder;**

**if (veliz > maxveliz ) veliz = maxveliz;**

**if (velder < 0) velder = 0;**

**if (veliz < 0) velder = 0;**

**digitalWrite(motorPower, HIGH);**

**digitalWrite(der1, HIGH);**

**digitalWrite(der2, LOW);**

**analogWrite(derpwm, velder);**

**digitalWrite(iz1, HIGH);**

**digitalWrite(iz2, LOW);**

**analogWrite(izpwm, veliz);**

**}**