Laboratorio 03

Actividad 1

1. Se implementó un driver device.c que provee la interfaz deseada.
Se diseñó una estructura adc_cfg para la mnipulación de los valores provistos por el ADC que incluye las variables:

1. canal: un entero que representa el canal a ser usado
2. confActual: un entero que representa su identidad en un arreglo de configuraciones múltiples
3. ultMedicion: un entero que representa la última lectura obtenida del ADC. Su valor varía entre 0 y 1023.
4. callback: Un puntero a función que apunta al espacio de memoria donde se almacena la función a ser usada por el canal 'canal'

Para valerse del valor provisto por el ADC se configuró el ADC en modo *single-conversion* y se usó la interrupción *ISR(ADC_vect).*

2. Para convertir el valor provisto por el ADC a grados centígrados, se leyó el manual de consulta del Atmega328p y se utilizó la siguiente fórmula:

$$\frac{Voltaje\ maximo(5) * Ultimo\ valor\ leido * Factor\ de\ escala(100)}{La\ cantidad\ de\ valores\ posibles\ por\ el\ analogRead(1024)}$$

El factor de escala es 100 dado que la T crece 1C cada 10mV.

Se estimó la frecuencia de muestreo. Dado que 15s / 100 = 0,15s se recurrió a tomar muestras cada 1,5s de tiempo. Dados algunos conflictos de sincronización internos y algunas necesidades externas (como darle tiempo al usuario de leer los valores en el lcd y la necesidad de sincronizar temporalmente con la aplicación externa), se utilizaron ciertos retardos dentro de la aplicación que hace que no se sincronice perfectamente con la frecuencia buscada.

Para manipular el teclado del LCD se implementó el archivo *teclado.c* con su correspondiente encabezado *teclado.h.*
Para manipular el sensor lm35 se implementó el archivo *device.c* con su correspondiente encabezado *device.h*.
Para configurar ambos dispositivos y mostrar mensajes en pantalla se implementó el archivo Lab3Act1.

El loop en el main basicamente ronda entre 2 tareas:
1. fnqueue_run(): La estructura de colas provista por la cátedra. En ella se almacena constantemente la función adc_loop().
2. mostrar(): Que según el modo seleccionado llama a funciones de sensor.c que devuelven los resultados buscados. Estos resultados son almacenados dentro de variables en sensor.c.
   Los resultados son actualizados gracias a la función adc_loop que constantemente se manda a la estructura de colas provista por la cátedra. En ella se elige un canal y manda una señal para

realizar una conversión simple.
Cuando la conversión finaliza, se procesa el resultado y se guarda en un arreglo de 100 elementos en *sensor.c*.

El main utiliza las funciones de sensor.h device.h y teclado.h.
device.h al ser el modulo donde se manipula la cola depende de fnqueue.h.
Sensor.h y teclado.h reservan sendos espacios en memoria con una estructura similar definida en device.h, por lo que dependen de ella.

Actividad 2

Se agregaron dos pasos al ciclo principal:

1. recibirModo(): Función que lee el puerto Serial y sensa si recibió mensajes. Si recibió un mensaje cambia de modo en el orden:
   Actual → Máximo - > Mínimo → Promedio → Actual
2. enviarMuestras(): Luego de obtener las señales del sensor y del teclado, las muestras procesadas del sensor de temperatura se envían a la GUI de processing.

Para que el Arduino y la interfaz se comuniquen se estableció el siguiente criterio:
Cada vez que se presiona el botón MODO en la GUI, se escribe en el Serial un caracter ‘X’. Cada vez que se ejecuta una vez la función *loop* en el programa hecho en el archivo ino de Arduino se lee el Serial. Si el Serial lee un caracter ‘X’ el modo cambia de la forma circular descripta anteriormente.

Se adaptó el ejemplo provisto por la cátedra de la siguiente forma:

- Se alteró el tamaño de la pantalla
- Se utilizaron 4 ScrollingFcnPlot en lugar de 2 para representar en pantalla los valores de los 4 modos.
- Se eliminó uno de los botones y se le dio otra funcionalidad al otro. Ahora cada vez que se hace click en el botón se manda un mensaje al Arduino para que cambie de modo.
- Se agregó la función setCaption a la clase Button. Esta función actualiza el valor de la variable caption si el parámetro no está vacío. Se agregó esta función para actualizar de una manera más cómoda las labels que representan las distintas temperaturas.