



REV-CONTROL

Bitácora de Campo (Registro de trabajo)

Curso: 7° 1° Aviónica

Comisión: C

Introducción

Esta bitácora de campo tiene como propósito documentar de manera detallada y sistemática todas las actividades relacionadas con la implementación, pruebas y ajustes del sistema de control de revoluciones (**Rev Control**). Aquí se registran observaciones, resultados de pruebas, modificaciones realizadas al sistema, así como cualquier incidencia o ajuste requerido durante las distintas etapas del proyecto.

El objetivo principal de este registro es servir como una herramienta de referencia para evaluar el desempeño del sistema, garantizar la trazabilidad de los cambios realizados y facilitar futuras mejoras o diagnósticos. La información contenida en esta carpeta de campo también será útil para documentar buenas prácticas y generar una base de conocimientos que pueda ser reutilizada en proyectos similares.

Estructura de la bitácora:

1. **Descripción del sistema:** Breve descripción del objetivo y funcionamiento de Rev Control.
2. **Planificación y pruebas:** Registro de fechas, equipos utilizados y procedimientos de prueba.
3. **Observaciones de campo:** Análisis de resultados, incidencias y comportamiento del sistema.
4. **Acciones correctivas:** Ajustes realizados y su impacto en el desempeño.
5. **Conclusiones y próximos pasos:** Resumen de hallazgos y propuestas de mejora.

Esta bitácora constituye un recurso clave para asegurar que el proyecto cumpla con los objetivos planteados y facilite un control de calidad óptimo.

En esta carpeta vamos a guardar y recopilar la información del progreso del proyecto y distintas problemáticas o datos curiosos que nos lleguemos a encontrar en estos meses de trabajo. (3 de Marzo)

Integrantes:

- Acosta, Gonzalo
- Ibaceta, Tadeo
- Flores, Santiago
- Martinez, Marcos
- He, Leonardo
- Quintero, Juan
- Alfaro, Lautaro

Cronología de Trabajo:

• Inicio de Marzo:

Iniciamos las clases y empezamos a armar los grupos, en la comisión C quedamos 2 grupos de 7 personas. Planteamos varias ideas de proyecto, 3 para ser exacto:

1. Un generador de Energía mareométrica
2. BioGás (Genera energía con gas metano)
3. **Medidor de parámetros para banco de prueba de motor**
(Proyecto elegido)

Al elegir nuestro proyecto nos topamos con varias problemáticas, principalmente el hecho de que trabajamos con algo fuera de nuestro ambiente principal entre otros temas (económico por ejemplo).

Pero nosotros tomamos la decisión de colaborar con un grupo de 7° 1° Aeronáutica Comisión C y este proyecto se terminó convirtiendo en un “proyecto colaborativo”.

Fuimos a consultar con profesores de aeronáutica nuestras dudas y en particular la viabilidad del mismo, su funcionamiento y electrónica involucrada.

También tomamos la decisión de usar un LPC845 como microprocesador

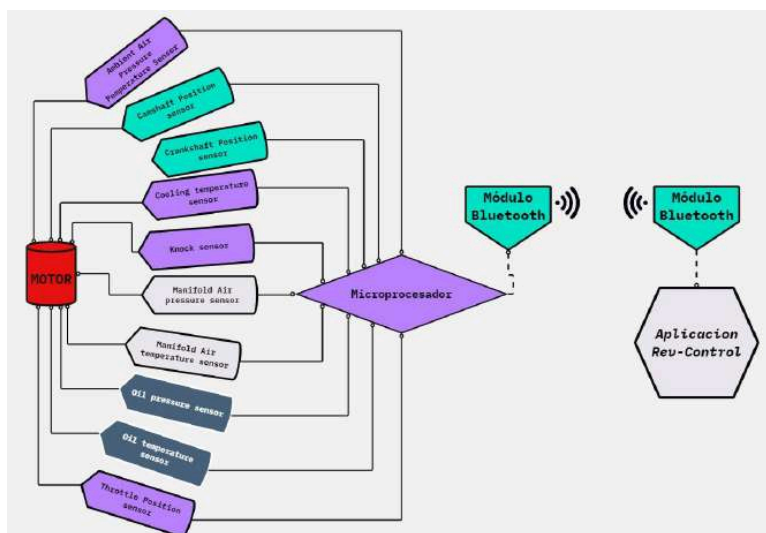
Empezamos con la planificación de trabajo y horarios para el proyecto.



Nosotros también buscábamos un motor para poder probar en el proyecto y que el grupo de Aeronautica trabajan. Se eligió un motor de 4 tiempos Rotax ULS 912:



Este será puesto en un banco con su helice incluida. Será donado a la escuela.



(Diagrama conceptual de funcionamiento)

● Finales de Marzo:

Concretamos el diseño y configuración del proyecto. En un principio nos íbamos a inspirar en un medidor de parámetros de avión (Garmin):



Pero quisimos compaginarlo todo en otro formato un poco más cómodo para nosotros y más general para un taller de autos y sobre todo “portátil”. Así que con profesores de Aeronáutica y el profesor Julián Scicolone formamos una segunda idea que sería la utilizada para el prototipo final:



Un diseño similar a este sería con el que presentamos el proyecto (Monitor arriba y el resto abajo). También estamos buscando los sensores e iniciamos la programación y los parámetros que mediaríamos.

Armamos página de Instagram del proyecto y también confeccionamos logotipo para el proyecto:





logotipo)

(Primer

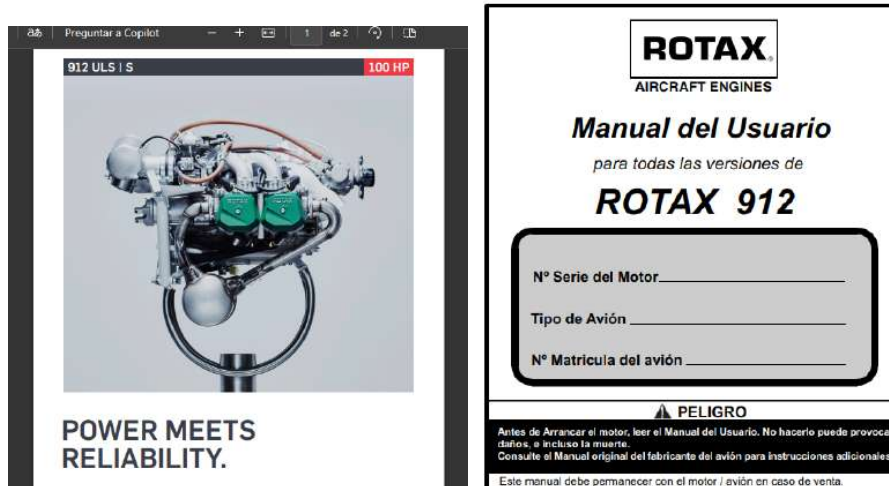


(2° logotipo y final)



- **Inicios de Abril:**

Estuvimos investigando los manuales de mantenimiento y de usuario del motor que íbamos a usar, para saber alrededor de qué valores se manejaba el motor en si.



<

También conseguimos por medios propios un maletín estilo Ambil por medios propios para usar de carcasa:



Hicimos página de linktree con acceso al Gmail y a la pagina de Instagram Imprimimos Stickers con código QR y pusimos algunos alrededor de Aviónica:



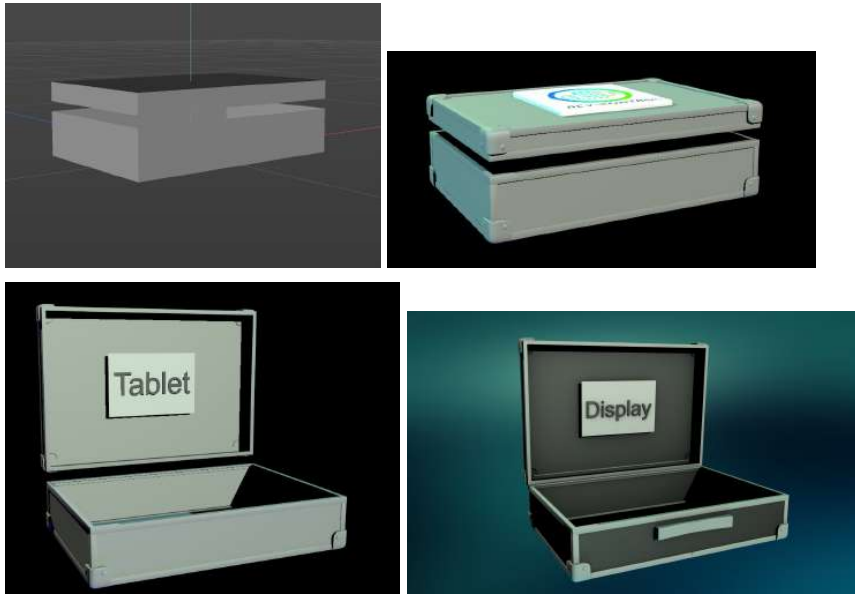
También estuvimos probando las funciones del LPC 845 con ayuda del profesor Fabrizio Carlassara (A mediados de Abril) y una semana después empezamos a iniciar el código para sensar la temperatura.

También trajimos una Tablet para probar si podíamos usar pero no encendía y no era la versión que necesitábamos para correr el programa que recibiría la información de los sensores en el maletín.



El profesor Julián Scicolone Nos ofreció una Tablet el mismo para poder usar, mientras que para la aplicación usaremos un celular normal.

Iniciamos el diseño de un modelo 3D del proyecto



- **Finales de abril:**

Conseguimos un LPC845 propio para poder usar (Al anterior era prestado)

Mandamos mails a distintas empresas, arreglamos una parte del Linktree, Speech Elevator, más contenido para las redes y el inicio de las historias de usuario.

- **Inicio de Mayo (03/05/2024 - 15/05/2024):**

Terminamos las historias de usuario y reorganizamos el tablero de Trello:

Historias de Usuario de Rev-Control

• Carta:

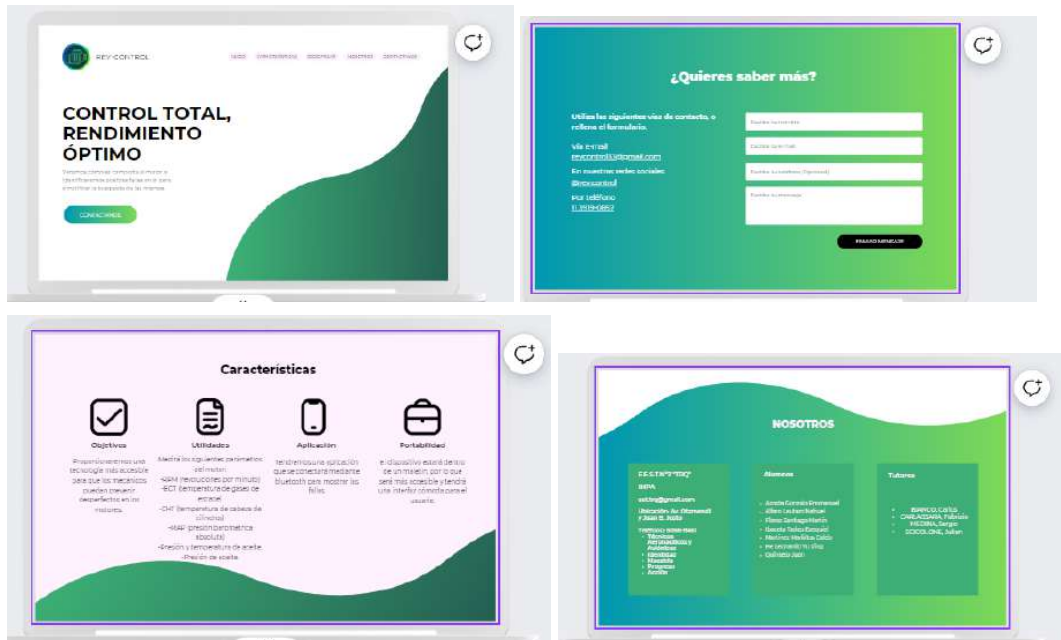
1. HU001

Yo como: Patrocinador
Quiero: Poder conocer más información sobre el grupo de Rev-Control
Para: saber quienes lo integran, de qué se trata y cuáles su objetivo.

Criterio de aceptación:

- Armar una página web con los links necesarios para

También conceptos de la página Web:

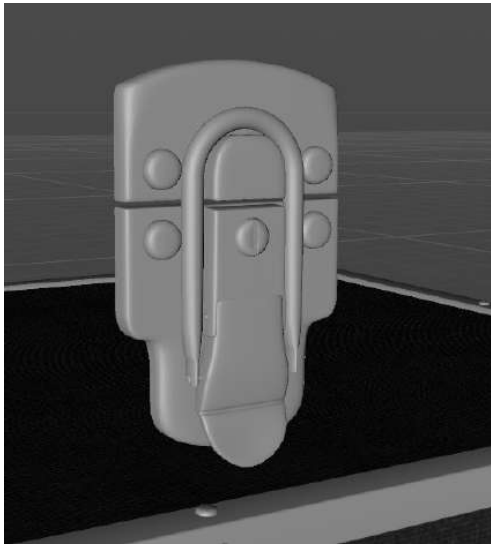


También agregamos 2 materiales para la lista de compras (

721_2024_RevControl), placas de cobre y 4 sensores con un módulo digital para poder probar.

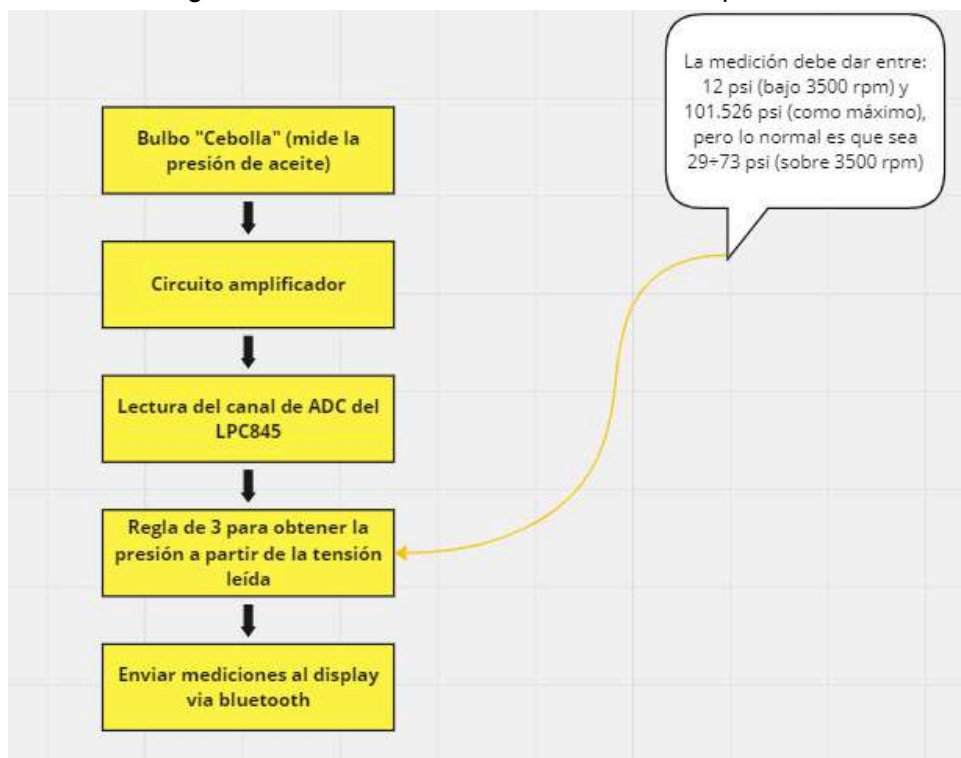
Avanzamos con el modelo 3D y en breve ya tendremos la futura presentación del mismo (El interior aún no está diseñado)



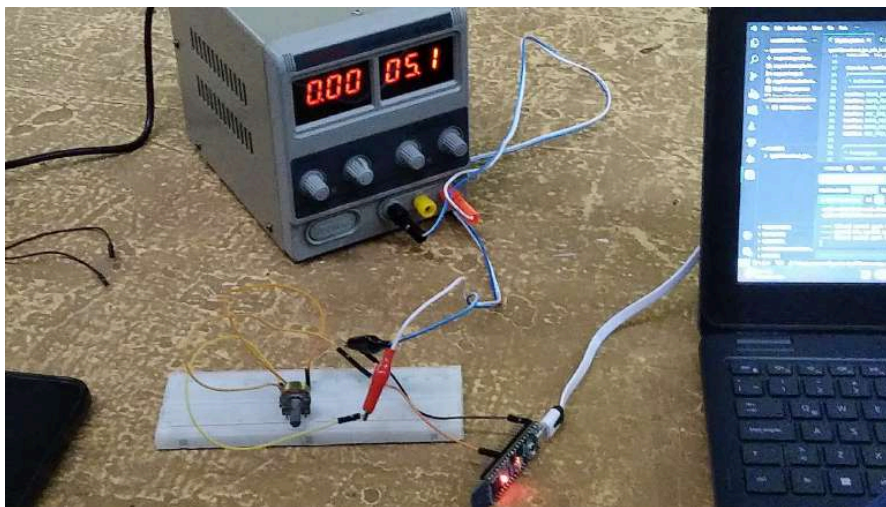


- **Finales de Mayo e inicios de Junio (15/05/2024 - 06/06/2024):**

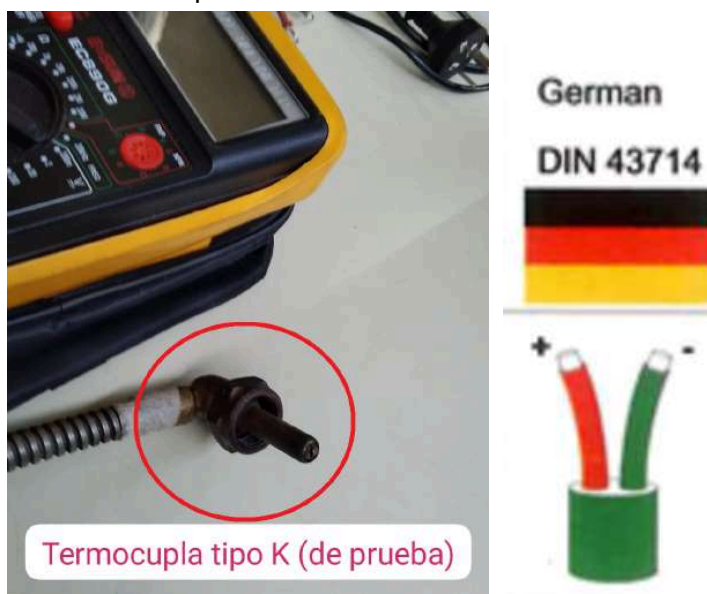
Se hizo un diagrama básico de como seria el sensor de presion de aceite:



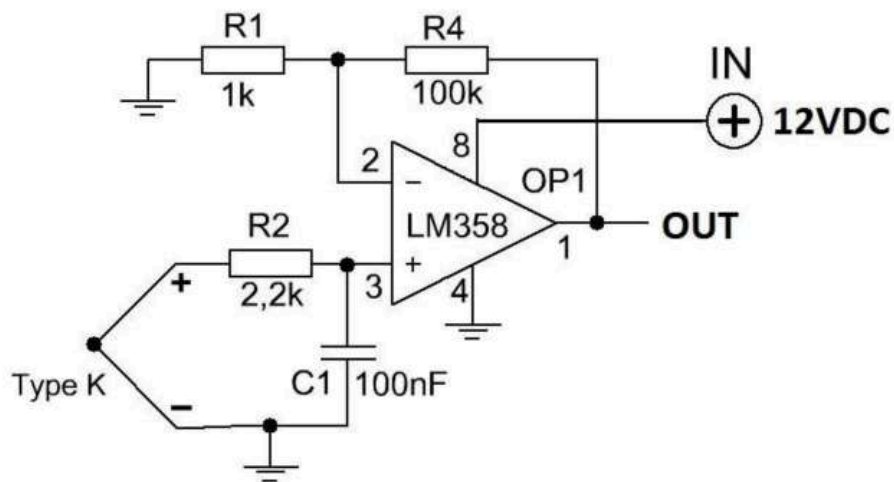
También terminamos de soldar todos los pines del Micro e hicimos pruebas con el potenciómetro incorporado en el LPC845 para probar la lectura del ADC (Lo leía bien), luego probamos con un potenciómetro externo pero el adc no lograba leer bien:



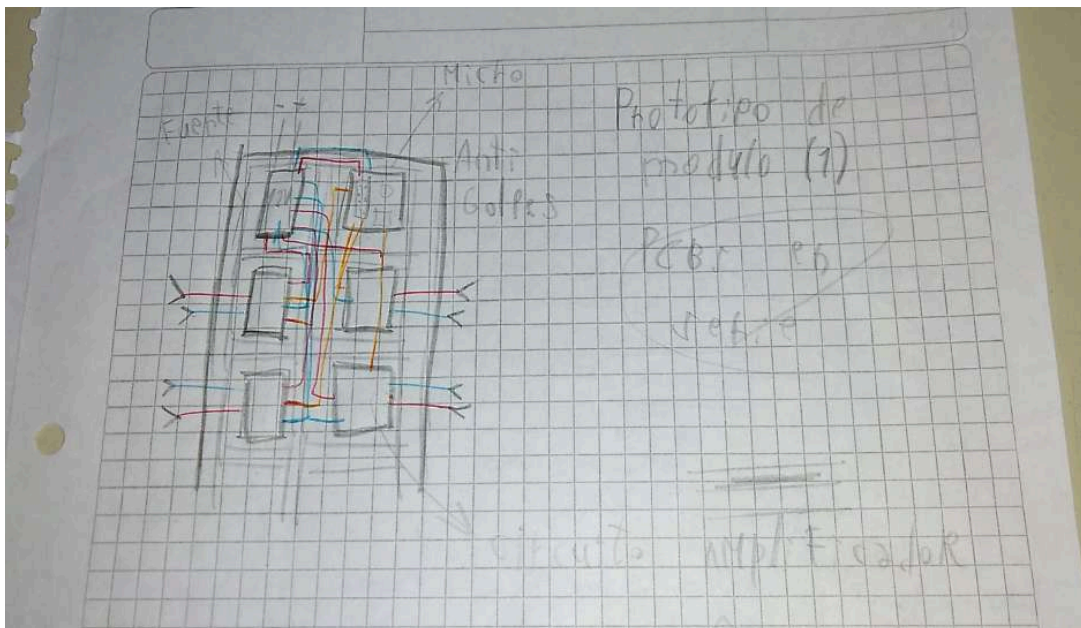
El día lunes 03/06/2024 nos prestaron de aeronáutica una termocupla tipo K (Alemana, con cable rojo y verde) para hacer el mismo tipo de pruebas que hicimos con el potenciómetro:



Armamos un circuito amplificador con un LM358 para que la señal pueda ser leída en el ADC (Nos dió 3V amplificado Aproximadamente):



También planteamos una nueva idea (El 05/06/2024) para la distribución de los sensores en el módulo que estará al lado del motor.



Este croquis es un planteo de los que sería el módulo de los amplificadores de los distintos sensores que vayamos a usar, esto sería para tener una comunicación más cómoda con el microprocesador. Los PCBs estarían en forma de "fichas" para que puedan ser puestos y extraídos con facilidad (Serían como "mini módulos") y en el caso de que no lleguemos a terminar

todos los módulos, estarán estos espacios disponibles para que en otro año los alumnos puedan continuarlo o incluso ampliarlo.

- **Finales de Junio e inicios de Julio (06/06/2024 - 05/07/2024):**

En esta fecha (06/06) hicimos pruebas de cómo funcionaban y poder plantear circuitos de amplificación de los mismos. Estos nos los prestaron de aeronáutica.:



Les mandamos aire comprimido a algunos para ver cómo iban reaccionando.

El 12/06-19/06 se arregló y completó para que se puedan leer los canales de ADC del micro simultáneamente.

```
//-----//
// defines
//-----//
#define ADC0_CH1    1 //Temp. cabeza de cilindro (CHT)
#define ADC0_CH2    2 //Temp. cabeza de cilindro (CHT)
#define ADC0_CH3    3 //Temp. cabeza de cilindro (CHT)
#define ADC0_CH4    4 //Temp. cabeza de cilindro (CHT)
#define ADC0_CH5    5 //Temp. gases de escape (EGT)
#define ADC0_CH6    6 //Temp. de aceite
#define ADC0_CH7    7 //Presión de aceite
#define ADC0_CH8    8 //Presión MAP
#define ADC0_CH9    9 //RPM
#define ADC_FULL_RANGE 4095U // Rango del ADC
//-----//
// Variables
//-----//
long i;
int r; //contador
uint32_t count_mseg;
adc_result_info_t adcResultInfoStruct;
uint32_t frequency;
uint8_t adc_conv_completo, a = 0;
const float referenceVoltage = 3.3; // Voltaje de referencia del ADC en voltios
uint16_t adc_channel[9] = {ADC0_CH1, ADC0_CH2, ADC0_CH3, ADC0_CH4, ADC0_CH5, ADC0_CH6, ADC0_CH7, ADC0_CH8, ADC0_CH9}; //array de canales
uint32_t channel_result[9] = {}; //array de resultados de los canales
uint16_t temperatura[9] = {}; //valores de temperatura almacenados
```



```
void ADC0_SEQA_IRQHandler(void)
{
    if (kADC_ConvSeqAInterruptFlag == (kADC_ConvSeqAInterruptFlag & ADC_GetStatusFlags(ADC0)))
    {
        for (r = 0; r < 9; r++){
            ADC_GetChannelConversionResult(ADC0, adc_channel[r], &adcResultInfoStruct);
            channel_result[r] = adcResultInfoStruct.result;
        }

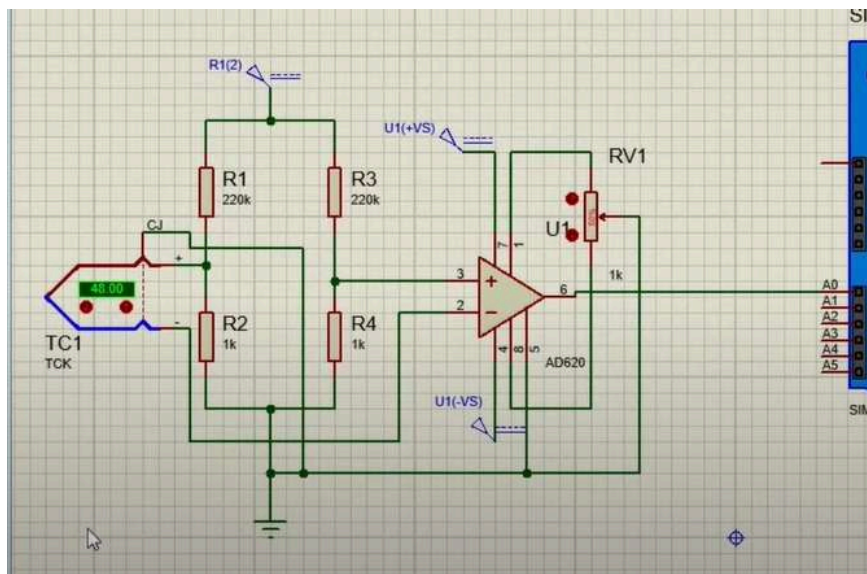
        ADC_ClearStatusFlags(ADC0, kADC_ConvSeqAInterruptFlag);
        adc_conv_complete = TRUE;
    }
}
```

Y también de la medición de temperatura:

```
while(1) {
    if(adc_conv_complete == TRUE){
        for (r = 0; r < 6; r++){
            temperature[r] = channel_result[r] * (referenceVoltage / ADC_FULL_RANGE)
            if (r >= 0 && r <= 3){
                printf("Temperatura de cabeza de cilindro: %ld\r\n °C, y su valor de ADC es: %ld\r\n", temperature[r], channel_result[r])
            }
            elif (r == 4){
                printf("Temperatura de gases de escape: %ld\r\n °C, y su valor de ADC es: %ld\r\n", temperature[4], channel_result[4])
            }
            elif (r == 5){
                printf("Temperatura de aceite: %ld\r\n °C, y su valor de ADC es: %ld\r\n", temperature[5], channel_result[5])
            }
        }
    }
}
```

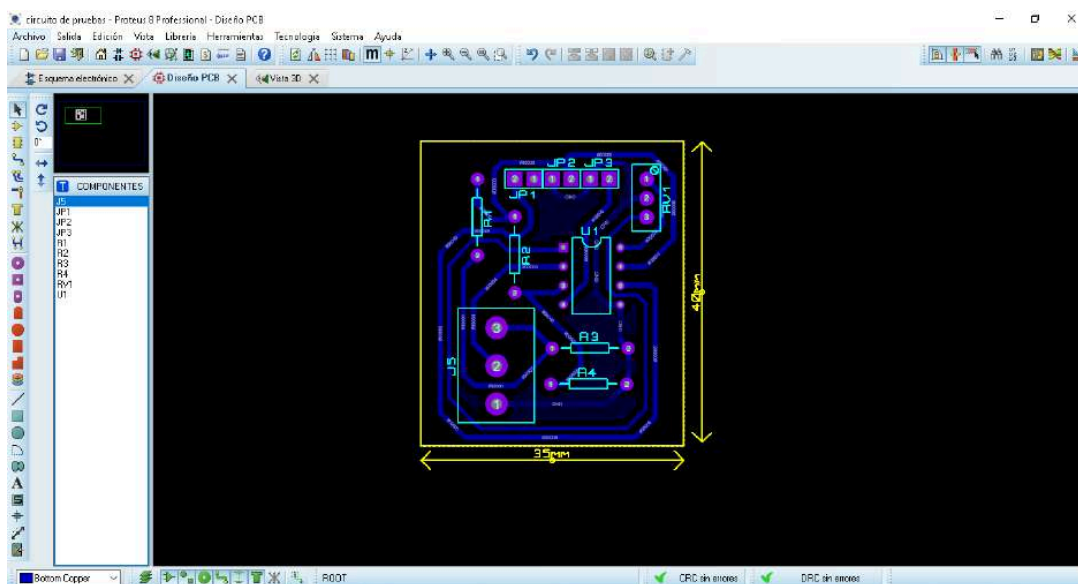
El 28/06 nos pusimos a compaginar otro diseño y final del circuito de Amplificación de la Termocupla:

Diagrama:



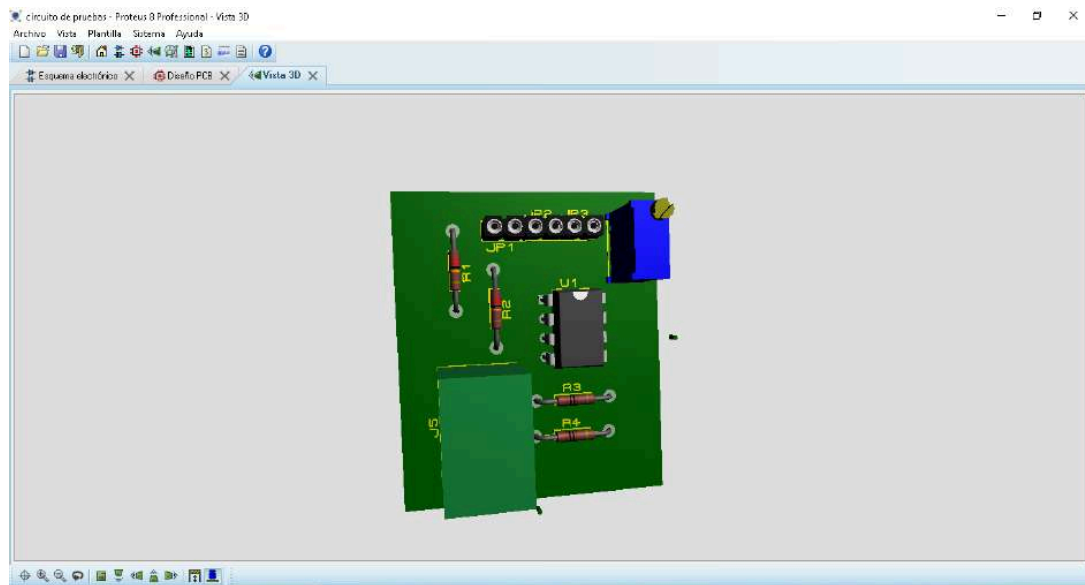
El 03/07 se estaba empezando a programar la página web usando el diseño previo y se armó el diseño de la PCB del amplificador de termocupla y un modelo 3D del mismo:

Diseño PCB:

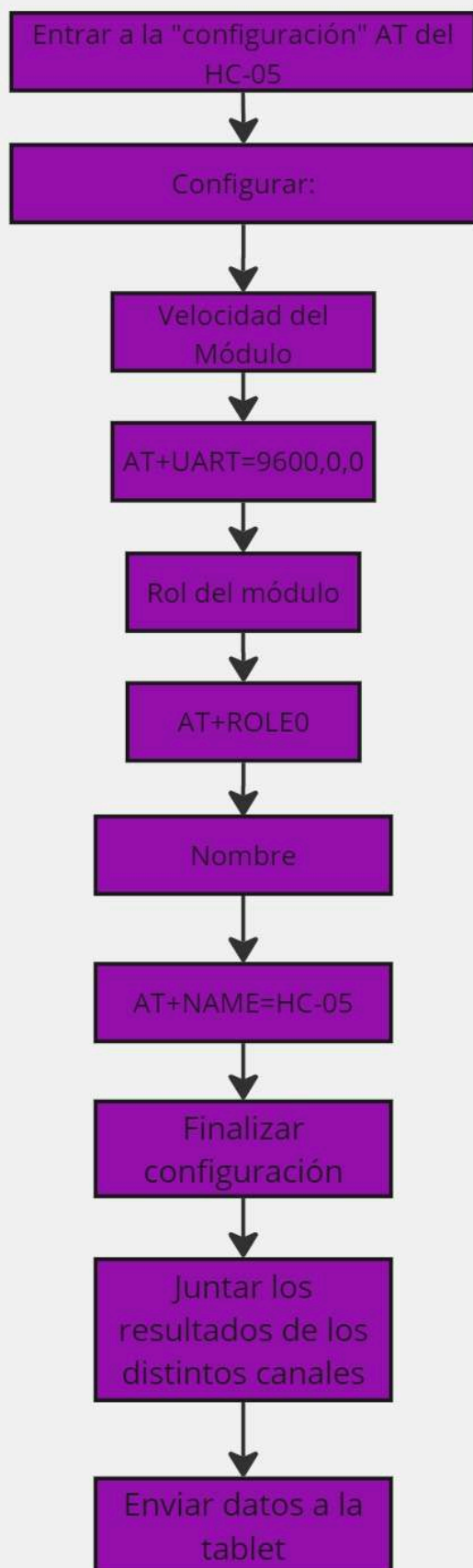


Diseño 3D:





También empezamos a programar la extensión bluetooth para el microprocesador con un diagrama de bloques del mismo:



El 5/7 trajimos material para empezar a construir el módulo con placas de acrílico, las siguientes semanas o después del receso de vacaciones de invierno seguiremos:



- **Semana 29/07-02/08:**

El motor Rotax llegó durante de la semana de vacaciones de invierno:



Durante esta semana hicimos más en conjunto el trabajo con el grupo de Aeronáutica, la bancada sigue en construcción, pero se usará una cabina móvil donde estará la banca del motor y también dentro de la cabina el espacio para meter Rev-Control.

Por cuestión de seguridad pedimos funda de malla para proteger los cables que vayan del módulo al motor en sí, por ende el módulo de amplificadores estará dentro de la cabina. En futuras configuraciones se podría poner en otro lugar, pero esta es la mejor alternativa.

También estamos en busca de una batería para alimentar el maletín (y el módulo de amplificación), ya que todo el banco no estará conectado a la corriente por cuestión de seguridad.

Se sigue revisando el tema de la aplicación y un “plan B” para reemplazar la tablet, que será directamente

Ya que ya tenemos el motor para trabajarlo físicamente, estamos buscando las termocuplas respectivas para las 4 cabezas de cilindro:

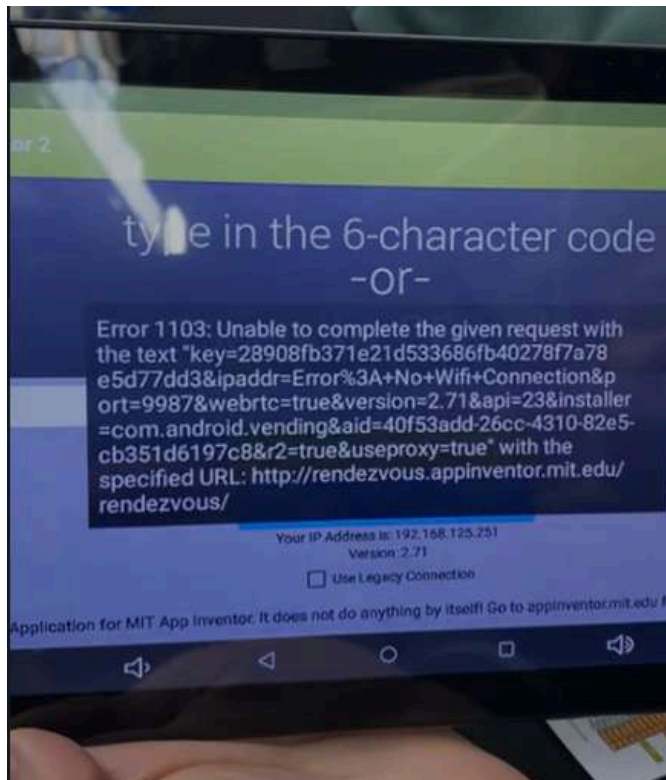


- **Semana 5/10-12/08:**

Durante esta semana estuvimos buscando obtener una batería 12v (preferiblemente con el mayor amperaje posible para poder usarlo durante un tiempo mas prolongado de tiempo en las operaciones).

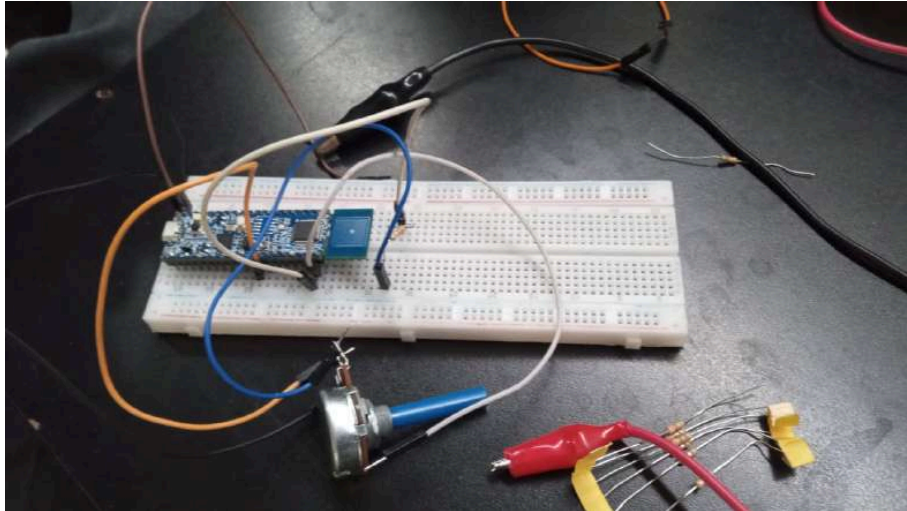


Y también estuvimos probando una tablet que nos dieron para mostrar la interfaz y lamentablemente no obtuvimos los resultados esperados

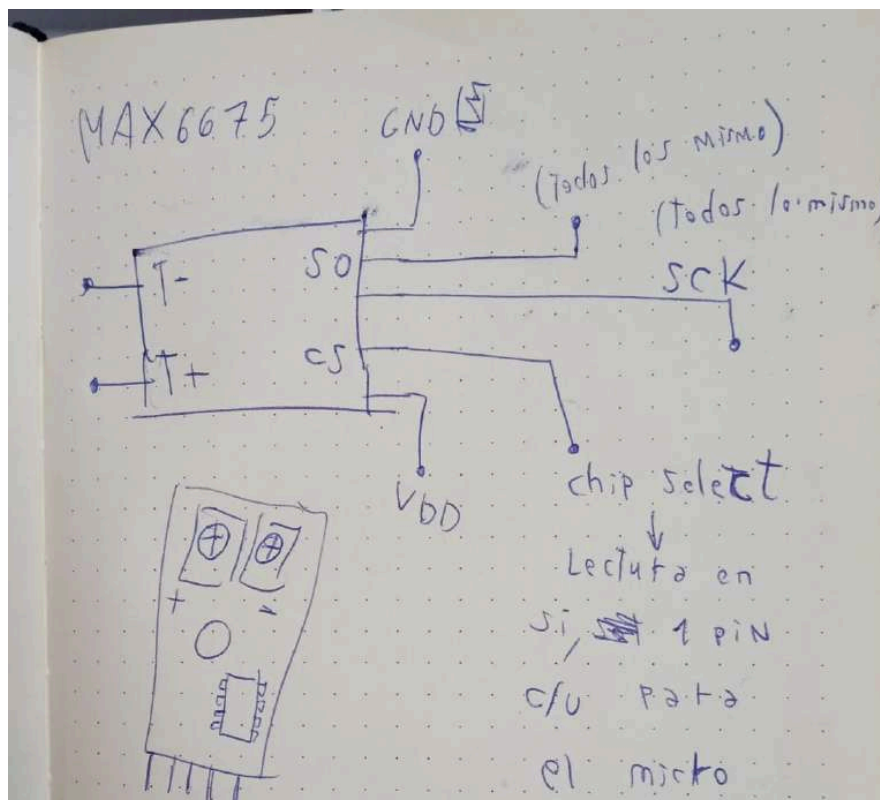


Esto fue principalmente a los problemas de conexión que tenía esta tablet y con la versión de android con la que trabajaba. Esto nos llevó a buscar un método de comunicación por cable en vez de usar Bluetooth (Cambio de diseño).

Afortunadamente logramos una lectura exitosa por parte de los pines del LPC:



También añadimos para comprar unos conversores digitales para las termocuplas por cuestión de tiempo y de practicidad (Así que el diseño de la pagina 15-16 quedó descartado). Lo que se adjunto es un sketch basico del funcionamiento del nuevo conversor. El MAX6675.



- **Semana 13/08-28/08:**

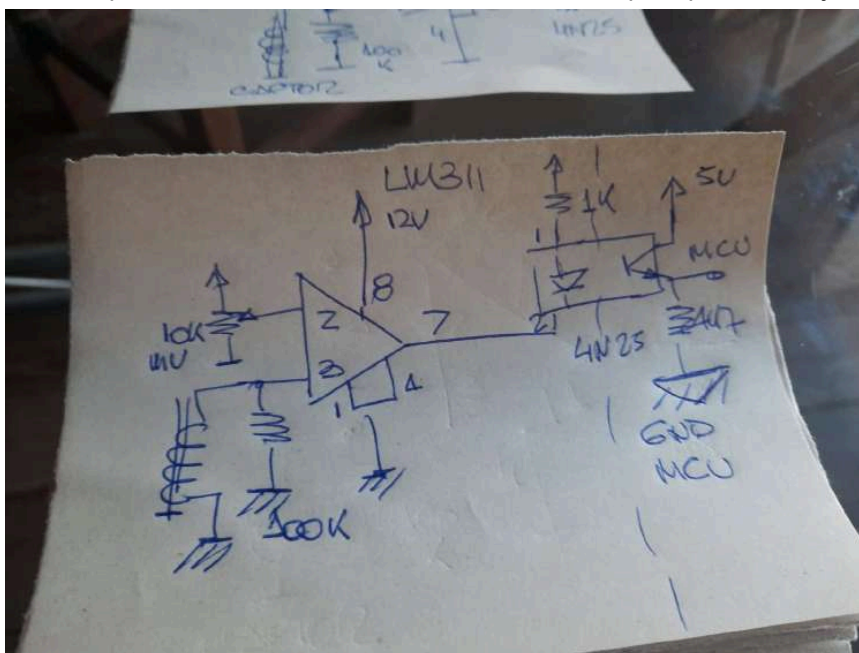
Estas semanas estuvimos teniendo ayuda de Norberto Molinari como tutor que nos daría apoyo al proyecto.

El nos proporcionó Termocuplas para las cabezas de cilindro:

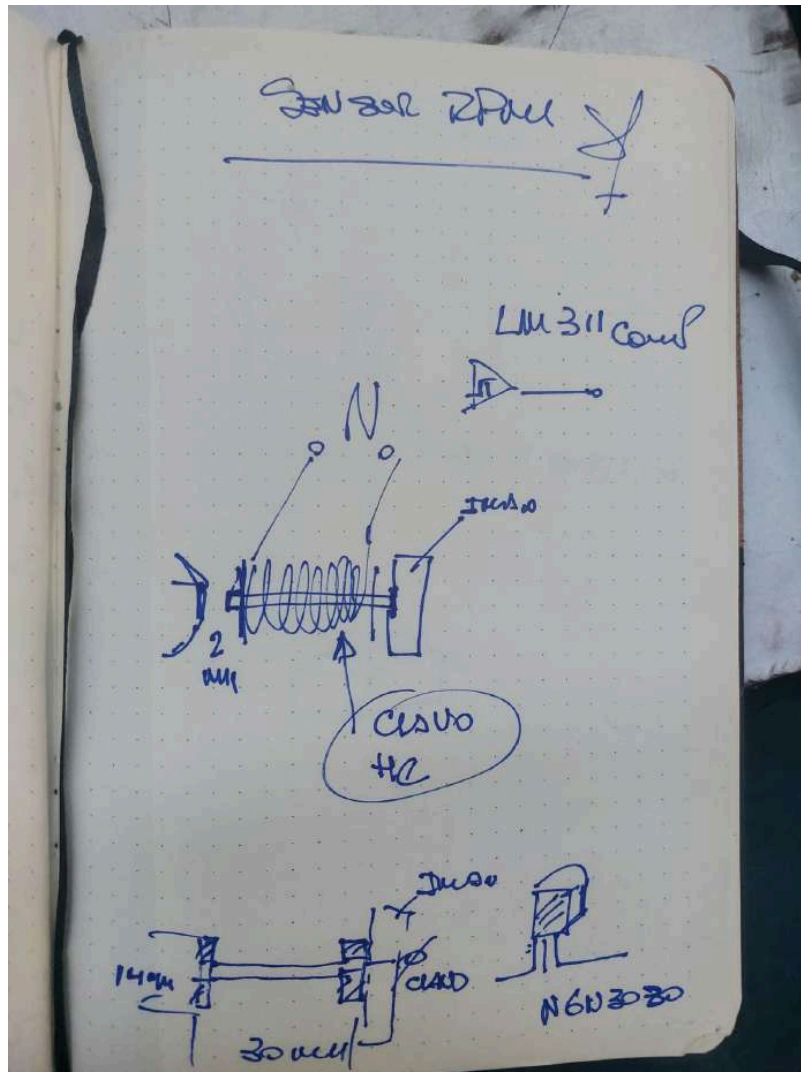


Estas mismas tuvieron y están en proceso de modificación debido al reducido espacio en el motor.

También planteamos nuevos circuitos más concretos para poder trabajar:



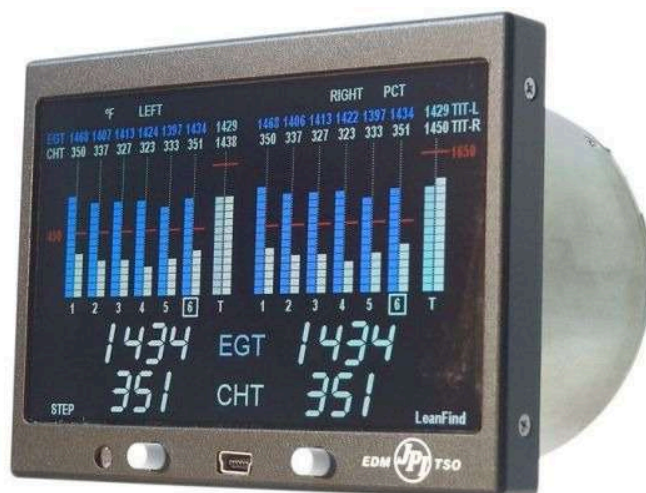
Este es el circuito comparador (Que usaremos para contar los pulsos) Y posteriormente se usa un optoacoplador para proteger el microprocesador. También una versión mejor hecha de lo que sería el sensor de RPM:



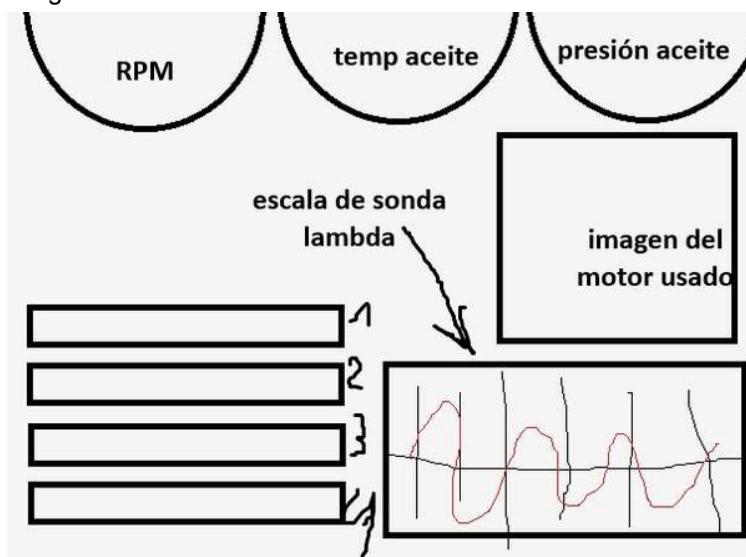
Esto actuaría como un sensor de efecto Hall pero de forma mas casera, para tener todo mas controlado. Durante la siguiente semana compaginamos estos circuitos.

- **Semana 4/09-09/09:**

Durante esta semana estuvimos armando un prototipo de lo que seria la interfaz de nuestro programa. Inspirandonos en uno de los instrumentos de GARMIN:



Luego realizamos un boceto:

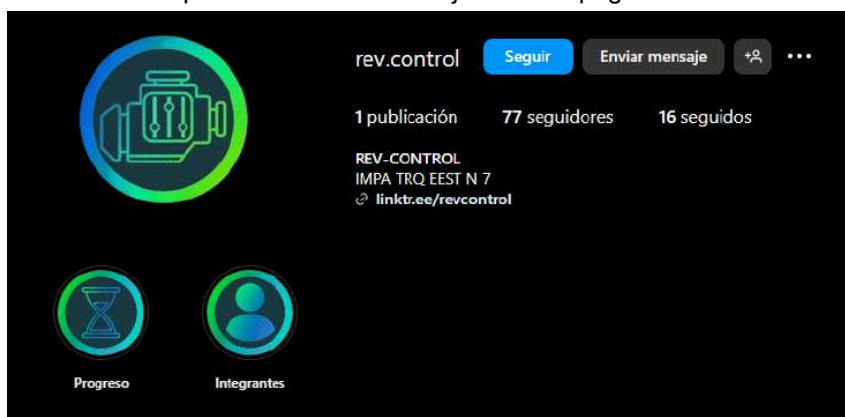


Y este es un esquema mejor terminado

7° 1° AV - Comisión C - Grupo Rev-Control- 2024



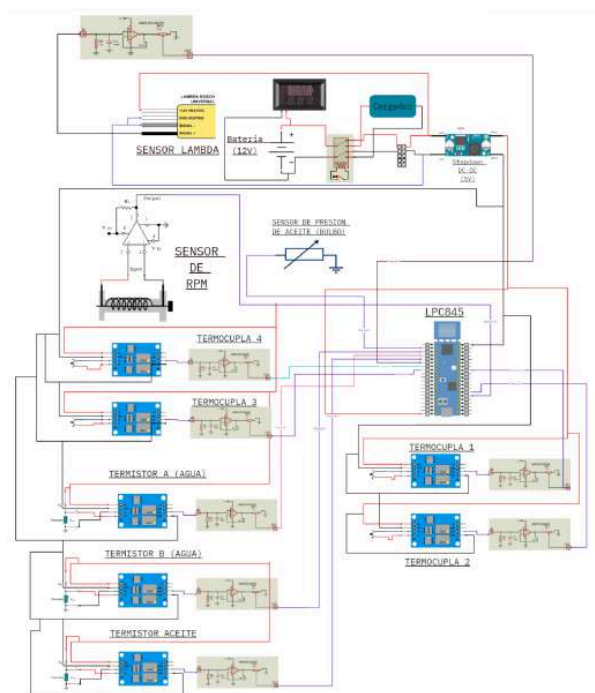
También se empezaron a hacer mas ajustes a la pagina de Github e instagram:





- **Semana 11/09-19/09:**

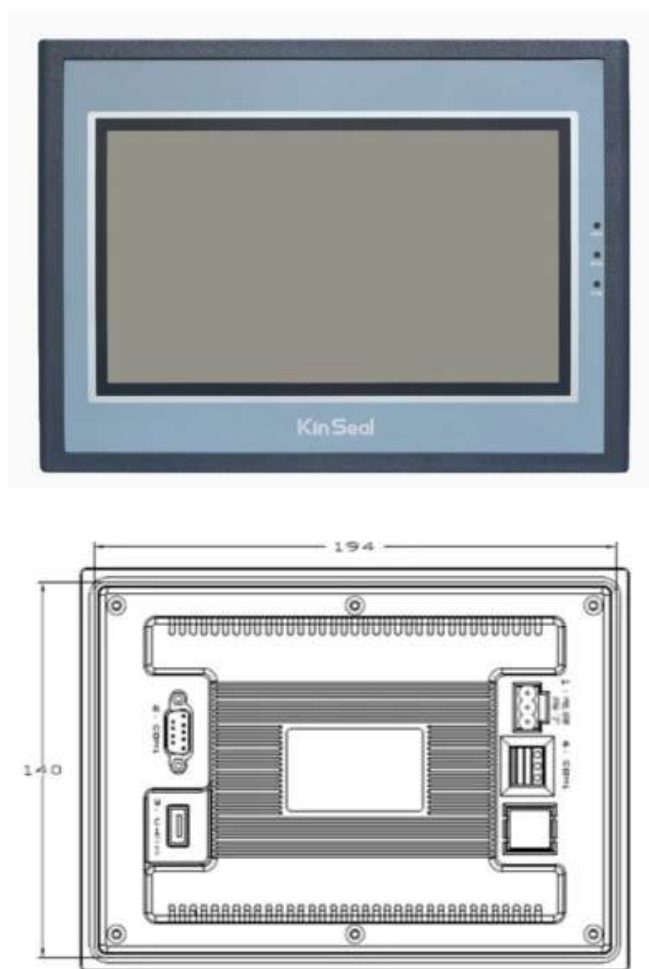
Esta semana armamos el primer diagrama completo del circuito. Este primer prototipo está ampliamente sujeta a cambios pero sirve como una manera mas “organizada” de tener los conceptos.



Aunque ya empezamos a hacer una versión distinta de este circuito que sea mas organizado y prolijo

- **Semana 20/09-30/09:**

Durante esta semana recibimos un monitor configurable para mostrar la interfaz del programa (Kinseal AMZ070W01RAGD)

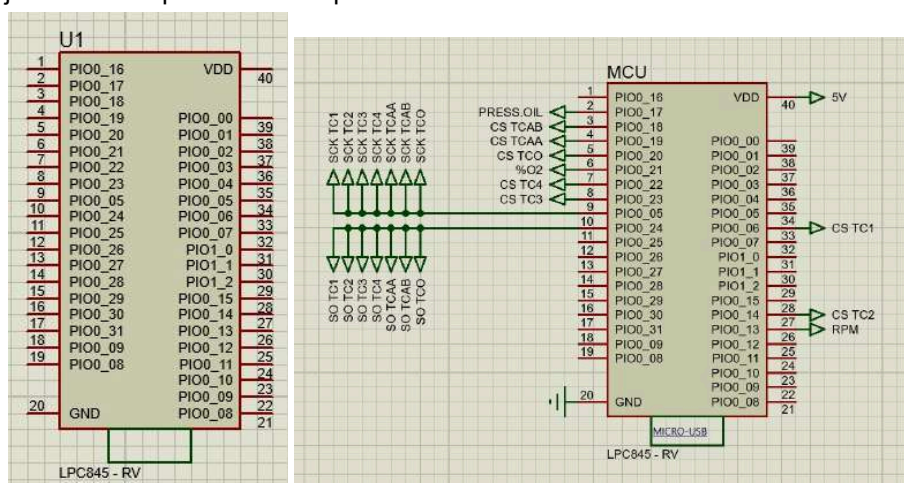


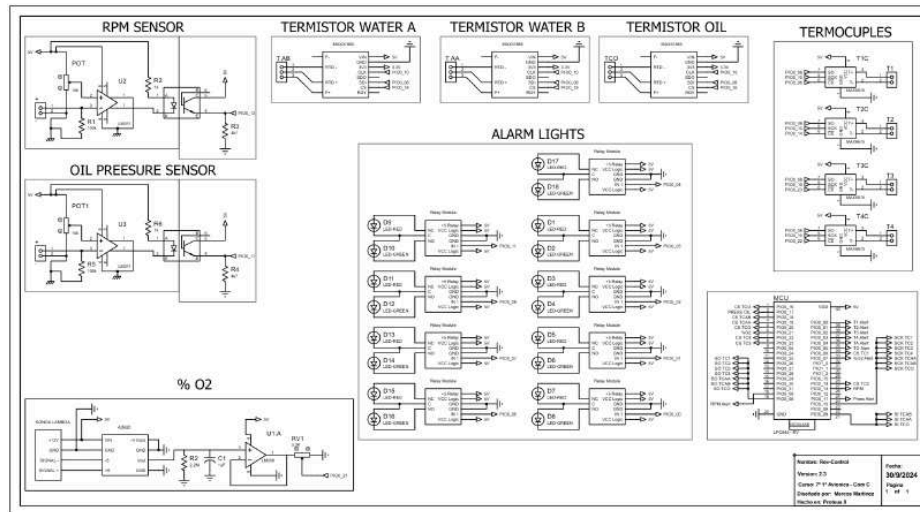
Este monitor fue proporcionado por Molinari también. Debido a la configuración del monitor tuvimos que abortar la idea de hacer la comunicación por Bluetooth y directamente la haremos por cable.

A su vez estuvimos armando una caja/tablero donde meter todos los circuitos electrónicos:



A su vez también hicimos una nueva versión del circuito general (Versión 2.3) para tener una visión más clara de las conexiones y distribución del tablero internamente junto a descripción de cada parte:

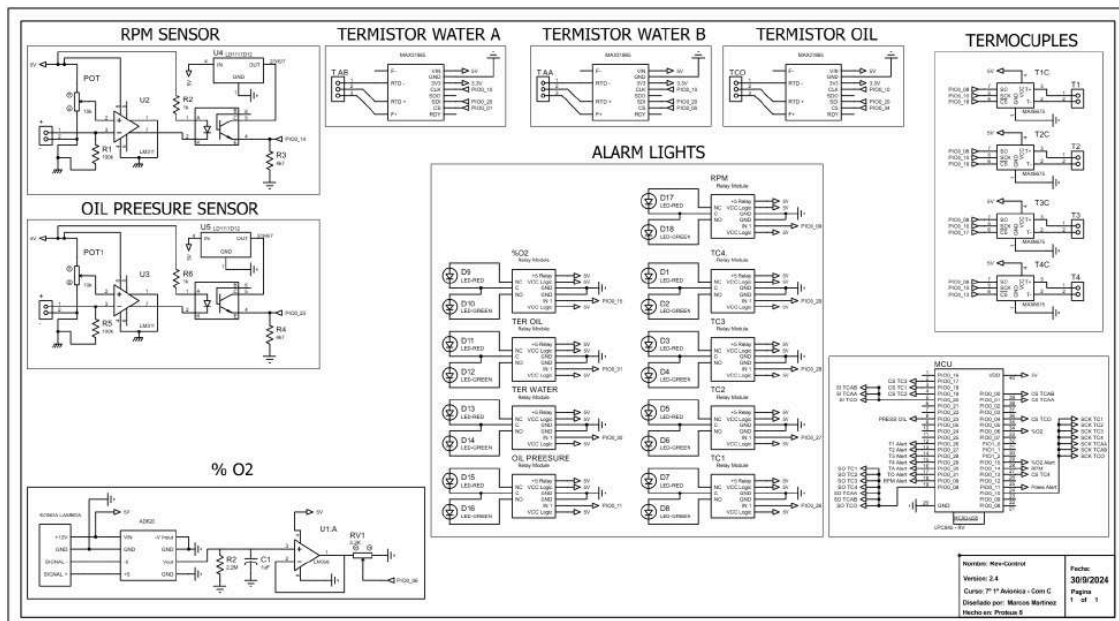




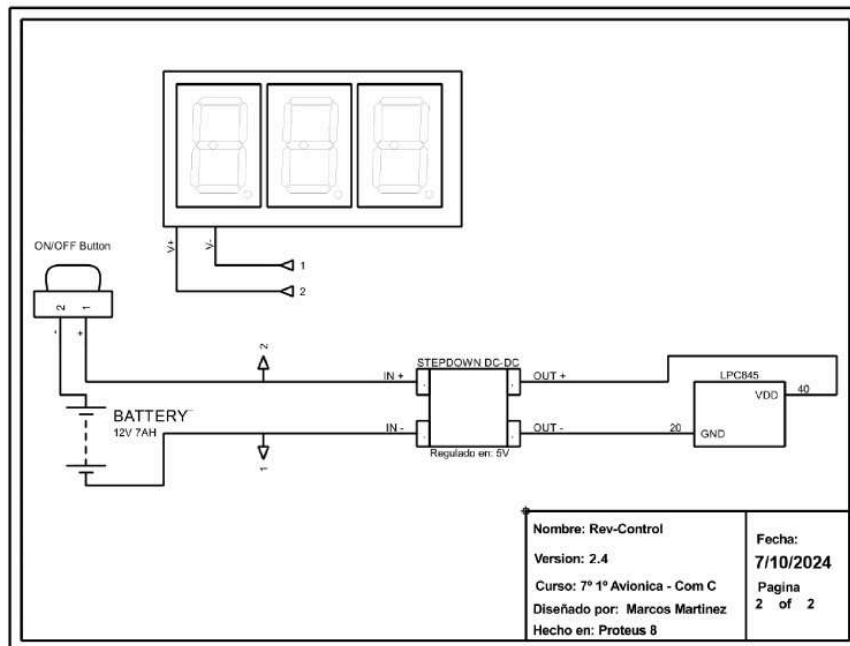
(Este esquema se puede ver más claramente en la página de Github: [Acceso](#))

• Semana 01/10-17/10:

Actualizamos el esquema del circuito (Version 2.4) y agregamos una segunda parte más secundaria (Este tambien se puede tener acceso en la página de Github [Acceso](#))



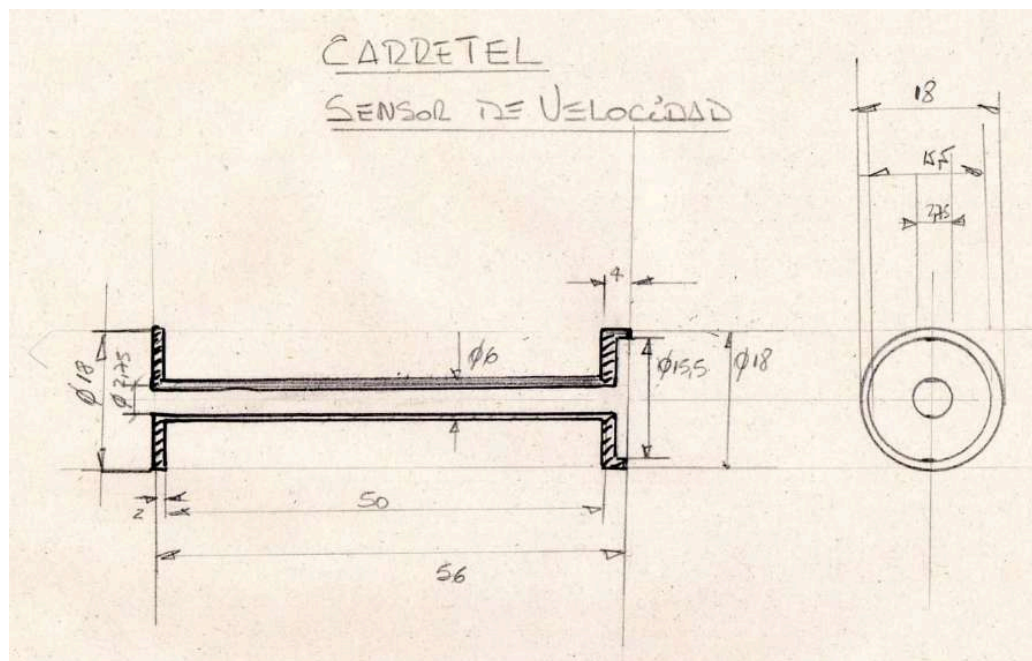
REV-CONTROL



Como se puede ver se agregó un botón de emergencia para apagar todo el circuito en caso de algun accidente o imprevisto:



A su vez tambien se hizo el plano con medidas el sensor de RPM (En un formato mas casero):





Este mismo actúa como un transformador que genera inducción para poder ir contando los pulsos en un comparador.

Debido a los cambios de circuito a partir de la version 2.3 empezamos a comprar los nuevos componentes que necesitabamos:



Conseguimos este tipo de ojos de buey de bajo voltaje para poder usar como sistema de alarmas (A esto le agregamos que cambiamos el cristal rojo en algunos de ellos por un acetato verde para poder tener ambos colores)



Y también una buzzer de bajo voltaje.

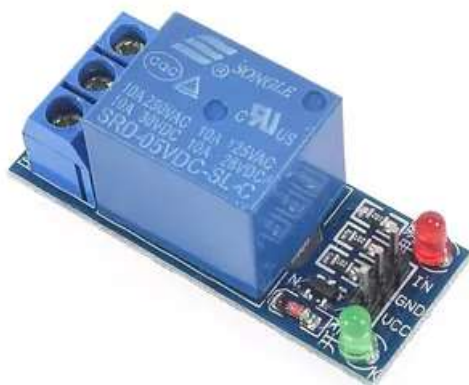
Se arregló la lectura del MAX6675 para las termocuplas:

```
117
118     while(1) {
119         for(t = 0; t < 7; t++){
120             // Leo la temperatura de la termocupla
121             float temp = max6675_get_temp();
122             // Muestro como entero
123             if (temp < 300){
124                 PRINTF("La temperatura fue: %d\n", (uint16_t)temp);
125             }
126             // Demora
127             for(uint32_t i = 0; i < 1000000; i++);
128         }
129     }
130     return 0;
```

```
10 // Pines de Slave Output (MISO)
11 #define SO      20
12
13 // Pin de clock
14 #define SCK     10
15
16 // Pines de Chip Select
17 #define CSTCAA  1
18 #define CSTCAB  0
19 #define CSTC0   4
20 #define CSTC1   18
21 #define CSTC2   19
22 #define CSTC3   17
23 #define CSTC4   13
24
25 //Array de todos los Chip Select
26 uint8_t CS[7] = {CSTCAA, CSTCAB, CSTC0, CSTC1, CSTC2, CSTC3, CSTC4};
27
```

También fuimos a comprar una placa de control con relés para poder encender estas luces y que el microprocesador con sus 3,3v pueda encenderlos cuando se necesite (Esto gracias a los optoacopladores):

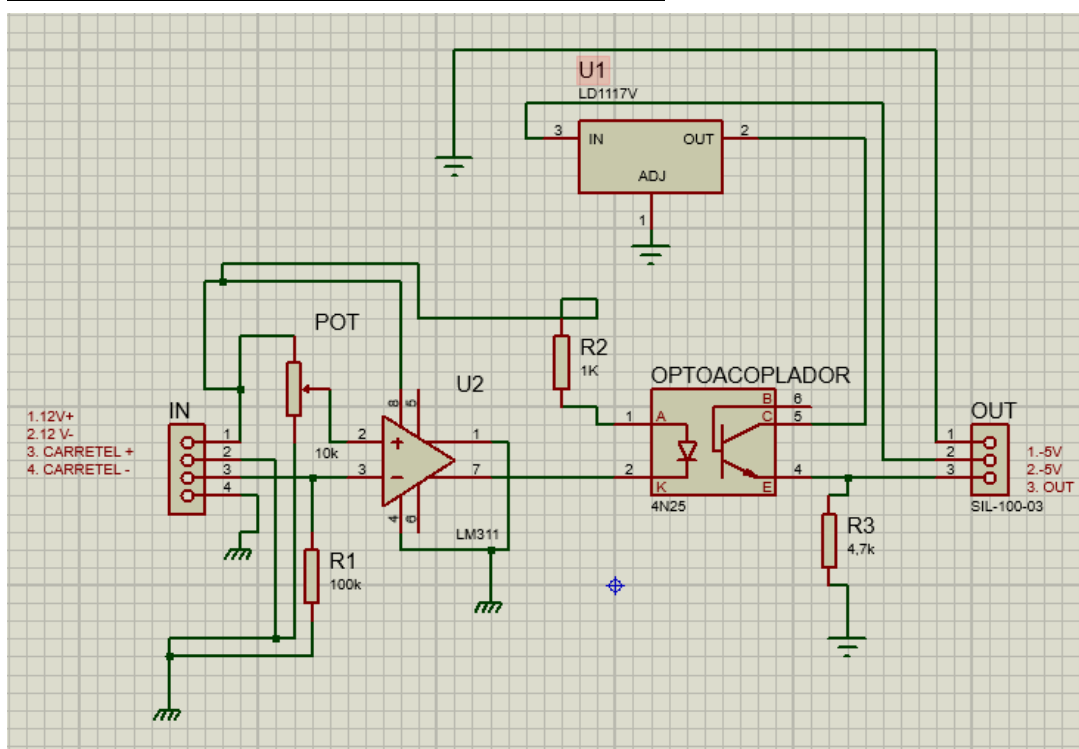
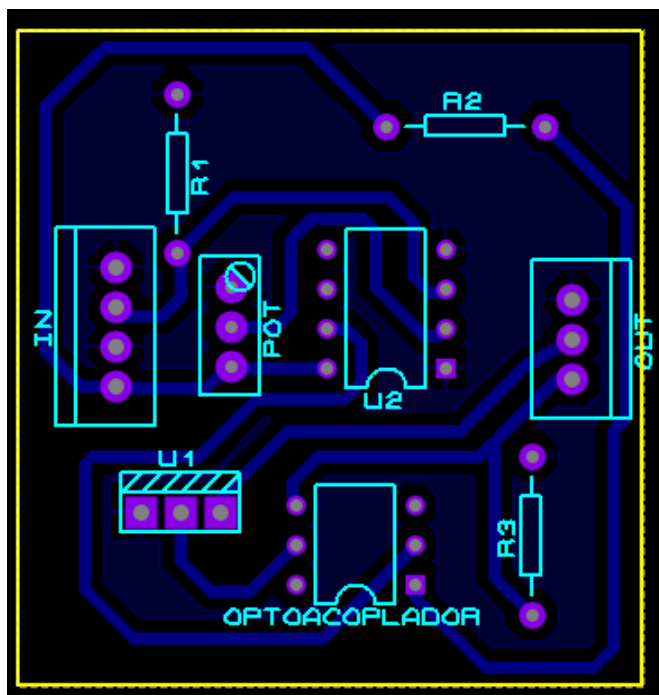




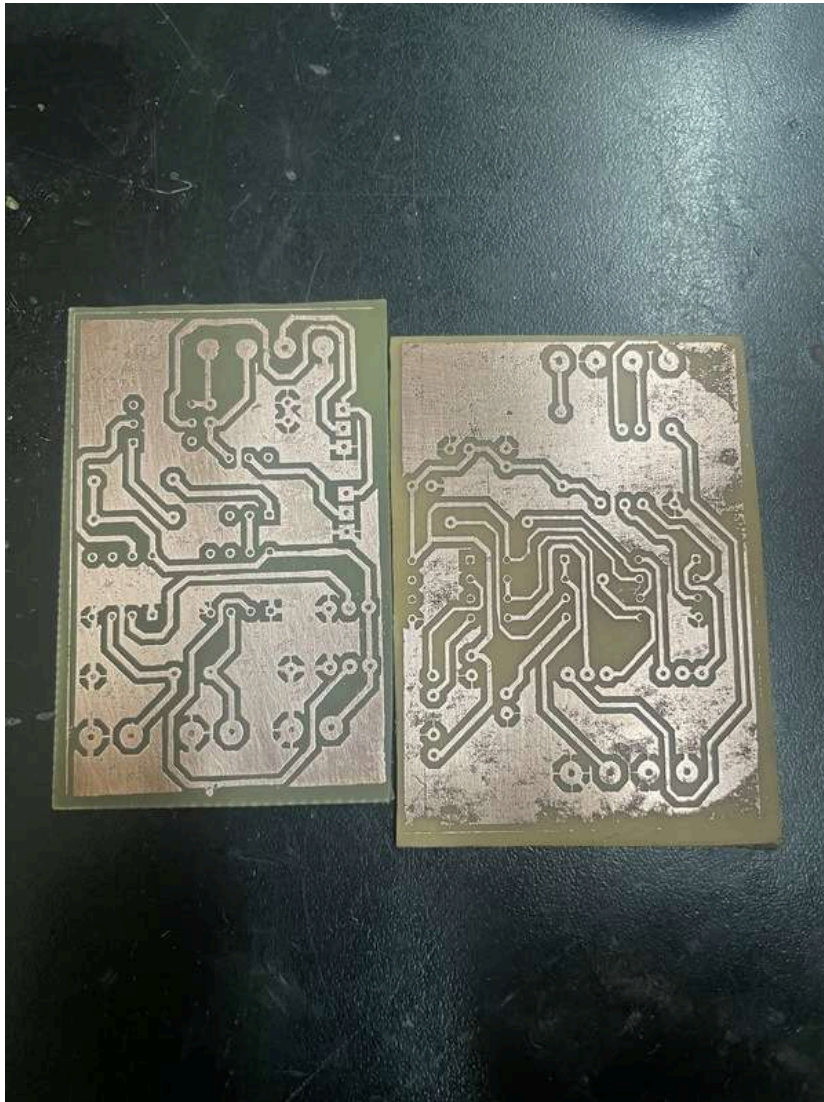
También compramos 3 conversores similares a los que usamos para las termocuplas pero para los 3 termistores (2 de agua y 1 de aceite) usando RTDs PT100:



También estuvimos armando las placas PCB de los circuitos:



REV-CONTROL



Informacion mas detalladas de estas en la pagina de [Trello](#) y [Github](#).

- **Semana 18/10-28/10:**

Arreglamos la lectura de los termistores y del MAX31865:

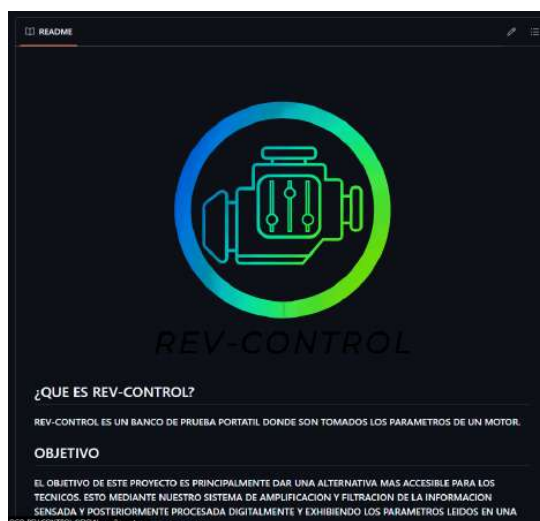
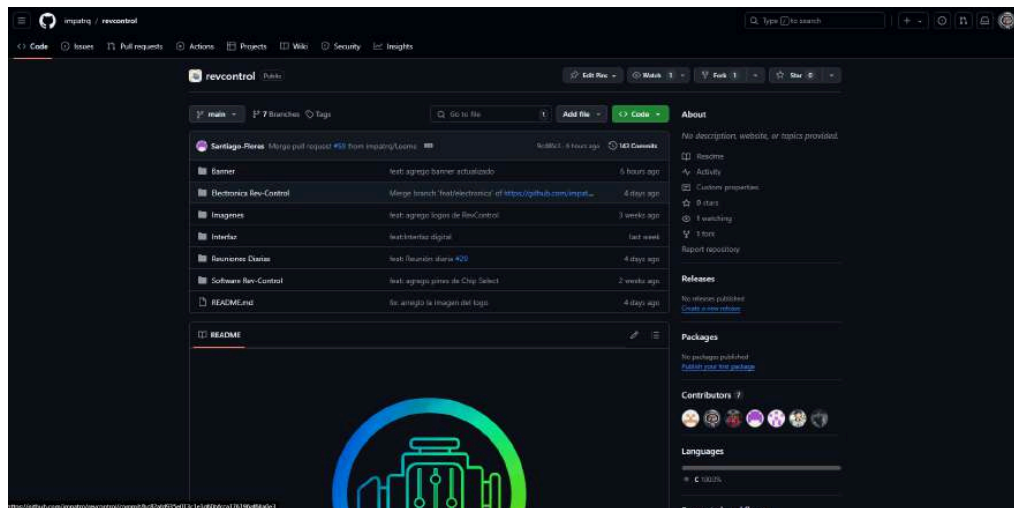
a

```

52 void Max31865_readRegisterN(Max31865_t *max31865, uint8_t addr, uint8_t *buffer, uint8_t n){
53     uint8_t tmp = 0xFF;
54     addr &= 0x7F;
55     GPIO_PinWrite(GPIO, 0, 17, 0);
56     SPI_MasterTransferBlocking(SPI0, &xfer); //transmite
57     spi_transfer_t xfer = {
58         .txData = NULL, // No hay nada para enviar
59         .rxData = buffer, // Array donde guardar lo leído
60         .dataSize = n, // Dos bytes de cuando que leer
61         .configFlags = kSPI_EndOfTransfer | kSPI_EndOfFrame
62     };
63     GPIO_WritePin(GPIO, 0, 17, 1);
64 }

```

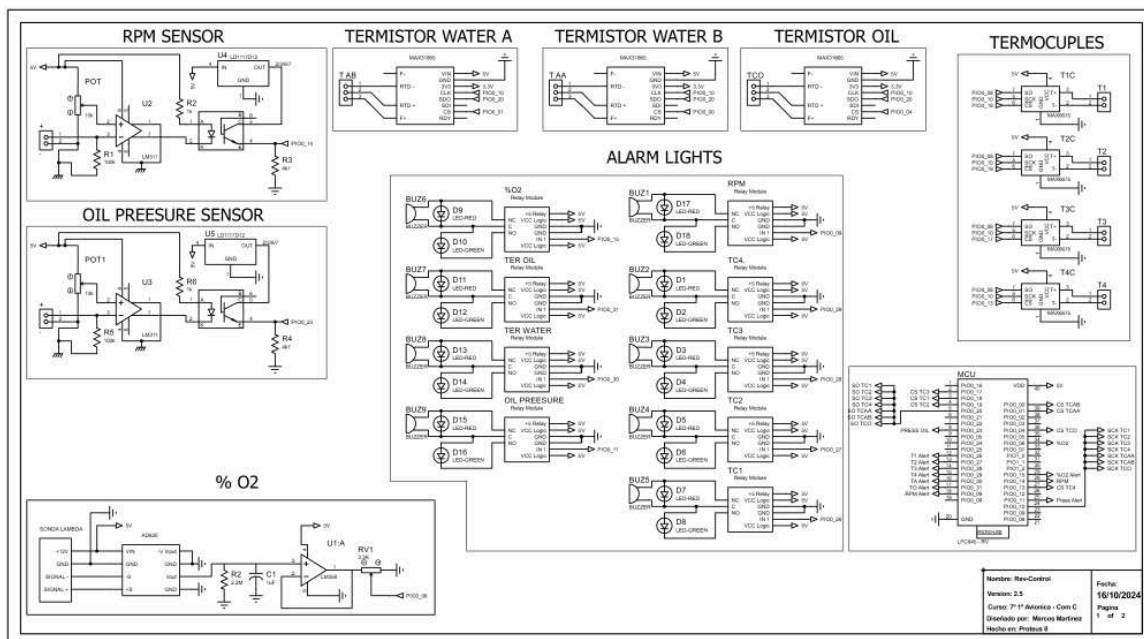
Durante este plazo empezamos a compaginar la carpeta técnica y a su vez subir toda la información digital del proyecto a la página de Github ([Pagina](#))



También durante este plazo tomamos de descartar la caja/tablero que habíamos hecho debido a su apariencia poco atractiva, así que quedo como un prototipo. Asi que recurrimos a otros medios y pudimos conseguir un contacto que nos podía proporcionar el nuevo tablero:



Tambien hicimos la versión 2.5 del circuito con arreglos en los pines de lectura:



Durante este plazo también se empezó con la programación de la interfaz digital del monitor Kinseal (Usando el programa: HMILite 2.22 y el Kinseal Studio), este mismo trabaja con un lenguaje similar al lenguaje Ladder del PLC::



REV-CONTROL

7° 1° AV - Comisión C - Grupo Rev-Control- 2024



También arreglamos algunas partes del modelo 3D del maletín:



A su vez también seguimos con la programación de MAX31685 de lectura de Termistor:

```
117
118     while(1) {
119         for(t = 0; t < 7; t++){
120             // Leo la temperatura de la termocupla
121             float temp = max6675_get_temp();
122             // Muestro como entero
123             if (temp < 300){
124                 PRINTF("La temperatura fue: %d\n", (uint16_t)temp);
125             }
126             // Demora
127             for(uint32_t i = 0; i < 1000000; i++);
128         }
129     }
130     return 0;
```

```
10 // Pines de Slave Output (MISO)
11 #define SO      20
12
13 // Pin de clock
14 #define SCK     10
15
16 // Pines de Chip Select
17 #define CSTCAA  1
18 #define CSTCAB  0
19 #define CSTCO   4
20 #define CSTC1   18
21 #define CSTC2   19
22 #define CSTC3   17
23 #define CSTC4   13
24
25 //Array de todos los Chip Select
26 uint8_t CS[7] = {CSTCAA, CSTCAB, CSTCO, CSTC1, CSTC2, CSTC3, CSTC4};
27
```

```
52 void Max31865_readRegisterN(Max31865_t *max31865, uint8_t addr, uint8_t *buffer, uint8_t n) {
53     uint8_t tmp = 0xFF;
54     addr &= 0x7F;
55     GPIO_PinWrite(GPIO, 0, 17, 0);
56     SPI_MasterTransferBlocking(SPI0, &xfer); //revisarxxxx
57     spi_transfer_t xfer = {
58         .txData = NULL, // No hay nada para enviar
59         .rxData = buffer, // Array donde guardar lo leído
60         .dataSize = n, // Dos bytes se tienen que leer
61         .configFlags = kSPI_EndOfTransfer | kSPI_EndOfFrame
62     };
63     GPIO_WritePin(GPIO, 0, 17, 1);
64 }
```

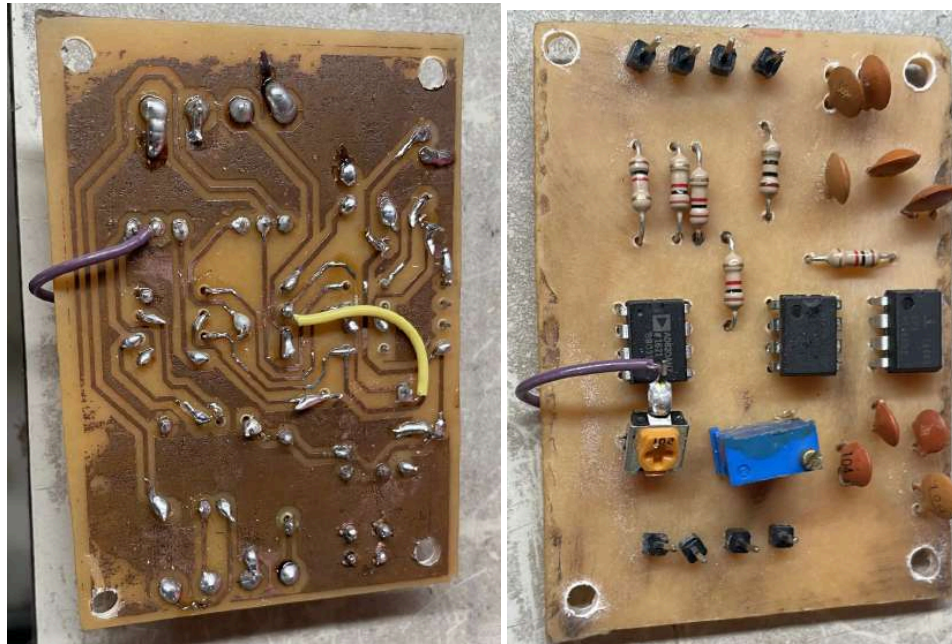
- **Semana 3/11-10/11:**

Esta semana finalizamos todos los pcs y soldamos sus componentes:

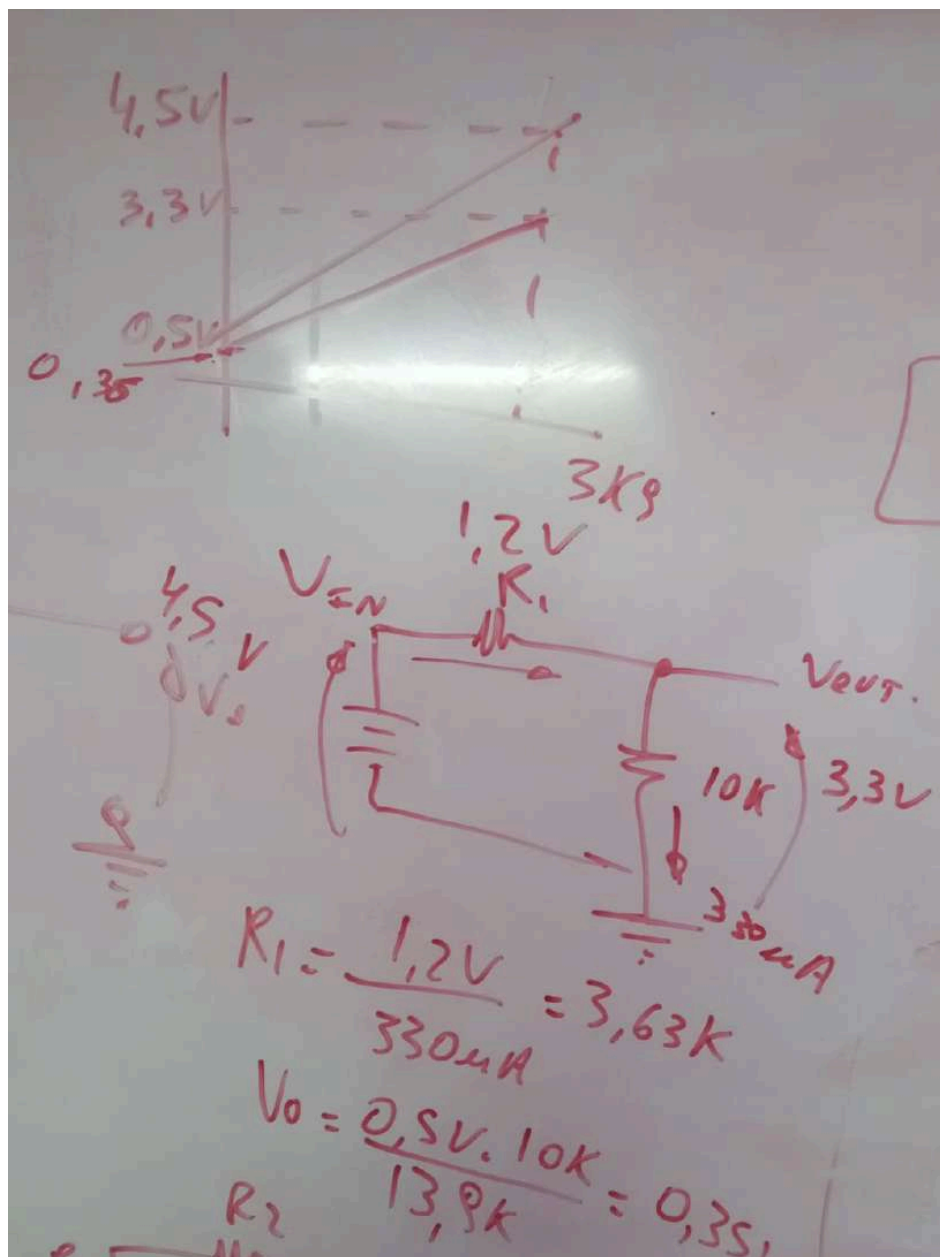


REV-CONTROL





Hicimos un rediseño apresurado del circuito de sensor de presión de aceite debido a que el “Comparador Doble” no era una buena alternativa, así que re armamos el circuito comparador y lo dejamos solamente para el RPM, para el sensor de presión de aceite usamos un divisor de voltaje de diseño propio, ya que este sensor solo muestra un rango de 0 a 4,5v pero lo bajamos de 0 a 3,3v (aprox) para que pueda ser leído por el pin de ADC del micro:



Este diseño fue calculado 3K con una en serio de 10K para que salgan esos 3,3v (aprox) de lectura.

Sensor de presión de aceite } Proto #1

Atenuador
Regulador

Divisor de Voltaje (4,5V → 3,3V)

$$V = \frac{R}{I} \Rightarrow V = \frac{R1 + R2}{I}$$

$$4,5V = \frac{3K + R2}{0,04A}$$

3K?

4,5V

10K

OUT

Caso simulación:
3,46V

No sirve

LM1117

IN

3,3V

OUT

GND

Le mando pero no tiene la suficiente carga:

3,45V?

No sirve?

No dan los números

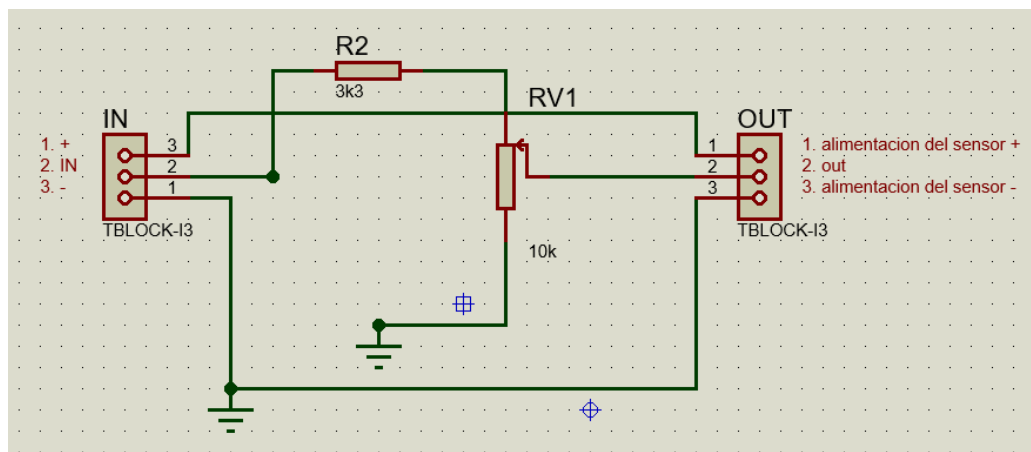
1NF

IN?

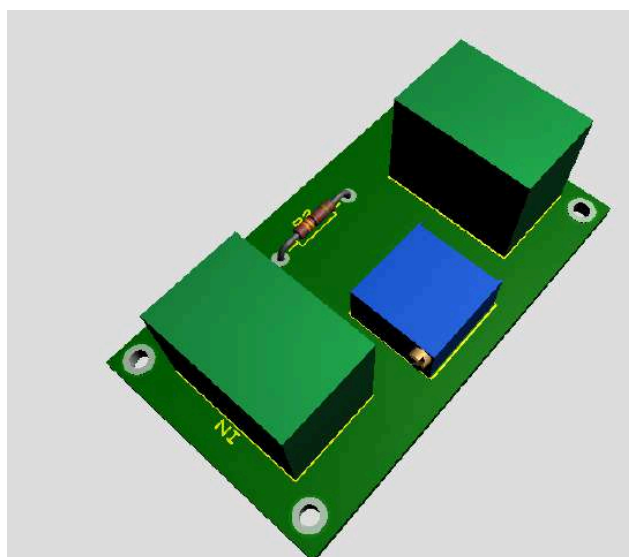
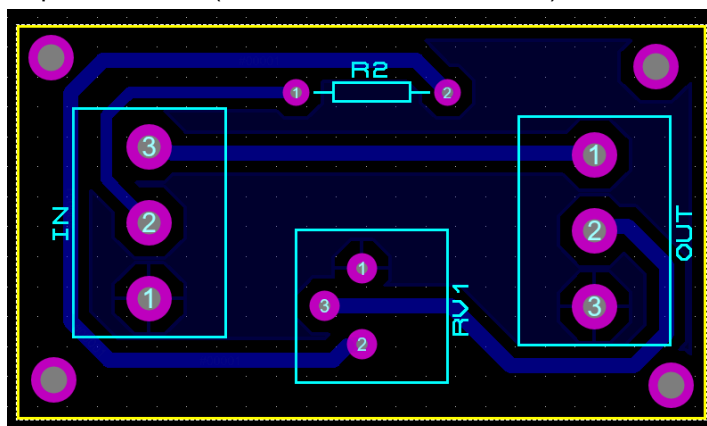
47Ω?

OUT





Cambiamos el diseño y pusimos un preset de 10k para poder tener mas controlado los valores de salida y a su vez pusimos pines de alimentación para para el sensor de presion aceite (Para facilitar las conexiones).

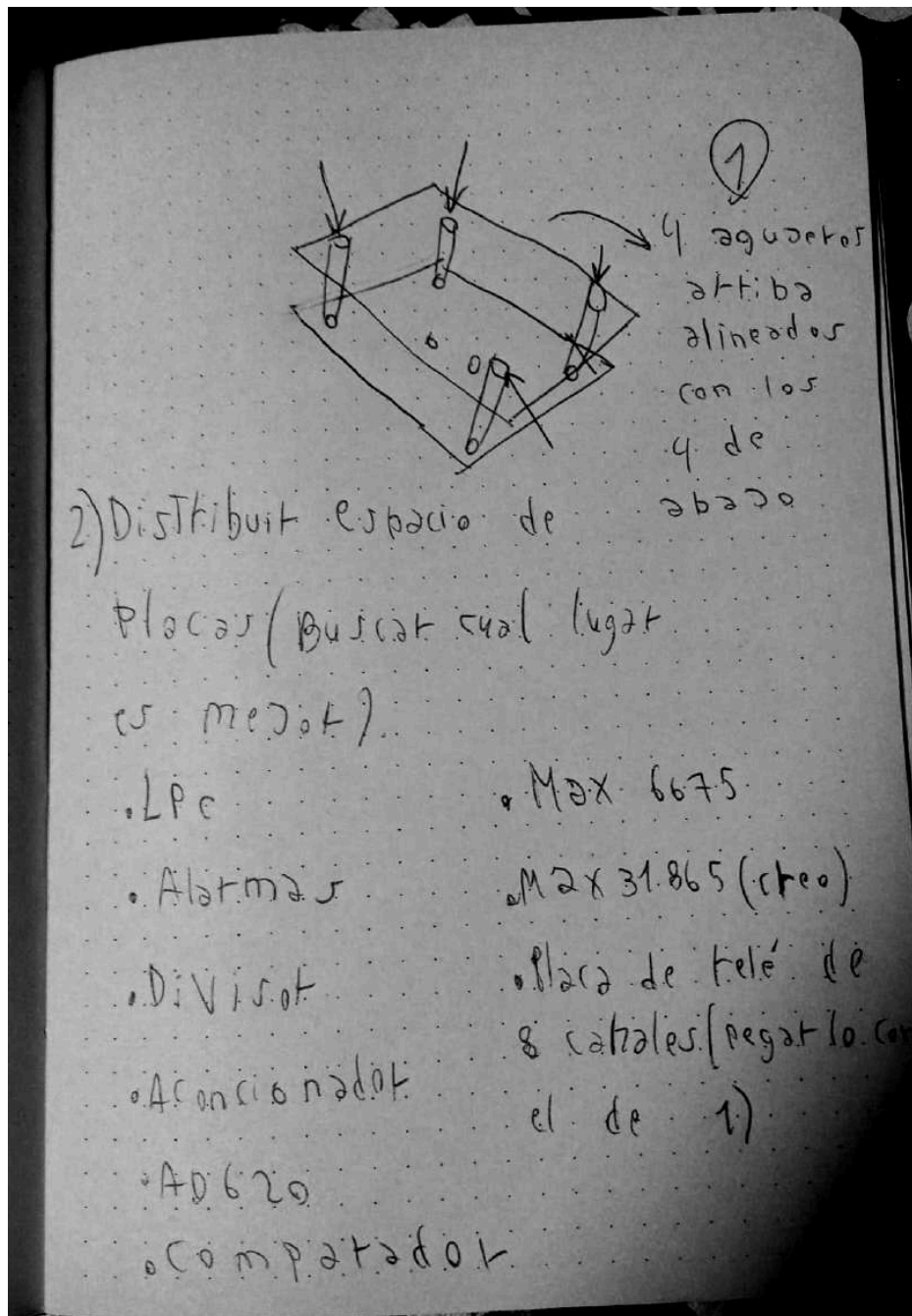


Tambien Norberto Molinari nos trajo uno de los Termistores para medir la temperatura de aceite y agua:



Se usan **termistores** en líquidos porque son más precisos en rangos moderados de temperatura (como $-40\text{ }^{\circ}\text{C}$ a $150\text{ }^{\circ}\text{C}$), más sensibles a pequeños cambios, más rápidos en respuesta, y más económicos que las **termocuplas**, que están diseñadas para rangos extremos y requieren circuitos complejos de compensación. Además, los termistores son más fáciles de integrar en sistemas simples de monitoreo.

Tambien armamos sketches de como se veria la estructura interna del modulo de lectura:



También, estuvimos haciendo más desarrollo con la parte del monitor Kinseal y revisando los protocolos:

Pin	signal	COM1 RS422	COM1 RS485	COM1 RS232	COM2 RS485
1	RX-(B)				RS485B(-)
2	RxD			RS232 transmit	
3	TxD			RS232 receiving	
4	TX-				RS485A(+)
5	SG	Signal ground			
6	RX+(A)	RS422 reception-			
7	RxD	RS422 reception+			
8	TxD	RS422 transmit-	RS485B(-)		
9	TX+	RS422 transmit+	RS485A(+)		

Custom protocol use case

Actual application requirements of customers:

The first type of macro is to click the key macro. Each time the key is pressed, one instruction is sent:

The second type of macro is to set the key macro. When the key is pressed, the command is sent once. When the key is popped up, the other side needs to send the command to pop up:

The third type of macro, data display macro, receives third-party instruction values.

2. The instruction format of sending frame and returning frame in practical application of free protocol: 5 bytes in total, the first touch screen address (user defined), the second key mode (LW / lb), the third key ID, the fourth and fifth parameters, global_Run macro, mainly returns to the touch screen value, lw01-05

First: STR [0] = 0x1: // touch screen address

Second: STR [1] = 0x00: // lw01 lb00

Third: STR [2] = 51: // key ID

Fourth: STR [3] = 0x01: // low byte of data

Fifth: STR [4] = 0x00: // high byte of data

According to the needs of customers, the following steps are taken to realize the simple demonstration function

Hardware environment: A host computer, a touch screen and a serial debugging assistant and a serial simulation line

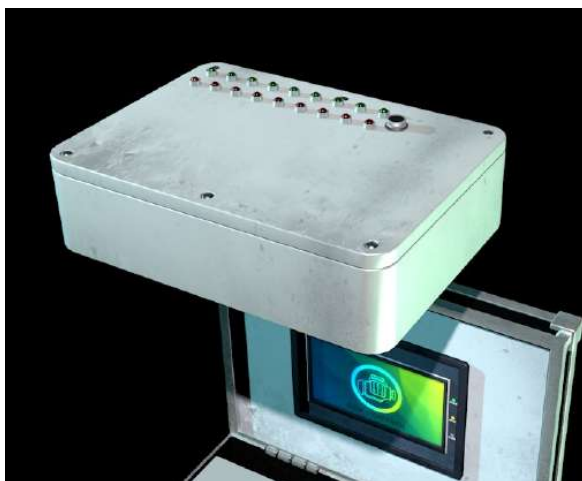
(1) Create a communication and create two macro instructions as shown in the following figure (Global Run and set51)

También se terminó el rediseño del Maletín en 3D para usar en publicidad:



REV-CONTROL

7° 1° AV - Comisión C - Grupo Rev-Control- 2024



7° 1° AV - Comisión C - Grupo Rev-Control- 2024



También terminamos de armar la primera parte de las placas:

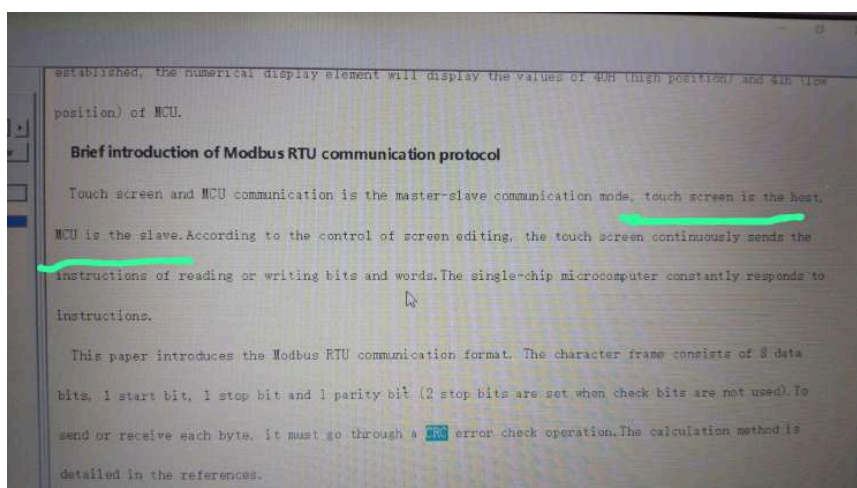


- **Semana 13/11-17/11:**

Conseguimos un cable con conector RS232 para la comunicacion con el monitor Kinseal y el micro:



También se empezó a usar un protocolo Modbus para la comunicación del micro con el monitor, esto debido a que es un protocolo standard y que se puede usar tanto en microcontroladores como en PLC.



Protocolo Modbus

Modbus es un protocolo de comunicación industrial creado en 1979, ampliamente usado para conectar PLC, sensores y actuadores. Opera en un modelo maestro-esclavo y es compatible con conexiones en serie (RTU, ASCII) y Ethernet (TCP/IP).

En **Modbus RTU**, los datos se transmiten en binario, lo que lo hace eficiente para RS-485 y RS-232. **Modbus TCP/IP** adapta el protocolo para redes Ethernet modernas, permitiendo múltiples conexiones simultáneas.

Es simple, ampliamente adoptado y fácil de implementar, aunque carece de seguridad avanzada como encriptación. Ideal para automatización industrial.

También armamos soportes pequeños para sostener las placas en el soporte de acrílico:



También estuvimos haciendo revisiones para conectar los sensores de presión de aceite y RPM:

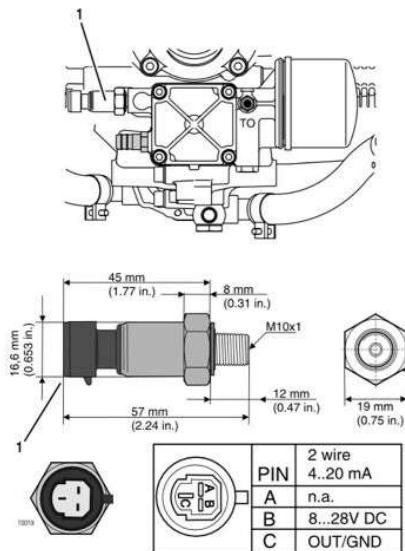
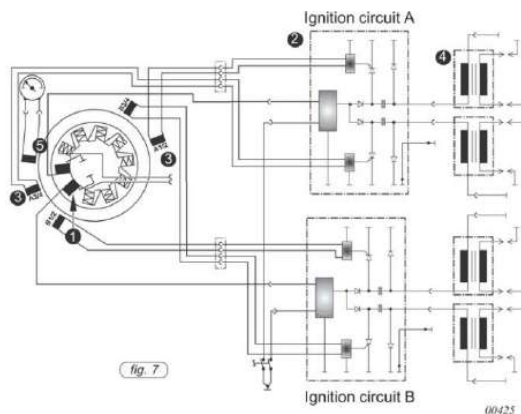


Figure 9.5: Oil pressure sensor

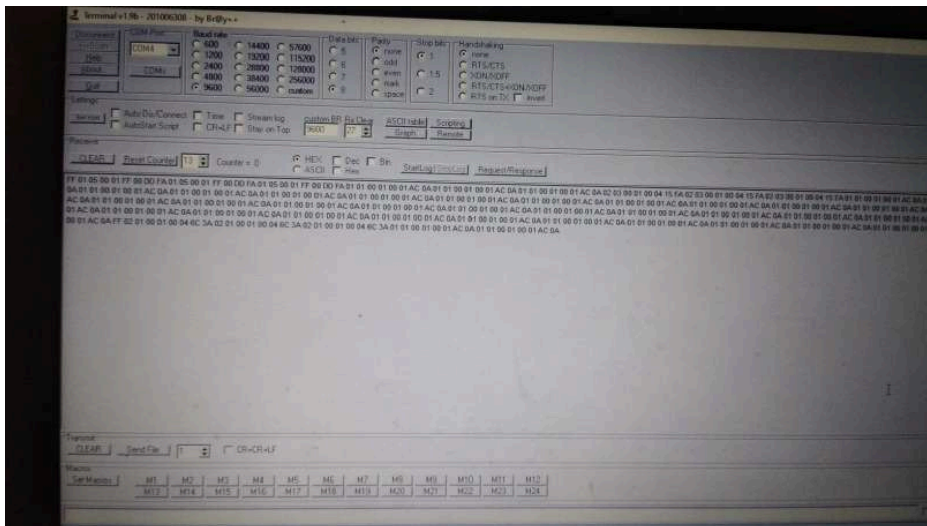


También nos sacamos la foto grupal para la carpeta técnica y presentación:

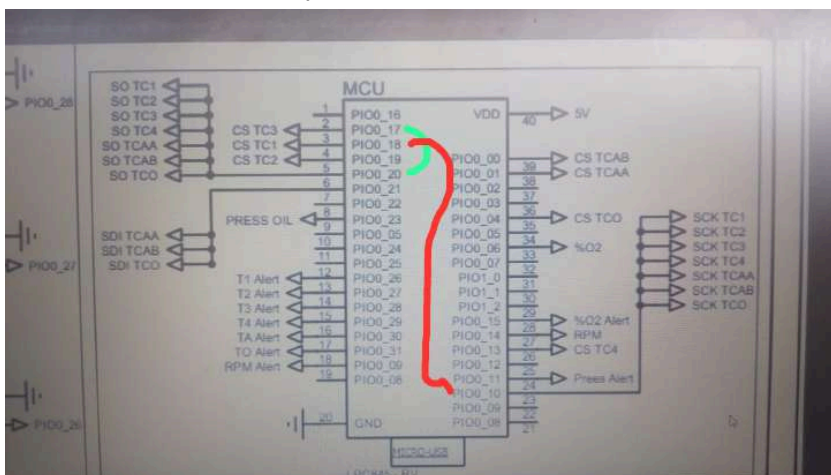
7° 1° AV - Comisión C - Grupo Rev-Control- 2024



Tambien hicimos las pruebas de comunicacion:



Tambien re acomodamos partes de las conexiones:



7° 1° AV - Comisión C - Grupo Rev-Control- 2024

Tambien armamos y terminamos la placa donde apoyar el monitor y tener una visualizacion clara:





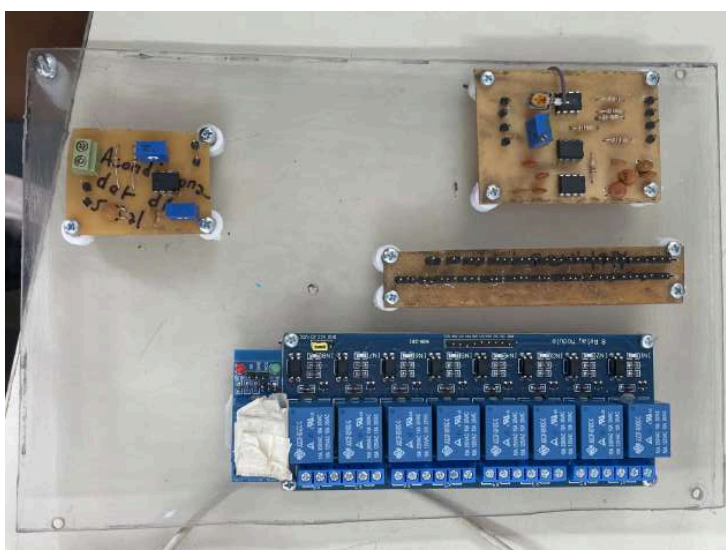
- **Semana 20/11-25/11:**

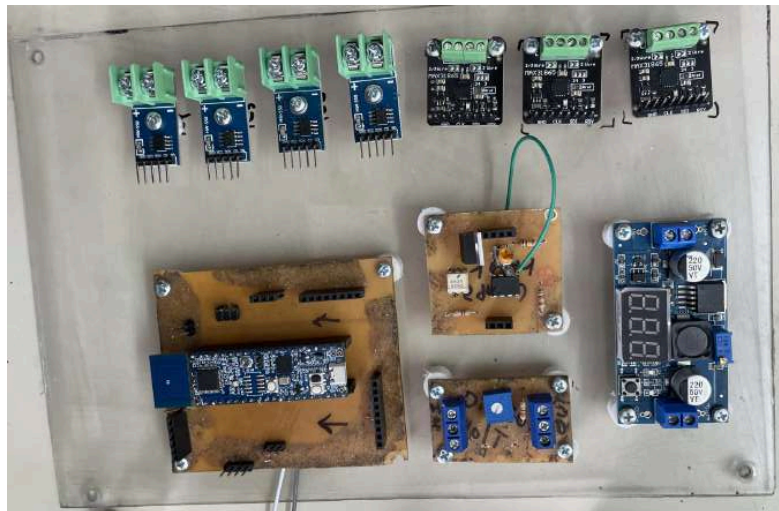
Esta semana fijamos todas las placas a las planchas de acrílico:



También terminamos los soportes de las placas de los circuitos aunque decidimos por cuestiones de tiempo, hacerlo en impresión 3D:

7° 1° AV - Comisión C - Grupo Rev-Control- 2024





Y terminamos el modulo de lectura:



También compaginamos el diseño final de Rev-Control, presentado en la Expo 2024, junto al proyecto ROTAX:



En opinión del equipo, fue una gran experiencia de poder trabajar no solo en conjunto con un grupo de Aeronautica o con el motor ROTAX en si, sino también el hecho de plantear y compaginar este proyecto que se convirtió en una gran oportunidad de poder ampliar nuestros conocimientos en electronica, programacion, diseño, publicidad y organización en el área de trabajo.

7° 1° AV - Comisión C - Grupo Rev-Control- 2024

