

Segundo Cuatrimestre 2018

ELECTRÓNICA
DIGITAL 2

MONITOREO DE SISTEMA INDOOR



Índice

Introducción.....	2
Marco Teórico.....	2
Estructura EDU-CIAA.....	3
Datasheet EDU-CIAA.....	3
Puertos utilizados.....	3
Sensores Utilizados.....	4
Diagrama en Bloques.....	4
Desarrollo.....	4
Diagrama en Bloques del Programa.....	5
Esquemático.....	6
Código.....	6
Resultados Experimentales.....	14
Conclusiones.....	14
Problemáticas.....	15
Bibliografía.....	15
Apendice.....	Error! Bookmark not defined.

Introducción

El presente trabajo detalla los dispositivos utilizados para crear un sistema de control de temperatura y humedad indoor digital, con el uso de una placa EDU-CIAA. Este tiene la posibilidad de medir los parámetros mencionados anteriormente mediante dos sensores, uno de ellos de entrada directa (temperatura) a la EDU-CIAA y otro mediante un circuito adaptador de impedancia; el mismo viene integrado en el sensor de humedad propiamente dicho. Se detalla el código utilizado dando una breve explicación de cada bloque junto con su correspondiente diagrama de flujo sobre el funcionamiento de la placa y sus periféricos. El resultado se podrá visualizar mediante la consola del debug. Adicionalmente, pero, sin ser parte de nuestro proyecto, se decidió colocar un módulo Bluetooth (UART) que permita la comunicación a una app, y así visualizar los datos obtenidos, mediante aplicación, en la pantalla del celular.

Marco Teórico

Para el presente trabajo práctico se utilizó una placa EDU-CIAA basada en la CIAA-NXP con un microcontrolador LPC4337 con un dual Core ARM de Cortex-M4F y Cortex-M0.

Adicionalmente a las clases tanto teóricas como prácticas se facilitaron los textos guía para el CórteX M3 y M4 (The Definitive Guide to ARM CORTEX –M3 AND CORTEX –M4 PROCESSORS) así como otros para comprender la arquitectura del mismo (Patterson Hennessy -Computer Architecture a Quantitative Approach (1990)).

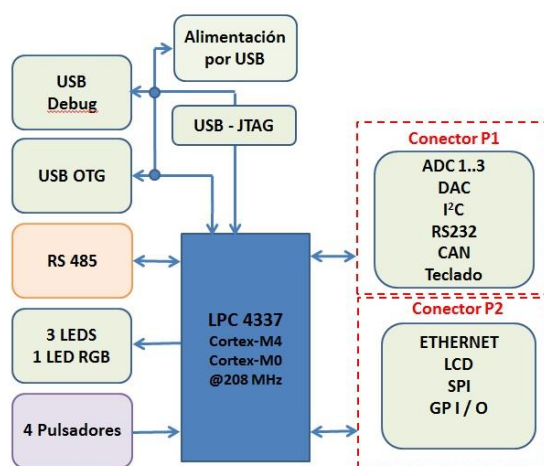


Ilustración 1: Diagrama en Bloques EDU-CIAA

Estructura EDU-CIAA

- 2 puertos micro-USB (uno para aplicaciones y Debugging, otro para alimentación).
- 4 salidas digitales implementadas con leds RGB.
- 4 entradas digitales con pulsadores.
- 1 puerto de comunicaciones RS 485 con bornera.
- 2 conectores de expansión:
 - 1:
 - 3 entradas analógicas (ADC0_1,2y3),
 - 1 salida analógica (DAC0),
 - 1 puerto I2C,
 - 1 puerto asincrónico full duplex (para RS-232).
 - 1 puerto CAN,
 - 1 conexión para un teclado de 3×4,
 - 2:
 - 1 puerto Ethernet.
 - 1 puerto SPI.
 - 1 puerto para Display LCD con 4 bits de datos, Enable y RS.
 - 9 pines genéricos de I/O.

Datasheet EDU-CIAA

- Placa: EDU-CIAA V1.1 NXP
- Microcontrolador: LPC4337 ARM Cortex-M4/M0
- Clock: Hasta 204 MHz
- DMIPS: 430
- Memoria Flash: 1024 kB
- Memoria RAM: 136 kB
- Memoria EEPROM: 16 kB
- ADC: 8 Canales, 10 bits a 400 ksps
- Timers: 41 de 32 bits
- Alimentación: 3.3V

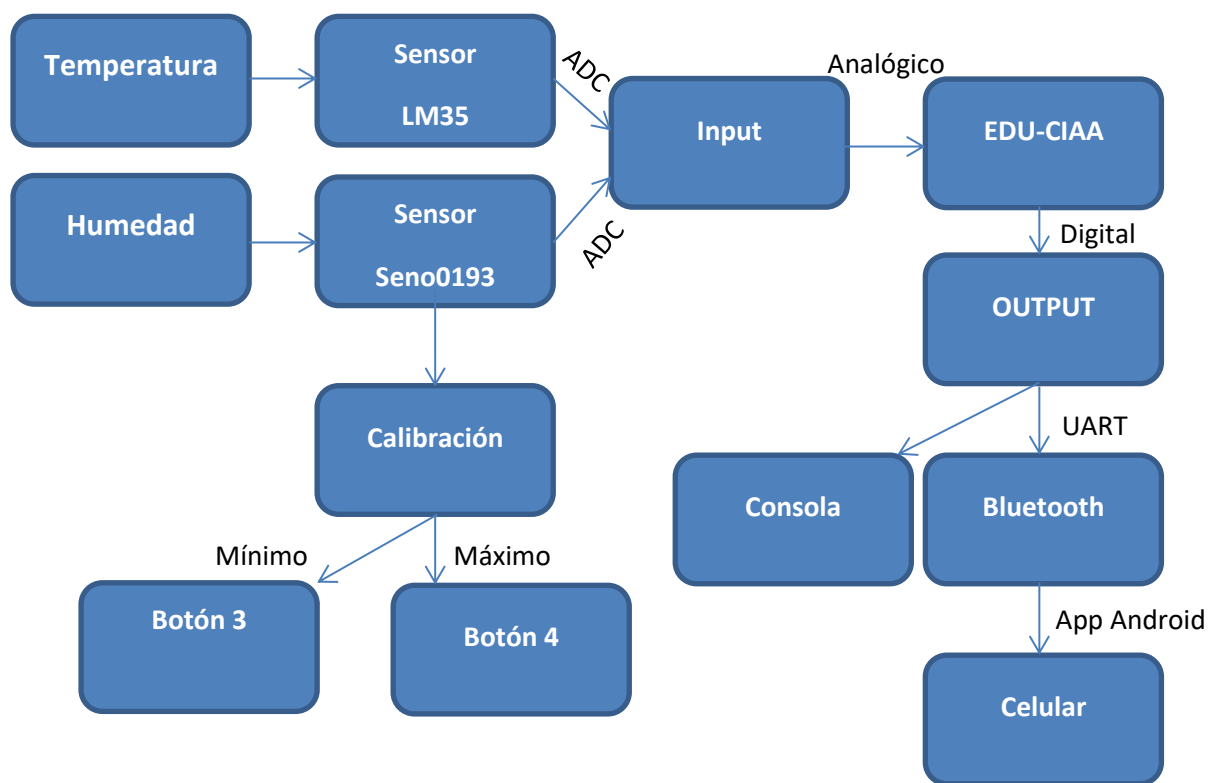
Puertos utilizados

- 1 puerto micro-USB
- 2 entradas digital con pulsador, para activar calibración del Sensor de Humedad
- 2 entradas analógicas ADC
- 1 puerto UART (serie) para conexión con el módulo Bluetooth

Sensores Utilizados

- Sensor de Temperatura LM35 (Anexo Datasheet)
- Sensor de Humedad Capacitivo Sen0193 (Anexo Datasheet)
- Modulo Bluetooth UART HC-06 (Anexo Datasheet)

Diagrama en Bloques



Desarrollo

El programa se escribió en lenguaje C para luego ser compilado y cargado en la placa. En primera instancia se definen las variables a utilizar y las direcciones, a fin de configurar únicamente los módulos necesarios para el proyecto.

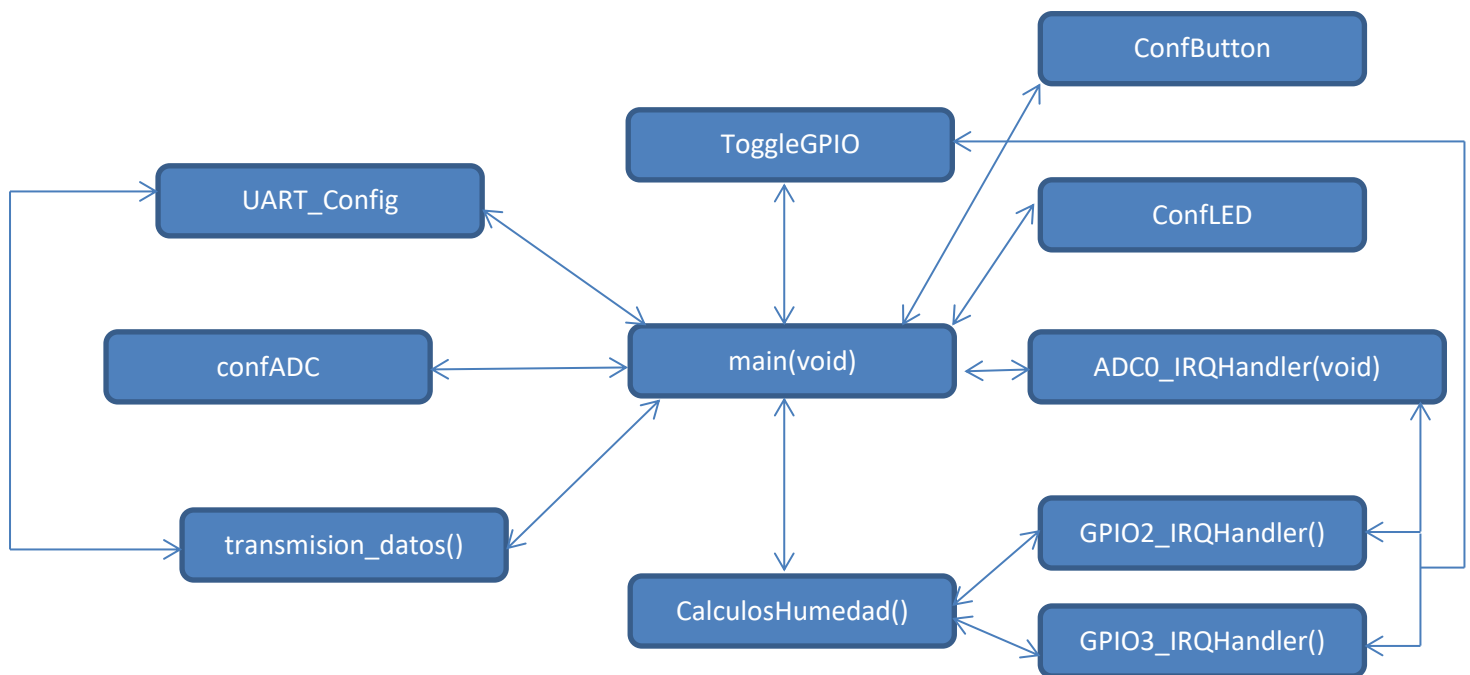
La adquisición de los datos se realizará mediante los puertos ADC de la EDU-CIAA, los mismos fueron configurados de forma de tomar muestras analógicas entregadas por ambos sensores, estos nos proporcionan un valor de tensión correlacionado con el valor de la magnitud física en cuestión (temperatura-humedad).

A su vez, se programó la calibración del sensor de la humedad, de esta forma, mediante dos pulsadores establecemos los límites (superior-inferior) de medición, esto es necesario ya que estaremos visualizando valores relativos.

Una vez definidas todas nuestras variables, puertos de entrada y salida, se procede al desarrollo de las funciones. En el diagrama de flujo anteriormente plasmado se puede interpretar la interacción entre cada uno de los bloques.

Cabe destacar, tal como se mencionó en la introducción, que dicho proyecto se centra en la obtención de los valores de temperatura y humedad de nuestro ambiente. Tratándose de un sistema que puede ser utilizado tanto en un lugar fijo o de traslado; nos pareció oportuno incorporar un módulo de transmisión Bluetooth (UART), a partir del cual podremos enviar la información recopilada a un celular y visualizarla mediante una aplicación (Electronics Bluetooth-Android). Para comunicar nuestra placa central y el módulo de transmisión se utilizaron las librerías de LPCOpen.

Diagrama en Bloques del Programa



Esquemático

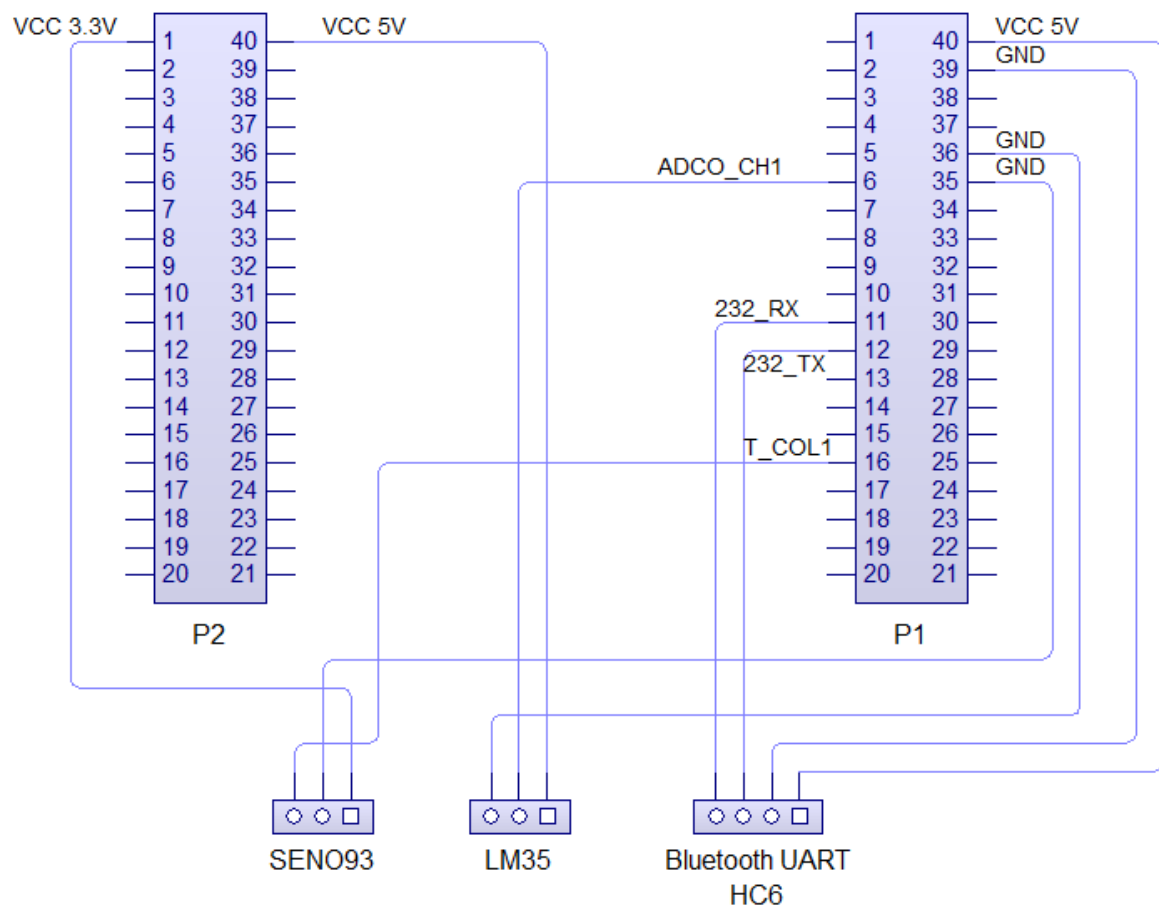


Ilustración 2: Esquemático de conexión

Código

```
#include "chip.h"

#include "Directions.h"

// Variables usada para almacenar el valor convertido

static int dataADC_1,dataADC_2, dataADC_3, dataADC_4;

static unsigned char menu0[] = "La temperatura es:  *C*\n\r";

static unsigned char Temperatura[] = "*T  * \n\r";

static unsigned char menu1[] = "La humedad es de:  % \n\r";

static unsigned char Humedad[] = "*H  * \n\r";
```

```
float temp0, hum0, MinHumedad=865, MaxHumedad=323, Pendiente=0, Ordenada=0 ;
```

```
static char flagADC = 0;
```

```
unsigned char buff0, buff1;
```

```
void ConfButton(int Dir_0, int Dir_1, int Port, int Pin, int Func, int Sel_Interrupt) {
```

```
    SCU_PORT->SFSP[Dir_0][Dir_1] = (BOTONES_MASK | Func); //func
```

```
    GPIO_PORT->DIR[Port] &= ~(1 << Pin); //pin boton
```

```
    SCU_PORT->PINTSEL[0] |= ((Port << 5) << (Sel_Interrupt * 8)) | ((Pin << 0) << (Sel_Interrupt * 8)); //Seleccion/Activacion de interrupcion por PIN
```

```
    GPIO_INT_PORT->ISEL &= ~GPIO_PIN_INT_MASK;
```

```
    GPIO_INT_PORT->SIENR = GPIO_PIN_INT_MASK;
```

```
    NVIC_PORT->ISER1 |= (1 << Sel_Interrupt); //Colocamos un 1 habilitando la interrupcion del GPIO para el Port elegido //port
```

```
}
```

```
void ConfLED(int Dir_0, int Dir_1, int Port, int Pin, int Func) {
```

```
    SCU_PORT->SFSP[Dir_0][Dir_1] = (LEDS_MASK | Func);
```

```
    GPIO_PORT->DIR[Port] |= (1 << Pin);
```

```
}
```

```
void ToggleGPIO(int Port, int Pin) {
```

```
    GPIO_PORT->NOT[Port] |= (1 << Pin); //Nego el valor del led que ya tiene (Si esta encendido lo apaga si esta apagado lo enciende)
```

```
}
```

```
void UART_Config(void){
```

```
    SCU_PORT->SFSP[2][3] = (SCU_MODE_DES | SCU_MODE_FUNC2); //Seteo el P2_3 Funcion 2(U3_TXD) y deshabilito los pullup/pulldown
```



```

SCU_PORT->SFSP[2][4] = (SCU_MODE_DES | ~SCU_MODE_EZI | SCU_MODE_ZIF_DIS |
SCU_MODE_FUNC2); //Seteo el P2_4 Funcion 2(U3_RXD) deshabilito: GLICH FILTER, los
pullup/pulldown Habilito: Buffer entrada

```

```

/* Setup UART for 9600 */

```

```

Chip_UART_Init(LPC_USART3);

```

```

Chip_UART_SetBaud(LPC_USART3, 9600);

```

```

Chip_UART_ConfigData(LPC_USART3, (UART_LCR_WLEN8 | UART_LCR_SBS_1BIT |
UART_LCR_PARITY_DIS)); //UART word length select: 8 bit data mode | UART stop bit select: 1 stop
bit | UART Parity Disable

```

```

Chip_UART_TXEnable(LPC_USART3); //Habilito la transmision

```

```

}

```

```

void confADC(){

```

```

    //CH0 (TFIL_3)

```

```

    // SCU_PORT->SFSP[4][3] = (ADC_MASK | SCU_MODE_FUNC0); //seleciono el pin 4_3 y
Deshabilito pullup y pulldown,EZI=0,Funcion 0

```

```

    // GPIO_PORT->DIR[2] |= (1 << 3); //Seteo el pin como INPUT GPIO2(3)

```

```

    //CH1 (Humedad)(TFIL_1)

```

```

    //SCU_PORT->SFSP[4][1] = (ADC_MASK | SCU_MODE_FUNC0); //seleciono el pin 4_1 y
Deshabilito pullup y pulldown,EZI=0,Funcion 0

```

```

    //GPIO_PORT->DIR[2] |= (1 << 1); //Seteo el pin como INPUT GPIO2(1)

```

```

    //CH3 (Temperatura)(TCOL_2)

```

```

    SCU_PORT->SFSP[7][5] = (ADC_MASK | SCU_MODE_FUNC0); //seleciono el pin 7_5 y
Deshabilito pullup y pulldown,EZI=0,Funcion 0

```

```

    GPIO_PORT->DIR[3] |= (1 << 13); //Seteo el pin como INPUT GPIO3(13)

```

```

//CH4 (Humedad)(TCOL_1)

SCU_PORT->SFSP[7][4] = (ADC_MASK | SCU_MODE_FUNC0); //seleciono el pin 7_4 y
Deshabilito pullup y pulldown,EZI=0,Funcion 0

GPIO_PORT->DIR[3] |= (1 << 13); //Seteo el pin como INPUT GPIO3(13)


SCU_PORT->ENAI0[0] |= (1 << 3)|(1 << 4); //1 en bit 3 y 1 en bit 4 para habilitar los canales
3 y 4 del ADC0 en modo analogico


ADC0_PORT->INTEN |= (1 << 1)|(1 << 4); //habilita los bits DONE FLAG de todas las
interrupciones de los canales del ADC


ADC0_PORT->CR = 0; //LO PONGO EN CERO, PARA REALIZAR LA CONFIGURACION INICIAL A
GUSTO


ADC0_PORT->CR |= (1 << 1)|(1 << 4) // SEL
                        | (0xFF << 8) // CLKDIV
                        | (0x1 << 16) // BURST
                        | (0x0 << 17) // CLKS
                        | (0x1 << 21) // POWERMODE
                        | (0x0 << 24) // START
                        | (0x0 << 27); // EDGE


NVIC_PORT->ISER0 |= (1 << 17); //Habilitacion de la interrupcion del ADC0 en el NVIC

NVIC_PORT->ICPR0 |= (1 << 17); //Limpieza de la interrupcion pendiente del ADC0 en el
NVIC

}


//Rutina de interrupción del ADC

void ADC0_IRQHandler(void){

```

```

    NVIC_DisableIRQ(ADC0_IRQn);

    flagADC = 1;

}

```

//Rutina de interrupción del BOTON 3

```

void GPIO2_IRQHandler() {

```

```

    if (ADC0_PORT->DR[4] & (1<<31)){

        MinHumedad=(float)dataADC_4;

    }

    CalculosHumedad();

    ToggleGPIO(1, 11); //Enciende/Apaga el led

    GPIO_INT_PORT->IST = (1 << 2); //Borra el estado peticion de la IRQ2

}

```

//Rutina de interrupción del BOTON 4

```

void GPIO3_IRQHandler() {

```

```

    if (ADC0_PORT->DR[4] & (1<<31)){

        MaxHumedad=(float)dataADC_4;

    }

    CalculosHumedad();

    ToggleGPIO(1, 12); //Enciende/Apaga el led

    GPIO_INT_PORT->IST = (1 << 3); //Borra el estado peticion de la IRQ3

}

```

//Rutina de procesamiento de medicion y transmision de datos

```

void transmision_datos(){

```

```

    //ETAPA SENSOR HUMEDAD

```

```

    if (ADC0_PORT->DR[4] & (1<<31)){

```

```

ToggleGPIO(5, 0); //Enciende/Apaga el led

dataADC_4 = (((ADC0_PORT->DR[4]) >> 6) & 0x3FF);

hum0 = (float)dataADC_4;

hum0 = Pendiente*hum0 + Ordenada; //Ecuacion de pendiente de humedad con
respecto a maximo y minimo definidos en variables aire y agua

buff1 = (uint8_t)(hum0/100) + 48;

    if(buff1 > 48){

        menu1[17] = buff1;

        Humedad[2] = hum0;

        hum0 = hum0-(buff0-48)*100;

    }else{

        menu1[17] = 48;

        Humedad[2] = 48 ;

    }

buff1 = (uint8_t)(hum0/10) + 48;

    if(buff1 > 48){

        menu1[18] = buff1;

        Humedad[3] = buff1;

        hum0 = hum0-(buff1-48)*10;

    }else{

        menu1[18] = 48;

        Humedad[3] = 48 ;

    }

buff1 = (uint8_t)(hum0) + 48;

menu1[19] = buff1;

Humedad[4] = buff1;

Chip_UART_SendBlocking(LPC_USART3, &menu1, sizeof(menu1));

Chip_UART_SendBlocking(LPC_USART3, &Humedad, sizeof(Humedad));

```

```

        printf("%s", Humedad);
    }

//ETAPA SENSOR TEMPERATURA

    if (ADC0_PORT->DR[1] & (1<<31)){

        ToggleGPIO(5, 1); //Enciende/Apaga el led

        dataADC_1 = (((ADC0_PORT->DR[1]) >> 6) & 0x3FF);

        temp0 = (float)dataADC_1;

        temp0 = (temp0/1023)*330;

        buff0 = (uint8_t)(temp0/100) + 48;

        if(buff0 > 48){

            menu0[19] = buff0;

            Temperatura[2] = buff0;

            temp0 = temp0-(buff0-48)*100;

        }else{

            menu0[19] = 48;

            Temperatura[2] = 48;

        }

        buff0 = (uint8_t)(temp0/10) + 48;

        if(buff0 > 48){

            menu0[20] = buff0;

            Temperatura[3] = buff0;

            temp0 = temp0-(buff0-48)*10;

        }else{

            menu0[20] = 48;

            Temperatura[3] = 48;

        }

        buff0 = (uint8_t)(temp0) + 48;
    }

```

```

        menu0[21] = buff0;

        Temperatura[4] = buff0;

        Chip_UART_SendBlocking(LPC_USART3, &menu0, sizeof(menu0));

        Chip_UART_SendBlocking(LPC_USART3, &Temperatura, sizeof(Temperatura));

        printf("%s", Temperatura);

    }

    flagADC = 0;

    NVIC_PORT->ISER0 |= (1 << 17); //Habilitacion de la interrupcion del ADC0 en el NVIC
}

```

//Rutina de Caculo de parametros para ecuacion de humedad

```

void CalculosHumedad(){

    Pendiente=(100)/(MaxHumedad-MinHumedad);

    Ordenada= 100 - (Pendiente*MaxHumedad);

}

```

int main(void){

```

    ConfLED(2, 0, 5, 0, 4); //Configuramos el SCU P2_0, GPIO5 puerto 5 pin 0 con la funcion 4
    ConfLED(2, 1, 5, 1, 4); //Configuramos el SCU P2_1, GPIO5 puerto 5 pin 1 con la funcion 4
    ConfLED(2, 2, 5, 2, 4); //Configuramos el SCU P2_2, GPIO5 puerto 5 pin 2 con la funcion 4
    ConfLED(2, 10, 0, 14, 0); //Configuramos el SCU P2_10, GPIO0 puerto 0 pin 14 con la funcion
0
    ConfLED(2, 11, 1, 11, 0); //Configuramos el SCU P2_11, GPIO1 puerto 1 pin 11 con la funcion
0
    ConfLED(2, 12, 1, 12, 0); //Configuramos el SCU P2_12, GPIO1 puerto 1 pin 12 con la funcion
0

```

```
ConfButton(1, 2, 0, 9, 0, 2); //Direccion 1_2; Puerto 0; Pin 9; Funcion 0 //Puls2 Pinout
//interrupcion n2 (BOTON PARA CALIBRAR HUMEDAD MINIMO-SECO)
```

```
ConfButton(1, 6, 1, 9, 0, 3); //Direccion 1_6; Puerto 1; Pin 9; Funcion 0 //Puls3 Pinout
//interrupcion n3 (BOTON PARA CALIBRAR HUMEDAD MAXIMO-AGUA)
```

```
CalculosHumedad();
```

```
UART_Config();
```

```
confADC();
```

```
while (1) {
```

```
    if(flagADC){
```

```
        transmision_datos();
```

```
    }
```

```
}
```

```
}
```

Resultados Experimentales

Se realizan pruebas experimentales para poder evaluar su desempeño; y para ello en paralelo se colocaron sensores anteriormente trazados. Para el caso del sensor de temperatura se observó un desvío de $\pm 0,5^{\circ}\text{C}$ con respecto a nuestro sensor patrón; y para el caso del sensor de humedad no se observó discrepancia, ya que el usuario es quien establecen los valores límite y tal magnitud se expresa de forma relativa (%)

Conclusiones

Hemos llegado a conclusión de que el ADC de la EDU-CIAA posee una resolución apropiada y velocidad de muestro suficiente para tal proyecto. Se observó, cómo posibilidad de mejora, la posibilidad de complejizarlo en el corto plazo, añadiendo control de riego y circulación de aire (invernadero automatizado).

Problemáticas

Al momento de realizar las pruebas con el sensor de humedad (YL-69), nos encontramos con una dificultad, dicho sensor cuenta con dos salidas; una digital que indica si la humedad está por arriba o debajo de un nivel prefijado mediante un preset; y 1 salida analógica que indica la humedad del suelo por medio de un valor de tensión que varía entre 0V y la tensión de alimentación (la tensión disminuye con la humedad del suelo). Intentamos utilizar la salida analógica, ya que era la indicada para nuestro fin, por temas de impedancia se cargaba la salida de nuestro sensor y no permitía realizar la adquisición de datos. Se procedió a cambiar de sensor por uno capacitivo (SEN093), que adicionalmente traía acoplada a su salida un AO, el cual, cumple la función de adaptador de impedancia.

Bibliografía

- Patterson Hennessy, Computer Architecture a Quantitative Approach, 1990
- Elsevier, The Definitive Guide to ARM Cortex –M3 and Cortex –M4 Processors, Third Edition
- <https://www.nxp.com/support/developer-resources/software-development-tools/lpc-developer-resources-/lpcopen-libraries-and-examples:LPC-OPEN-LIBRARIES>

Apéndice

```
26          .loc 1 307 0
27          .cfi_startproc
28          @ args = 0, pretend = 0, frame = 8
29          @ frame_needed = 1, uses_anonymous_args = 0
30          @ link register save eliminated.
31 0000 80B4          push    {r7}
32          .cfi_def_cfa_offset 4
33          .cfi_offset 7, -4
34 0002 83B0          sub     sp, sp, #12
35          .cfi_def_cfa_offset 16
36 0004 00AF          add     r7, sp, #0
37          .cfi_def_cfa_register 7
38 0006 7860          str     r0, [r7, #4]
```


308:C:\Users\ferna\Desktop\WP2\lpc_chip_43xx\inc\uart_18xx_43xx.h **** pUART->TER2 =
UART_TER2_TXEN;

39 .loc 1 308 0

40 0008 7B68 ldr r3, [r7, #4]

41 000a 0122 movs r2, #1

42 000c DA65 str r2, [r3, #92]

309:C:\Users\ferna\Desktop\WP2\lpc_chip_43xx\inc\uart_18xx_43xx.h **** }

43 .loc 1 309 0

44 000e 00BF nop

45 0010 0C37 adds r7, r7, #12

46 .cfi_def_cfa_offset 4

47 0012 BD46 mov sp, r7

48 .cfi_def_cfa_register 13

49 @ sp needed

50 0014 5DF8047B ldr r7, [sp], #4

51 .cfi_restore 7

52 .cfi_def_cfa_offset 0

53 0018 7047 bx lr

54 .cfi_endproc

55 .LFE427:

57 .section.text.Chip_UART_ConfigData,"ax",%progbits

58 .align 1

59 .syntax unified

60 .thumb

61 .thumb_func

62 .fpu fpv4-sp-d16

64 Chip_UART_ConfigData:

65 .LFB436:

66 .loc 1 426 0

67 .cfi_startproc

```

68          @ args = 0, pretend = 0, frame = 8
69          @ frame_needed = 1, uses_anonymous_args = 0
70          @ link register save eliminated.
71 0000 80B4          push    {r7}
72          .cfi_def_cfa_offset 4
73          .cfi_offset 7, -4
74 0002 83B0          sub     sp, sp, #12
75          .cfi_def_cfa_offset 16
76 0004 00AF          add     r7, sp, #0
77          .cfi_def_cfa_register 7
78 0006 7860          str     r0, [r7, #4]
79 0008 3960          str     r1, [r7]
427:C:\Users\ferna\Desktop\WP2\lpc_chip_43xx\inc\uart_18xx_43xx.h **** pUART->LCR = config;
80          .loc 1 427 0
81 000a 7B68          ldr     r3, [r7, #4]
82 000c 3A68          ldr     r2, [r7]
83 000e DA60          str     r2, [r3, #12]
428:C:\Users\ferna\Desktop\WP2\lpc_chip_43xx\inc\uart_18xx_43xx.h **** }
84          .loc 1 428 0
85 0010 00BF          nop
86 0012 0C37          adds   r7, r7, #12
87          .cfi_def_cfa_offset 4
88 0014 BD46          mov     sp, r7
89          .cfi_def_cfa_register 13
90          @ sp needed
91 0016 5DF8047B      ldr     r7, [sp], #4
92          .cfi_restore 7
93          .cfi_def_cfa_offset 0
94 001a 7047          bx     lr
95          .cfi_endproc

```

```

96          .LFE436:
97
98          .section.bss.dataADC_1,"aw",%nobits
99          .align 2
100
101          dataADC_1:
102          0000 00000000          .space 4
103
104          .section.bss.dataADC_2,"aw",%nobits
105          .align 2
106
107          dataADC_2:
108          0000 00000000          .space 4
109
110          .section.bss.dataADC_3,"aw",%nobits
111          .align 2
112
113          dataADC_3:
114          0000 00000000          .space 4
115
116          .section.bss.dataADC_4,"aw",%nobits
117          .align 2
118
119          dataADC_4:
120          0000 00000000          .space 4
121
122          .section.data.menu0,"aw",%progbits
123          .align 2
124
125          menu0:
126          0000 4C612074          .ascii "La temperatura es: *C*\012\015\000"
127          656D7065
127          72617475
127          72612065
127          733A2020
128
129          .section.data.Temperatura,"aw",%progbits
130          .align 2
131
132          Temperatura:
133          0000 2A542020          .ascii "*T * \012\015\000"
134          202A200A

```

```

133 0D00
134          .section.data.menu1,"aw",%progbits
135          .align 2
138      menu1:
139 0000 4C612068          .ascii  "La humedad es de: % \012\015\000"
139 756D6564
139 61642065
139 73206465
139 3A202020
140          .section.data.Humedad,"aw",%progbits
141          .align 2
144      Humedad:
145 0000 2A482020          .ascii  "**H * \012\015\000"
145 202A200A
145 0D00
146          .global temp0
147          .section.bss.temp0,"aw",%nobits
148          .align 2
151      temp0:
152 0000 00000000          .space 4
153          .global hum0
154          .section.bss.hum0,"aw",%nobits
155          .align 2
158      hum0:
159 0000 00000000          .space 4
160          .global MinHumedad
161          .section.data.MinHumedad,"aw",%progbits
162          .align 2
165      MinHumedad:
166 0000 00405844          .word 1146634240

```

```

167             .global MaxHumedad
168             .section.data.MaxHumedad,"aw",%progbits
169             .align 2
172         MaxHumedad:
173 0000 0080A143             .word 1134657536
174             .global Pendiente
175             .section.bss.Pendiente,"aw",%nobits
176             .align 2
179         Pendiente:
180 0000 00000000             .space 4
181             .global Ordenada
182             .section.bss.Ordenada,"aw",%nobits
183             .align 2
186         Ordenada:
187 0000 00000000             .space 4
188             .section.bss.flagADC,"aw",%nobits
191         flagADC:
192 0000 00             .space 1
193             .global buff0
194             .section.bss.buff0,"aw",%nobits
197         buff0:
198 0000 00             .space 1
199             .global buff1
200             .section.bss.buff1,"aw",%nobits
203         buff1:
204 0000 00             .space 1
205             .section.text.ConfButton,"ax",%progbits
206             .align 1
207             .global ConfButton
208             .syntax unified

```

```

209             .thumb
210             .thumb_func
211             .fpu fpv4-sp-d16
213     ConfButton:
214     .LFB479:
215             .file 2 "../src/Final.c"
216
217     1:../src/Final.c **** #include "chip.h"
218
219     2:../src/Final.c **** #include "Final.h"
220
221     3:../src/Final.c ****
222
223     4:../src/Final.c ****
224
225     5:../src/Final.c **** // Variables usada para almacenar el valor convertido
226
227     6:../src/Final.c **** static int dataADC_1, dataADC_2, dataADC_3, dataADC_4;
228
229     7:../src/Final.c **** static unsigned char menu0[] = "La temperatura es:  *C*\n\r";
230
231     8:../src/Final.c **** static unsigned char Temperatura[] = "*T  * \n\r";
232
233     9:../src/Final.c **** static unsigned char menu1[] = "La humedad es de:  % \n\r";
234
235     10:../src/Final.c **** static unsigned char Humedad[] = "*H  * \n\r";
236
237     11:../src/Final.c **** float temp0, hum0, MinHumedad=865, MaxHumedad=323, Pendiente=0,
Ordenada=0 ;
238
239     12:../src/Final.c **** static char flagADC = 0;
240
241     13:../src/Final.c **** unsigned char buff0, buff1;
242
243     14:../src/Final.c ****
244
245     15:../src/Final.c **** void ConfButton(int Dir_0, int Dir_1, int Port, int Pin, int Func, int
Sel_Interrupt) {
246
247     216             .loc 2 15 0
248
249     217             .cfi_startproc
250
251     218             @ args = 8, pretend = 0, frame = 16
252
253     219             @ frame_needed = 1, uses_anonymous_args = 0
254
255     220             @ link register save eliminated.
256
257     221 0000 80B4             push    {r7}
258
259     222             .cfi_def_cfa_offset 4
260
261     223             .cfi_offset 7, -4

```

```

224 0002 85B0          sub    sp, sp, #20
225                  .cfi_def_cfa_offset 24
226 0004 00AF          add    r7, sp, #0
227                  .cfi_def_cfa_register 7
228 0006 F860          str     r0, [r7, #12]
229 0008 B960          str     r1, [r7, #8]
230 000a 7A60          str     r2, [r7, #4]
231 000c 3B60          str     r3, [r7]

16:../src/Final.c **** SCU_PORT->SFSP[Dir_0][Dir_1] = (BOTONES_MASK | Func);//func
232                  .loc 2 16 0
233 000e 2449          ldr     r1, .L4
234 0010 FB68          ldr     r3, [r7, #12]
235 0012 5A01          lsls    r2, r3, #5
236 0014 BB68          ldr     r3, [r7, #8]
237 0016 1344          add     r3, r3, r2
238 0018 4FF0FF32      mov     r2, #-1
239 001c 41F82320      str     r2, [r1, r3, lsl #2]

17:../src/Final.c **** GPIO_PORT->DIR[Port] &= ~(1 << Pin);//pin boton
240                  .loc 2 17 0
241 0020 204A          ldr     r2, .L4+4
242 0022 7B68          ldr     r3, [r7, #4]
243 0024 03F50063      add     r3, r3, #2048
244 0028 52F82320      ldr     r2, [r2, r3, lsl #2]
245 002c 0121          movs    r1, #1
246 002e 3B68          ldr     r3, [r7]
247 0030 01FA03F3      lsl     r3, r1, r3
248 0034 DB43          mvns    r3, r3
249 0036 1B49          ldr     r1, .L4+4
250 0038 1A40          ands    r2, r2, r3
251 003a 7B68          ldr     r3, [r7, #4]

```

```

252 003c 03F50063      add    r3, r3, #2048
253 0040 41F82320      str    r2, [r1, r3, lsl #2]

18:../src/Final.c **** SCU_PORT->PINTSEL[0] |= ((Port << 5) << (Sel_Interrupt * 8)) | ((Pin << 0) <<
(Sel_Interrupt * 8))

254                  .loc 2 18 0
255 0044 164B          ldr    r3, .L4
256 0046 D3F8002E      ldr    r2, [r3, #3584]
257 004a 7B68          ldr    r3, [r7, #4]
258 004c 5901          lsls   r1, r3, #5
259 004e FB69          ldr    r3, [r7, #28]
260 0050 DB00          lsls   r3, r3, #3
261 0052 9940          lsls   r1, r1, r3
262 0054 FB69          ldr    r3, [r7, #28]
263 0056 DB00          lsls   r3, r3, #3
264 0058 3868          ldr    r0, [r7]
265 005a 00FA03F3      lsl    r3, r0, r3
266 005e 0B43          orrs   r3, r3, r1
267 0060 0F49          ldr    r1, .L4
268 0062 1343          orrs   r3, r3, r2
269 0064 C1F8003E      str    r3, [r1, #3584]

19:../src/Final.c **** GPIO_INT_PORT->ISEL &= ~GPIO_PIN_INT_MASK;

270                  .loc 2 19 0
271 0068 0F4B          ldr    r3, .L4+8
272 006a 1B68          ldr    r3, [r3]
273 006c 0E4A          ldr    r2, .L4+8
274 006e 23F00103      bic    r3, r3, #1
275 0072 1360          str    r3, [r2]

20:../src/Final.c **** GPIO_INT_PORT->SIENR = GPIO_PIN_INT_MASK;

276                  .loc 2 20 0
277 0074 0C4B          ldr    r3, .L4+8

```



```

278 0076 0F22          movs   r2, #15
279 0078 9A60          str     r2, [r3, #8]

21:../src/Final.c **** NVIC_PORT->ISER1 |= (1 << Sel_Interrupt); //Colocamos un 1 habilitando la
interrupcion del GPIO pa

280                  .loc 2 21 0

281 007a 4FF0E023       mov     r3, #-536813568
282 007e D3F80421       ldr     r2, [r3, #260]
283 0082 0121          movs   r1, #1
284 0084 FB69          ldr     r3, [r7, #28]
285 0086 01FA03F3       lsl     r3, r1, r3
286 008a 4FF0E021       mov     r1, #-536813568
287 008e 1343          orrs   r3, r3, r2
288 0090 C1F80431       str     r3, [r1, #260]

22:../src/Final.c **** }

289                  .loc 2 22 0

290 0094 00BF          nop
291 0096 1437          adds   r7, r7, #20
292                  .cfi_def_cfa_offset 4
293 0098 BD46          mov     sp, r7
294                  .cfi_def_cfa_register 13
295                  @ sp needed
296 009a 5DF8047B       ldr     r7, [sp], #4
297                  .cfi_restore 7
298                  .cfi_def_cfa_offset 0
299 009e 7047          bx      lr
300                  .L5:
301                  .align   2
302                  .L4:
303 00a0 00600840       .word   1074290688
304 00a4 00400F40       .word   1074741248

```

```

305 00a8 00700840      .word  1074294784
306                  .cfi_endproc
307      .LFE479:
309                  .section.text.ConfLED,"ax",%progbits
310                  .align  1
311                  .global  ConfLED
312                  .syntax unified
313                  .thumb
314                  .thumb_func
315                  .fpu fpv4-sp-d16
317      ConfLED:
318      .LFB480:
24:../src/Final.c **** void ConfLED(int Dir_0, int Dir_1, int Port, int Pin, int Func) {
319                  .loc 2 24 0
320                  .cfi_startproc
321                  @ args = 4, pretend = 0, frame = 16
322                  @ frame_needed = 1, uses_anonymous_args = 0
323                  @ link register save eliminated.
324 0000 80B4          push   {r7}
325                  .cfi_def_cfa_offset 4
326                  .cfi_offset 7, -4
327 0002 85B0          sub     sp, sp, #20
328                  .cfi_def_cfa_offset 24
329 0004 00AF          add     r7, sp, #0
330                  .cfi_def_cfa_register 7
331 0006 F860          str     r0, [r7, #12]
332 0008 B960          str     r1, [r7, #8]
333 000a 7A60          str     r2, [r7, #4]
334 000c 3B60          str     r3, [r7]
25:../src/Final.c **** SCU_PORT->SFSP[Dir_0][Dir_1] = (LEDS_MASK | Func);

```

```

335                .loc 2 25 0
336 000e 1048      ldr    r0, .L7
337 0010 BB69      ldr    r3, [r7, #24]
338 0012 43F01002  orr    r2, r3, #16
339 0016 FB68      ldr    r3, [r7, #12]
340 0018 5901      lsls   r1, r3, #5
341 001a BB68      ldr    r3, [r7, #8]
342 001c 0B44      add    r3, r3, r1
343 001e 40F82320  str    r2, [r0, r3, lsl #2]

```

26:../src/Final.c **** GPIO_PORT->DIR[Port] |= (1 << Pin);

```

344                .loc 2 26 0
345 0022 0C4A      ldr    r2, .L7+4
346 0024 7B68      ldr    r3, [r7, #4]
347 0026 03F50063  add    r3, r3, #2048
348 002a 52F82320  ldr    r2, [r2, r3, lsl #2]
349 002e 0121      movs   r1, #1
350 0030 3B68      ldr    r3, [r7]
351 0032 01FA03F3  lsl    r3, r1, r3
352 0036 0749      ldr    r1, .L7+4
353 0038 1A43      orrs   r2, r2, r3
354 003a 7B68      ldr    r3, [r7, #4]
355 003c 03F50063  add    r3, r3, #2048
356 0040 41F82320  str    r2, [r1, r3, lsl #2]

```

27:../src/Final.c **** }

```

357                .loc 2 27 0
358 0044 00BF      nop
359 0046 1437      adds   r7, r7, #20
360                .cfi_def_cfa_offset 4
361 0048 BD46      mov    sp, r7
362                .cfi_def_cfa_register 13

```

```

363                @ sp needed
364 004a 5DF8047B        ldr    r7, [sp], #4
365                .cfi_restore 7
366                .cfi_def_cfa_offset 0
367 004e 7047          bx     lr
368                .L8:
369                .align  2
370                .L7:
371 0050 00600840        .word  1074290688
372 0054 00400F40        .word  1074741248
373                .cfi_endproc
374                .LFE480:
376                .section.text.ToggleGPIO,"ax",%progbits
377                .align  1
378                .global ToggleGPIO
379                .syntax unified
380                .thumb
381                .thumb_func
382                .fpu fpv4-sp-d16
384                ToggleGPIO:
385                .LFB481:
29:../src/Final.c **** void ToggleGPIO(int Port, int Pin) {
386                .loc 2 29 0
387                .cfi_startproc
388                @ args = 0, pretend = 0, frame = 8
389                @ frame_needed = 1, uses_anonymous_args = 0
390                @ link register save eliminated.
391 0000 80B4          push   {r7}
392                .cfi_def_cfa_offset 4
393                .cfi_offset 7, -4

```

```

394 0002 83B0          sub    sp, sp, #12
395                  .cfi_def_cfa_offset 16
396 0004 00AF          add     r7, sp, #0
397                  .cfi_def_cfa_register 7
398 0006 7860          str     r0, [r7, #4]
399 0008 3960          str     r1, [r7]

30:../src/Final.c **** GPIO_PORT->NOT[Port] |= (1 << Pin); //Nego el valor del led que ya tiene (Si
esta encendido lo apag

400                  .loc 2 30 0
401 000a 0B4A          ldr     r2, .L10
402 000c 7B68          ldr     r3, [r7, #4]
403 000e 03F50C63      add     r3, r3, #2240
404 0012 52F82320      ldr     r2, [r2, r3, lsl #2]
405 0016 0121          movs   r1, #1
406 0018 3B68          ldr     r3, [r7]
407 001a 01FA03F3      lsl     r3, r1, r3
408 001e 0649          ldr     r1, .L10
409 0020 1A43          orrs   r2, r2, r3
410 0022 7B68          ldr     r3, [r7, #4]
411 0024 03F50C63      add     r3, r3, #2240
412 0028 41F82320      str     r2, [r1, r3, lsl #2]

31:../src/Final.c **** }

413                  .loc 2 31 0
414 002c 00BF          nop
415 002e 0C37          adds   r7, r7, #12
416                  .cfi_def_cfa_offset 4
417 0030 BD46          mov     sp, r7
418                  .cfi_def_cfa_register 13
419                  @ sp needed
420 0032 5DF8047B      ldr     r7, [sp], #4

```

```

421                .cfi_restore 7
422                .cfi_def_cfa_offset 0
423 0036 7047                bx    lr
424                .L11:
425                .align 2
426                .L10:
427 0038 00400F40                .word 1074741248
428                .cfi_endproc
429                .LFE481:
431                .section.text.UART_Config,"ax",%progbits
432                .align 1
433                .global UART_Config
434                .syntax unified
435                .thumb
436                .thumb_func
437                .fpu fpv4-sp-d16
439                UART_Config:
440                .LFB482:
441                33:../src/Final.c **** void UART_Config(void){
442                .loc 2 33 0
443                .cfi_startproc
444                @ args = 0, pretend = 0, frame = 0
445                @ frame_needed = 1, uses_anonymous_args = 0
446                0000 80B5                push    {r7, lr}
447                .cfi_def_cfa_offset 8
448                .cfi_offset 7, -8
449                .cfi_offset 14, -4
450                0002 00AF                add     r7, sp, #0
451                .cfi_def_cfa_register 7

```

34:../src/Final.c **** SCU_PORT->SFSP[2][3] = (SCU_MODE_DES | SCU_MODE_FUNC2); //Seteo el P2_3 Funcion 2(U3_TXD) y deshab

```
451                .loc 2 34 0
452 0004 0C4B          ldr    r3, .L13
453 0006 1222          movs   r2, #18
454 0008 C3F80C21      str     r2, [r3, #268]
```

36:../src/Final.c **** SCU_PORT->SFSP[2][4] = (SCU_MODE_DES | ~SCU_MODE_EZI | SCU_MODE_ZIF_DIS | SCU_MODE_FUNC2); //Seteo

```
455                .loc 2 36 0
456 000c 0A4B          ldr     r3, .L13
457 000e 4FF0FF32      mov     r2, #-1
458 0012 C3F81021      str     r2, [r3, #272]
```

38:../src/Final.c **** /* Setup UART for 9600 */

39:../src/Final.c **** Chip_UART_Init(LPC_USART3);

```
459                .loc 2 39 0
460 0016 0948          ldr     r0, .L13+4
461 0018 FFF7FEFF      bl      Chip_UART_Init
```

41:../src/Final.c **** Chip_UART_SetBaud(LPC_USART3, 9600);

```
462                .loc 2 41 0
463 001c 4FF41651      mov     r1, #9600
464 0020 0648          ldr     r0, .L13+4
465 0022 FFF7FEFF      bl      Chip_UART_SetBaud
```

42:../src/Final.c **** Chip_UART_ConfigData(LPC_USART3, (UART_LCR_WLEN8 | UART_LCR_SBS_1BIT | UART_LCR_PARITY_DIS)); //UA

```
466                .loc 2 42 0
467 0026 0321          movs   r1, #3
468 0028 0448          ldr     r0, .L13+4
469 002a FFF7FEFF      bl      Chip_UART_ConfigData
```

43:../src/Final.c **** Chip_UART_TXEnable(LPC_USART3); //Habilito la transmision

```
470                .loc 2 43 0
471 002e 0348          ldr     r0, .L13+4
```

```

472 0030 FFF7FEFF      bl      Chip_UART_TXEnable
473                  .loc 2 44 0
474 0034 00BF          nop
475 0036 80BD          pop     {r7, pc}
476      .L14:
477                  .align   2
478      .L13:
479 0038 00600840        .word   1074290688
480 003c 00200C40        .word   1074536448
481                  .cfi_endproc
482      .LFE482:
484                  .section.text.confADC,"ax",%progbits
485                  .align   1
486                  .global   confADC
487                  .syntax unified
488                  .thumb
489                  .thumb_func
490                  .fpu fpv4-sp-d16
492      confADC:
493      .LFB483:
45:../src/Final.c ****
46:../src/Final.c **** void confADC(){
494                  .loc 2 46 0
495                  .cfi_startproc
496                  @ args = 0, pretend = 0, frame = 0
497                  @ frame_needed = 1, uses_anonymous_args = 0
498                  @ link register save eliminated.
499 0000 80B4          push    {r7}
500                  .cfi_def_cfa_offset 4
501                  .cfi_offset 7, -4

```



```

502 0002 00AF          add    r7, sp, #0

503                  .cfi_def_cfa_register 7

55:../src/Final.c **** //CH3 (Temperatura)(TCOL_2)

56:../src/Final.c **** SCU_PORT->SFSP[7][5] = (ADC_MASK | SCU_MODE_FUNC0); //seleciono el
pin 7_5 y Deshabilito pullup y

504                  .loc 2 56 0

505 0004 274B          ldr     r3, .L16

506 0006 1022          movs   r2, #16

507 0008 C3F89423      str     r2, [r3, #916]

57:../src/Final.c **** GPIO_PORT->DIR[3] |= (1 << 13); //Seteo el pin como INPUT GPIO3(13)

508                  .loc 2 57 0

509 000c 264B          ldr     r3, .L16+4

510 000e 03F50053      add     r3, r3, #8192

511 0012 0C33          adds   r3, r3, #12

512 0014 1B68          ldr     r3, [r3]

513 0016 2449          ldr     r1, .L16+4

514 0018 43F40052      orr     r2, r3, #8192

515 001c 01F50053      add     r3, r1, #8192

516 0020 0C33          adds   r3, r3, #12

517 0022 1A60          str     r2, [r3]

59:../src/Final.c **** //CH4 (Humedad)(TCOL_1)

60:../src/Final.c **** SCU_PORT->SFSP[7][4] = (ADC_MASK | SCU_MODE_FUNC0); //seleciono el
pin 7_4 y Deshabilito pullup y

518                  .loc 2 60 0

519 0024 1F4B          ldr     r3, .L16

520 0026 1022          movs   r2, #16

521 0028 C3F89023      str     r2, [r3, #912]

61:../src/Final.c **** GPIO_PORT->DIR[3] |= (1 << 13); //Seteo el pin como INPUT GPIO3(13)

522                  .loc 2 61 0

523 002c 1E4B          ldr     r3, .L16+4

524 002e 03F50053      add     r3, r3, #8192

```

```

525 0032 0C33      adds    r3, r3, #12
526 0034 1B68      ldr     r3, [r3]
527 0036 1C49      ldr     r1, .L16+4
528 0038 43F40052   orr     r2, r3, #8192
529 003c 01F50053   add     r3, r1, #8192
530 0040 0C33      adds    r3, r3, #12
531 0042 1A60      str     r2, [r3]

```

63:../src/Final.c **** SCU_PORT->ENAIO[0] |= (1 << 3)|(1 << 4); //1 en bit 3 y 1 en bit 4 para habilitar los canales 3 y

```

532                .loc 2 63 0
533 0044 174B      ldr     r3, .L16
534 0046 D3F8883C   ldr     r3, [r3, #3208]
535 004a 164A      ldr     r2, .L16
536 004c 43F01803   orr     r3, r3, #24
537 0050 C2F8883C   str     r3, [r2, #3208]

```

66:../src/Final.c **** ADC0_PORT->INTEN |= (1 << 1)|(1 << 4); //habilita los bits DONE FLAG de todas las interrupciones

```

538                .loc 2 66 0
539 0054 154B      ldr     r3, .L16+8
540 0056 DB68      ldr     r3, [r3, #12]
541 0058 144A      ldr     r2, .L16+8
542 005a 43F01203   orr     r3, r3, #18
543 005e D360      str     r3, [r2, #12]

```

68:../src/Final.c **** ADC0_PORT->CR = 0; //LO PONGO EN CERO, PARA REALIZAR LA CONFIGURACION INICIAL A GUSTO

```

544                .loc 2 68 0
545 0060 124B      ldr     r3, .L16+8
546 0062 0022      movs    r2, #0
547 0064 1A60      str     r2, [r3]

```

70:../src/Final.c **** ADC0_PORT->CR |= (1 << 1)|(1 << 4) // SEL

```

548                .loc 2 70 0

```

```

549 0066 114B      ldr    r3, .L16+8
550 0068 1A68      ldr    r2, [r3]
551 006a 1049      ldr    r1, .L16+8
552 006c 104B      ldr    r3, .L16+12
553 006e 1343      orrs   r3, r3, r2
554 0070 0B60      str    r3, [r1]

71:../src/Final.c ****          | (0xFF << 8) // CLKDIV
72:../src/Final.c ****          | (0x1 << 16) // BURST
73:../src/Final.c ****          | (0x0 << 17) // CLKS
74:../src/Final.c ****          | (0x1 << 21) // POWERMODE
75:../src/Final.c ****          | (0x0 << 24) // START
76:../src/Final.c ****          | (0x0 << 27); // EDGE

77:../src/Final.c ****  NVIC_PORT->ISER0 |= (1 << 17); //Habilitacion de la interrupcion del ADC0
en el NVIC

555                .loc 2 77 0
556 0072 4FF0E023   mov    r3, #-536813568
557 0076 D3F80031   ldr    r3, [r3, #256]
558 007a 4FF0E022   mov    r2, #-536813568
559 007e 43F40033   orr    r3, r3, #131072
560 0082 C2F80031   str    r3, [r2, #256]

78:../src/Final.c ****  NVIC_PORT->ICPR0 |= (1 << 17); //Limpieza de la interrupcion pendiente del
ADC0 en el NVIC

561                .loc 2 78 0
562 0086 4FF0E023   mov    r3, #-536813568
563 008a D3F88032   ldr    r3, [r3, #640]
564 008e 4FF0E022   mov    r2, #-536813568
565 0092 43F40033   orr    r3, r3, #131072
566 0096 C2F88032   str    r3, [r2, #640]

567                .loc 2 79 0
568 009a 00BF       nop
569 009c BD46       mov    sp, r7

```

```

570                .cfi_def_cfa_register 13
571                @ sp needed
572 009e 5DF8047B        ldr    r7, [sp], #4
573                .cfi_restore 7
574                .cfi_def_cfa_offset 0
575 00a2 7047            bx     lr
576                .L17:
577                .align    2
578                .L16:
579 00a4 00600840        .word  1074290688
580 00a8 00400F40        .word  1074741248
581 00ac 00300E40        .word  1074671616
582 00b0 12FF2100        .word  2227986
583                .cfi_endproc
584                .LFE483:
585                .section.text.CalculosHumedad,"ax",%progbits
586                .align    1
587                .global  CalculosHumedad
588                .syntax unified
589                .thumb
590                .thumb_func
591                .fpu fpv4-sp-d16
592                CalculosHumedad:
593                .LFB484:
594                81:../src/Final.c **** //Rutina de Caculo de parametros para ecuacion de humedad
595                82:../src/Final.c **** void CalculosHumedad(){
596                .loc 2 82 0
597                .cfi_startproc
598                @ args = 0, pretend = 0, frame = 0
599                @ frame_needed = 1, uses_anonymous_args = 0

```

```

600                                @ link register save eliminated.
601 0000 80B4                    push    {r7}
602                                .cfi_def_cfa_offset 4
603                                .cfi_offset 7, -4
604 0002 00AF                    add     r7, sp, #0
605                                .cfi_def_cfa_register 7
83:../src/Final.c **** Pendiente=(100)/(MaxHumedad-MinHumedad);
606                                .loc 2 83 0
607 0004 114B                    ldr     r3, .L19
608 0006 93ED007A                vldr.32 s14, [r3]
609 000a 114B                    ldr     r3, .L19+4
610 000c D3ED007A                vldr.32 s15, [r3]
611 0010 37EE677A                vsub.f32      s14, s14, s15
612 0014 DFED0F6A                vldr.32 s13, .L19+8
613 0018 C6EE877A                vdiv.f32      s15, s13, s14
614 001c 0E4B                    ldr     r3, .L19+12
615 001e C3ED007A                vstr.32 s15, [r3]
84:../src/Final.c **** Ordenada= 100 - (Pendiente*MaxHumedad);
616                                .loc 2 84 0
617 0022 0D4B                    ldr     r3, .L19+12
618 0024 93ED007A                vldr.32 s14, [r3]
619 0028 084B                    ldr     r3, .L19
620 002a D3ED007A                vldr.32 s15, [r3]
621 002e 67EE277A                vmul.f32      s15, s14, s15
622 0032 9FED087A                vldr.32 s14, .L19+8
623 0036 77EE677A                vsub.f32      s15, s14, s15
624 003a 084B                    ldr     r3, .L19+16
625 003c C3ED007A                vstr.32 s15, [r3]
85:../src/Final.c **** }
626                                .loc 2 85 0

```

```

627 0040 00BF          nop
628 0042 BD46          mov    sp, r7
629                  .cfi_def_cfa_register 13
630                  @ sp needed
631 0044 5DF8047B       ldr     r7, [sp], #4
632                  .cfi_restore 7
633                  .cfi_def_cfa_offset 0
634 0048 7047          bx      lr
635          .L20:
636 004a 00BF          .align  2
637          .L19:
638 004c 00000000       .word   MaxHumedad
639 0050 00000000       .word   MinHumedad
640 0054 0000C842       .word   1120403456
641 0058 00000000       .word   Pendiente
642 005c 00000000       .word   Ordenada
643                  .cfi_endproc
644          .LFE484:
646                  .section.text.ADC0_IRQHandler,"ax",%progbits
647                  .align  1
648                  .global  ADC0_IRQHandler
649                  .syntax unified
650                  .thumb
651                  .thumb_func
652                  .fpu fpv4-sp-d16
654          ADC0_IRQHandler:
655          .LFB485:
88:../src/Final.c **** //Rutina de interrupción del ADC
89:../src/Final.c **** void ADC0_IRQHandler(void){
656                  .loc 2 89 0

```

```

657                .cfi_startproc
658                @ args = 0, pretend = 0, frame = 0
659                @ frame_needed = 1, uses_anonymous_args = 0
660                @ link register save eliminated.
661 0000 80B4                push    {r7}
662                .cfi_def_cfa_offset 4
663                .cfi_offset 7, -4
664 0002 00AF                add     r7, sp, #0
665                .cfi_def_cfa_register 7
90:../src/Final.c **** NVIC_PORT->ICERO |= (1 << 17); //Deshabilitacion de la interrupcion del
ADC0 en el NVIC
666                .loc 2 90 0
667 0004 4FF0E023            mov     r3, #-536813568
668 0008 D3F88031            ldr     r3, [r3, #384]
669 000c 4FF0E022            mov     r2, #-536813568
670 0010 43F40033            orr     r3, r3, #131072
671 0014 C2F88031            str     r3, [r2, #384]
91:../src/Final.c **** flagADC = 1;
672                .loc 2 91 0
673 0018 034B                ldr     r3, .L22
674 001a 0122                movs    r2, #1
675 001c 1A70                strb    r2, [r3]
92:../src/Final.c **** }
676                .loc 2 92 0
677 001e 00BF                nop
678 0020 BD46                mov     sp, r7
679                .cfi_def_cfa_register 13
680                @ sp needed
681 0022 5DF8047B            ldr     r7, [sp], #4
682                .cfi_restore 7

```

```

683                .cfi_def_cfa_offset 0
684 0026 7047                bx      lr
685                .L23:
686                .align 2
687                .L22:
688 0028 00000000                .word  flagADC
689                .cfi_endproc
690                .LFE485:
692                .section.text.GPIO2_IRQHandler,"ax",%progbits
693                .align 1
694                .global GPIO2_IRQHandler
695                .syntax unified
696                .thumb
697                .thumb_func
698                .fpu fpv4-sp-d16
700                GPIO2_IRQHandler:
701                .LFB486:
93:../src/Final.c ****
94:../src/Final.c **** //Rutina de interrupción del BOTON 3
95:../src/Final.c **** void GPIO2_IRQHandler() {
702                .loc 2 95 0
703                .cfi_startproc
704                @ args = 0, pretend = 0, frame = 0
705                @ frame_needed = 1, uses_anonymous_args = 0
706 0000 80B5                push    {r7, lr}
707                .cfi_def_cfa_offset 8
708                .cfi_offset 7, -8
709                .cfi_offset 14, -4
710 0002 00AF                add     r7, sp, #0
711                .cfi_def_cfa_register 7

```



```

96:../src/Final.c **** if (ADC0_PORT->DR[4] & (1<<31)){
712                .loc 2 96 0
713 0004 0B4B                ldr    r3, .L26
714 0006 1B6A                ldr    r3, [r3, #32]
715 0008 002B                cmp    r3, #0
716 000a 08DA                bge    .L25
97:../src/Final.c **** MinHumedad=(float)dataADC_4;
717                .loc 2 97 0
718 000c 0A4B                ldr    r3, .L26+4
719 000e 1B68                ldr    r3, [r3]
720 0010 07EE903A            vmov   s15, r3 @ int
721 0014 F8EEE77A            vcvtf.f32.s32    s15, s15
722 0018 084B                ldr    r3, .L26+8
723 001a C3ED007A            vstr.32 s15, [r3]
724                .L25:
98:../src/Final.c **** }
99:../src/Final.c **** CalculosHumedad();
725                .loc 2 99 0
726 001e FFF7FEFF            bl     CalculosHumedad
100:../src/Final.c **** ToggleGPIO(1, 11); //Enciende/Apaga el led
727                .loc 2 100 0
728 0022 0B21                movs   r1, #11
729 0024 0120                movs   r0, #1
730 0026 FFF7FEFF            bl     ToggleGPIO
101:../src/Final.c **** GPIO_INT_PORT->IST = (1 << 2); //Borra el estado peticion de la IRQ2
731                .loc 2 101 0
732 002a 054B                ldr    r3, .L26+12
733 002c 0422                movs   r2, #4
734 002e 5A62                str    r2, [r3, #36]
102:../src/Final.c **** }

```

```

735                .loc 2 102 0
736 0030 00BF                nop
737 0032 80BD                pop    {r7, pc}
738                .L27:
739                .align    2
740                .L26:
741 0034 00300E40            .word  1074671616
742 0038 00000000            .word  dataADC_4
743 003c 00000000            .word  MinHumedad
744 0040 00700840            .word  1074294784
745                .cfi_endproc
746                .LFE486:
748                .section.text.GPIO3_IRQHandler,"ax",%progbits
749                .align    1
750                .global  GPIO3_IRQHandler
751                .syntax unified
752                .thumb
753                .thumb_func
754                .fpu fpv4-sp-d16
756                GPIO3_IRQHandler:
757                .LFB487:
103:../src/Final.c **** //Rutina de interrupción del BOTON 4
104:../src/Final.c **** void GPIO3_IRQHandler() {
758                .loc 2 104 0
759                .cfi_startproc
760                @ args = 0, pretend = 0, frame = 0
761                @ frame_needed = 1, uses_anonymous_args = 0
762 0000 80B5                push    {r7, lr}
763                .cfi_def_cfa_offset 8
764                .cfi_offset 7, -8

```

```

765                .cfi_offset 14, -4
766 0002 00AF                add    r7, sp, #0
767                .cfi_def_cfa_register 7
105:../src/Final.c **** if (ADC0_PORT->DR[4] & (1<<31)){
768                .loc 2 105 0
769 0004 0B4B                ldr    r3, .L30
770 0006 1B6A                ldr    r3, [r3, #32]
771 0008 002B                cmp    r3, #0
772 000a 08DA                bge    .L29
106:../src/Final.c **** MaxHumedad=(float)dataADC_4;
773                .loc 2 106 0
774 000c 0A4B                ldr    r3, .L30+4
775 000e 1B68                ldr    r3, [r3]
776 0010 07EE903A            vmov    s15, r3 @ int
777 0014 F8EEE77A            vcvtf.f32.s32    s15, s15
778 0018 084B                ldr    r3, .L30+8
779 001a C3ED007A            vstr.32 s15, [r3]
780                .L29:
107:../src/Final.c **** }
108:../src/Final.c **** CalculosHumedad();
781                .loc 2 108 0
782 001e FFF7FEFF            bl     CalculosHumedad
109:../src/Final.c **** ToggleGPIO(1, 12); //Enciende/Apaga el led
783                .loc 2 109 0
784 0022 0C21                movs   r1, #12
785 0024 0120                movs   r0, #1
786 0026 FFF7FEFF            bl     ToggleGPIO
110:../src/Final.c **** GPIO_INT_PORT->IST = (1 << 3); //Borra el estado peticion de la IRQ3
787                .loc 2 110 0
788 002a 054B                ldr    r3, .L30+12

```

```

789 002c 0822          movs   r2, #8
790 002e 5A62          str     r2, [r3, #36]
111:../src/Final.c **** }
791                  .loc 2 111 0
792 0030 00BF          nop
793 0032 80BD          pop     {r7, pc}
794          .L31:
795                  .align   2
796          .L30:
797 0034 00300E40        .word   1074671616
798 0038 00000000        .word   dataADC_4
799 003c 00000000        .word   MaxHumedad
800 0040 00700840        .word   1074294784
801                  .cfi_endproc
802          .LFE487:
804                  .section.text.transmision_datos,"ax",%progbits
805                  .align   1
806                  .global  transmision_datos
807                  .syntax unified
808                  .thumb
809                  .thumb_func
810                  .fpu fpv4-sp-d16
812          transmision_datos:
813          .LFB488:
112:../src/Final.c ****
113:../src/Final.c **** //Rutina de procesamiento de medicion y transmision de datos
114:../src/Final.c **** void transmision_datos(){
814                  .loc 2 114 0
815                  .cfi_startproc
816                  @ args = 0, pretend = 0, frame = 8

```

```

817                @ frame_needed = 1, uses_anonymous_args = 0
818 0000 80B5                push    {r7, lr}
819                .cfi_def_cfa_offset 8
820                .cfi_offset 7, -8
821                .cfi_offset 14, -4
822 0002 82B0                sub     sp, sp, #8
823                .cfi_def_cfa_offset 16
824 0004 00AF                add     r7, sp, #0
825                .cfi_def_cfa_register 7
115:../src/Final.c **** //ETAPA SENSOR HUMEDAD
116:../src/Final.c **** if (ADC0_PORT->DR[4] & (1<<31)){
826                .loc 2 116 0
827 0006 A64B                ldr     r3, .L43
828 0008 1B6A                ldr     r3, [r3, #32]
829 000a 002B                cmp     r3, #0
830 000c 80F2B880            bge     .L33
117:../src/Final.c ****      ToggleGPIO(5, 0); //Enciende/Apaga el led
831                .loc 2 117 0
832 0010 0021                movs    r1, #0
833 0012 0520                movs    r0, #5
834 0014 FFF7FEFF            bl      ToggleGPIO
118:../src/Final.c ****      dataADC_4 = (((ADC0_PORT->DR[4]) >> 6) & 0x3FF);
835                .loc 2 118 0
836 0018 A14B                ldr     r3, .L43
837 001a 1B6A                ldr     r3, [r3, #32]
838 001c 9B09                lsrs    r3, r3, #6
839 001e C3F30903            ubfx    r3, r3, #0, #10
840 0022 A04A                ldr     r2, .L43+4
841 0024 1360                str     r3, [r2]
119:../src/Final.c ****      hum0 = (float)dataADC_4;

```

```

842                .loc 2 119 0
843 0026 9F4B      ldr    r3, .L43+4
844 0028 1B68      ldr    r3, [r3]
845 002a 07EE903A  vmov   s15, r3 @ int
846 002e F8EEE77A  vcvtf.f32.s32   s15, s15
847 0032 9D4B      ldr    r3, .L43+8
848 0034 C3ED007A  vstr.32 s15, [r3]

120:../src/Final.c ****      hum0 = Pendiente*hum0 + Ordenada; //Ecuacion de pendiente de
humedad con respecto a maximo y mini

849                .loc 2 120 0
850 0038 9C4B      ldr    r3, .L43+12
851 003a 93ED007A  vldr.32 s14, [r3]
852 003e 9A4B      ldr    r3, .L43+8
853 0040 D3ED007A  vldr.32 s15, [r3]
854 0044 27EE277A  vmul.f32        s14, s14, s15
855 0048 994B      ldr    r3, .L43+16
856 004a D3ED007A  vldr.32 s15, [r3]
857 004e 77EE277A  vadd.f32        s15, s14, s15
858 0052 954B      ldr    r3, .L43+8
859 0054 C3ED007A  vstr.32 s15, [r3]

121:../src/Final.c ****      buff1 = (uint8_t)(hum0/100) + 48;

860                .loc 2 121 0
861 0058 934B      ldr    r3, .L43+8
862 005a 93ED007A  vldr.32 s14, [r3]
863 005e DFED956A  vldr.32 s13, .L43+20
864 0062 C7EE267A  vdiv.f32        s15, s14, s13
865 0066 FCEEE77A  vcvtf.u32.f32   s15, s15
866 006a C7ED017A  vstr.32 s15, [r7, #4] @ int
867 006e 3B79      ldrb   r3, [r7, #4]
868 0070 DBB2      uxtb   r3, r3

```

```

869 0072 3033          adds    r3, r3, #48
870 0074 DAB2          uxtb     r2, r3
871 0076 904B          ldr      r3, .L43+24
872 0078 1A70          strb     r2, [r3]
122:../src/Final.c ****          if(buff1 > 48){
873                  .loc 2 122 0
874 007a 8F4B          ldr      r3, .L43+24
875 007c 1B78          ldrb     r3, [r3] @ zero_extendqisi2
876 007e 302B          cmp      r3, #48
877 0080 21D9          bls      .L34
123:../src/Final.c ****          menu1[17] = buff1;
878                  .loc 2 123 0
879 0082 8D4B          ldr      r3, .L43+24
880 0084 1A78          ldrb     r2, [r3] @ zero_extendqisi2
881 0086 8D4B          ldr      r3, .L43+28
882 0088 5A74          strb     r2, [r3, #17]
124:../src/Final.c ****          Humedad[2] = hum0;
883                  .loc 2 124 0
884 008a 874B          ldr      r3, .L43+8
885 008c D3ED007A      vldr.32 s15, [r3]
886 0090 FCEEE77A      vcvtt.u32.f32    s15, s15
887 0094 C7ED017A      vstr.32 s15, [r7, #4] @ int
888 0098 3B79          ldrb     r3, [r7, #4]
889 009a DAB2          uxtb     r2, r3
890 009c 884B          ldr      r3, .L43+32
891 009e 9A70          strb     r2, [r3, #2]
125:../src/Final.c ****          hum0 = hum0-(buff0-48)*100;
892                  .loc 2 125 0
893 00a0 814B          ldr      r3, .L43+8
894 00a2 93ED007A      vldr.32 s14, [r3]

```

```

895 00a6 874B      ldr    r3, .L43+36
896 00a8 1B78      ldrb   r3, [r3] @ zero_extendqisi2
897 00aa 303B      subs   r3, r3, #48
898 00ac 6422      movs   r2, #100
899 00ae 02FB03F3   mul    r3, r2, r3
900 00b2 07EE903A   vmov   s15, r3 @ int
901 00b6 F8EEE77A   vcvt.f32.s32  s15, s15
902 00ba 77EE677A   vsub.f32      s15, s14, s15
903 00be 7A4B      ldr    r3, .L43+8
904 00c0 C3ED007A   vstr.32 s15, [r3]
905 00c4 05E0      b      .L35
906          .L34:
126:../src/Final.c ****      }else{
127:../src/Final.c ****      menu1[17] = 48;
907          .loc 2 127 0
908 00c6 7D4B      ldr    r3, .L43+28
909 00c8 3022      movs   r2, #48
910 00ca 5A74      strb   r2, [r3, #17]
128:../src/Final.c ****      Humedad[2] = 48 ;
911          .loc 2 128 0
912 00cc 7C4B      ldr    r3, .L43+32
913 00ce 3022      movs   r2, #48
914 00d0 9A70      strb   r2, [r3, #2]
915          .L35:
129:../src/Final.c ****      }
130:../src/Final.c ****      buff1 = (uint8_t)(hum0/10) + 48;
916          .loc 2 130 0
917 00d2 754B      ldr    r3, .L43+8
918 00d4 93ED007A   vldr.32 s14, [r3]
919 00d8 F2EE046A   vmov.f32      s13, #1.0e+1

```



```

920 00dc C7EE267A      vdiv.f32      s15, s14, s13
921 00e0 FCEEE77A      vcvtn.u32.f32  s15, s15
922 00e4 C7ED017A      vstr.32  s15, [r7, #4]    @ int
923 00e8 3B79          ldrb      r3, [r7, #4]
924 00ea DBB2          uxtb      r3, r3
925 00ec 3033          adds     r3, r3, #48
926 00ee DAB2          uxtb      r2, r3
927 00f0 714B          ldr       r3, .L43+24
928 00f2 1A70          strb      r2, [r3]
131:../src/Final.c ****      if(buff1 > 48){
929                      .loc 2 131 0
930 00f4 704B          ldr       r3, .L43+24
931 00f6 1B78          ldrb      r3, [r3] @ zero_extendqisi2
932 00f8 302B          cmp      r3, #48
933 00fa 1CD9          bls      .L36
132:../src/Final.c ****      menu1[18] = buff1;
934                      .loc 2 132 0
935 00fc 6E4B          ldr       r3, .L43+24
936 00fe 1A78          ldrb      r2, [r3] @ zero_extendqisi2
937 0100 6E4B          ldr       r3, .L43+28
938 0102 9A74          strb      r2, [r3, #18]
133:../src/Final.c ****      Humedad[3] = buff1;
939                      .loc 2 133 0
940 0104 6C4B          ldr       r3, .L43+24
941 0106 1A78          ldrb      r2, [r3] @ zero_extendqisi2
942 0108 6D4B          ldr       r3, .L43+32
943 010a DA70          strb      r2, [r3, #3]
134:../src/Final.c ****      hum0 = hum0-(buff1-48)*10;
944                      .loc 2 134 0
945 010c 664B          ldr       r3, .L43+8

```

```

946 010e 93ED007A      vldr.32 s14, [r3]
947 0112 694B          ldr    r3, .L43+24
948 0114 1B78          ldrb   r3, [r3] @ zero_extendqisi2
949 0116 A3F13002      sub    r2, r3, #48
950 011a 1346          mov    r3, r2
951 011c 9B00          lsls   r3, r3, #2
952 011e 1344          add    r3, r3, r2
953 0120 5B00          lsls   r3, r3, #1
954 0122 07EE903A      vmov   s15, r3 @ int
955 0126 F8EEE77A      vcvt.f32.s32  s15, s15
956 012a 77EE677A      vsub.f32      s15, s14, s15
957 012e 5E4B          ldr    r3, .L43+8
958 0130 C3ED007A      vstr.32 s15, [r3]
959 0134 05E0          b      .L37
960                  .L36:
135:../src/Final.c ****      }else{
136:../src/Final.c ****      menu1[18] = 48;
961                  .loc 2 136 0
962 0136 614B          ldr    r3, .L43+28
963 0138 3022          movs   r2, #48
964 013a 9A74          strb   r2, [r3, #18]
137:../src/Final.c ****      Humedad[3] = 48 ;
965                  .loc 2 137 0
966 013c 604B          ldr    r3, .L43+32
967 013e 3022          movs   r2, #48
968 0140 DA70          strb   r2, [r3, #3]
969                  .L37:
138:../src/Final.c ****      }
139:../src/Final.c ****      buff1 = (uint8_t)(hum0) + 48;
970                  .loc 2 139 0

```

```

971 0142 594B      ldr    r3, .L43+8
972 0144 D3ED007A  vldr.32 s15, [r3]
973 0148 FCEEE77A  vcvtt.u32.f32    s15, s15
974 014c C7ED017A  vstr.32 s15, [r7, #4]    @ int
975 0150 3B79      ldrb   r3, [r7, #4]
976 0152 DBB2      uxtb   r3, r3
977 0154 3033      adds   r3, r3, #48
978 0156 DAB2      uxtb   r2, r3
979 0158 574B      ldr    r3, .L43+24
980 015a 1A70      strb   r2, [r3]
140:../src/Final.c ****      menu1[19] = buff1;
981                .loc 2 140 0
982 015c 564B      ldr    r3, .L43+24
983 015e 1A78      ldrb   r2, [r3] @ zero_extendqisi2
984 0160 564B      ldr    r3, .L43+28
985 0162 DA74      strb   r2, [r3, #19]
141:../src/Final.c ****      Humedad[4] = buff1;
986                .loc 2 141 0
987 0164 544B      ldr    r3, .L43+24
988 0166 1A78      ldrb   r2, [r3] @ zero_extendqisi2
989 0168 554B      ldr    r3, .L43+32
990 016a 1A71      strb   r2, [r3, #4]
142:../src/Final.c ****      Chip_UART_SendBlocking(LPC_USART3, &menu1, sizeof(menu1));
991                .loc 2 142 0
992 016c 1922      movs   r2, #25
993 016e 5349      ldr    r1, .L43+28
994 0170 5548      ldr    r0, .L43+40
995 0172 FFF7FEFF  bl     Chip_UART_SendBlocking
143:../src/Final.c ****      Chip_UART_SendBlocking(LPC_USART3, &Humedad,
sizeof(Humedad));

```

```

996                                .loc 2 143 0
997 0176 0A22                      movs   r2, #10
998 0178 5149                      ldr    r1, .L43+32
999 017a 5348                      ldr    r0, .L43+40
1000 017c FFF7FEFF                 bl     Chip_UART_SendBlocking
1001                                .L33:
144:../src/Final.c **** }
145:../src/Final.c **** //ETAPA SENSOR TEMPERATURA
146:../src/Final.c **** if (ADC0_PORT->DR[1] & (1<<31)){
1002                                .loc 2 146 0
1003 0180 474B                      ldr    r3, .L43
1004 0182 5B69                      ldr    r3, [r3, #20]
1005 0184 002B                      cmp    r3, #0
1006 0186 80F2D280                 bge    .L38
147:../src/Final.c ****          ToggleGPIO(5, 1); //Enciende/Apaga el led
1007                                .loc 2 147 0
1008 018a 0121                      movs   r1, #1
1009 018c 0520                      movs   r0, #5
1010 018e FFF7FEFF                 bl     ToggleGPIO
148:../src/Final.c ****          dataADC_1 = (((ADC0_PORT->DR[1]) >> 6) & 0x3FF);
1011                                .loc 2 148 0
1012 0192 434B                      ldr    r3, .L43
1013 0194 5B69                      ldr    r3, [r3, #20]
1014 0196 9B09                      lsrs   r3, r3, #6
1015 0198 C3F30903                 ubfx   r3, r3, #0, #10
1016 019c 4B4A                      ldr    r2, .L43+44
1017 019e 1360                      str    r3, [r2]
149:../src/Final.c ****          temp0 = (float)dataADC_1;
1018                                .loc 2 149 0
1019 01a0 4A4B                      ldr    r3, .L43+44

```

```

1020 01a2 1B68      ldr    r3, [r3]
1021 01a4 07EE903A   vmov   s15, r3 @ int
1022 01a8 F8EEE77A   vcvf.f32.s32    s15, s15
1023 01ac 484B      ldr    r3, .L43+48
1024 01ae C3ED007A   vstr.32 s15, [r3]
150:../src/Final.c ****      temp0 = (temp0/1023)*330;
1025                .loc 2 150 0
1026 01b2 474B      ldr    r3, .L43+48
1027 01b4 93ED007A   vldr.32 s14, [r3]
1028 01b8 DFED466A   vldr.32 s13, .L43+52
1029 01bc C7EE267A   vdiv.f32      s15, s14, s13
1030 01c0 9FED457A   vldr.32 s14, .L43+56
1031 01c4 67EE877A   vmul.f32      s15, s15, s14
1032 01c8 414B      ldr    r3, .L43+48
1033 01ca C3ED007A   vstr.32 s15, [r3]
151:../src/Final.c ****      buff0 = (uint8_t)(temp0/100) + 48;
1034                .loc 2 151 0
1035 01ce 404B      ldr    r3, .L43+48
1036 01d0 93ED007A   vldr.32 s14, [r3]
1037 01d4 DFED376A   vldr.32 s13, .L43+20
1038 01d8 C7EE267A   vdiv.f32      s15, s14, s13
1039 01dc FCEEE77A   vcvf.u32.f32    s15, s15
1040 01e0 C7ED017A   vstr.32 s15, [r7, #4] @ int
1041 01e4 3B79      ldrb   r3, [r7, #4]
1042 01e6 DBB2      uxtb   r3, r3
1043 01e8 3033      adds   r3, r3, #48
1044 01ea DAB2      uxtb   r2, r3
1045 01ec 354B      ldr    r3, .L43+36
1046 01ee 1A70      strb   r2, [r3]
152:../src/Final.c ****      if(buff0 > 48){

```

```

1047                .loc 2 152 0
1048 01f0 344B            ldr    r3, .L43+36
1049 01f2 1B78            ldrb   r3, [r3] @ zero_extendqisi2
1050 01f4 302B            cmp    r3, #48
1051 01f6 1AD9            bls    .L39
153:../src/Final.c ****          menu0[19] = buff0;
1052                .loc 2 153 0
1053 01f8 324B            ldr    r3, .L43+36
1054 01fa 1A78            ldrb   r2, [r3] @ zero_extendqisi2
1055 01fc 374B            ldr    r3, .L43+60
1056 01fe DA74            strb   r2, [r3, #19]
154:../src/Final.c ****          Temperatura[2] = buff0;
1057                .loc 2 154 0
1058 0200 304B            ldr    r3, .L43+36
1059 0202 1A78            ldrb   r2, [r3] @ zero_extendqisi2
1060 0204 364B            ldr    r3, .L43+64
1061 0206 9A70            strb   r2, [r3, #2]
155:../src/Final.c ****          temp0 = temp0-(buff0-48)*100;
1062                .loc 2 155 0
1063 0208 314B            ldr    r3, .L43+48
1064 020a 93ED007A        vldr.32 s14, [r3]
1065 020e 2D4B            ldr    r3, .L43+36
1066 0210 1B78            ldrb   r3, [r3] @ zero_extendqisi2
1067 0212 303B            subs   r3, r3, #48
1068 0214 6422            movs   r2, #100
1069 0216 02FB03F3        mul    r3, r2, r3
1070 021a 07EE903A        vmov   s15, r3 @ int
1071 021e F8EEE77A        vcvt.f32.s32 s15, s15
1072 0222 77EE677A        vsub.f32 s15, s14, s15
1073 0226 2A4B            ldr    r3, .L43+48

```

```

1074 0228 C3ED007A      vstr.32 s15, [r3]
1075 022c 05E0          b      .L40
1076      .L39:
156:../src/Final.c ****      }else{
157:../src/Final.c ****      menu0[19] = 48;
1077      .loc 2 157 0
1078 022e 2B4B          ldr      r3, .L43+60
1079 0230 3022          movs     r2, #48
1080 0232 DA74          strb     r2, [r3, #19]
158:../src/Final.c ****      Temperatura[2] = 48;
1081      .loc 2 158 0
1082 0234 2A4B          ldr      r3, .L43+64
1083 0236 3022          movs     r2, #48
1084 0238 9A70          strb     r2, [r3, #2]
1085      .L40:
159:../src/Final.c ****      }
160:../src/Final.c ****      buff0 = (uint8_t)(temp0/10) + 48;
1086      .loc 2 160 0
1087 023a 254B          ldr      r3, .L43+48
1088 023c 93ED007A      vldr.32 s14, [r3]
1089 0240 F2EE046A      vmov.f32      s13, #1.0e+1
1090 0244 C7EE267A      vdiv.f32      s15, s14, s13
1091 0248 FCEEE77A      vcvf.u32.f32  s15, s15
1092 024c C7ED017A      vstr.32 s15, [r7, #4]    @ int
1093 0250 3B79          ldrb     r3, [r7, #4]
1094 0252 DBB2          uxtb     r3, r3
1095 0254 3033          adds     r3, r3, #48
1096 0256 DAB2          uxtb     r2, r3
1097 0258 1A4B          ldr      r3, .L43+36
1098 025a 1A70          strb     r2, [r3]

```

```

161:../src/Final.c ****          if(buff0 > 48){
1099          .loc 2 161 0
1100 025c 194B          ldr    r3, .L43+36
1101 025e 1B78          ldrb   r3, [r3] @ zero_extendqisi2
1102 0260 302B          cmp    r3, #48
1103 0262 3FD9          bls    .L41
162:../src/Final.c ****          menu0[20] = buff0;
1104          .loc 2 162 0
1105 0264 174B          ldr    r3, .L43+36
1106 0266 1A78          ldrb   r2, [r3] @ zero_extendqisi2
1107 0268 1C4B          ldr    r3, .L43+60
1108 026a 1A75          strb   r2, [r3, #20]
163:../src/Final.c ****          Temperatura[3] = buff0;
1109          .loc 2 163 0
1110 026c 154B          ldr    r3, .L43+36
1111 026e 1A78          ldrb   r2, [r3] @ zero_extendqisi2
1112 0270 1B4B          ldr    r3, .L43+64
1113 0272 DA70          strb   r2, [r3, #3]
164:../src/Final.c ****          temp0 = temp0-(buff0-48)*10;
1114          .loc 2 164 0
1115 0274 164B          ldr    r3, .L43+48
1116 0276 93ED007A      vldr.32 s14, [r3]
1117 027a 124B          ldr    r3, .L43+36
1118 027c 1B78          ldrb   r3, [r3] @ zero_extendqisi2
1119 027e A3F13002      sub    r2, r3, #48
1120 0282 1346          mov    r3, r2
1121 0284 9B00          lsls   r3, r3, #2
1122 0286 1344          add    r3, r3, r2
1123 0288 5B00          lsls   r3, r3, #1
1124 028a 07EE903A      vmov   s15, r3 @ int

```


1125 028e F8EEE77A	vcvt.f32.s32	s15, s15
1126 0292 77EE677A	vsub.f32	s15, s14, s15
1127 0296 0E4B	ldr	r3, .L43+48
1128 0298 C3ED007A	vstr.32	s15, [r3]
1129 029c 28E0	b	.L42
1130	.L44:	
1131 029e 00BF	.align	2
1132	.L43:	
1133 02a0 00300E40	.word	1074671616
1134 02a4 00000000	.word	dataADC_4
1135 02a8 00000000	.word	hum0
1136 02ac 00000000	.word	Pendiente
1137 02b0 00000000	.word	Ordenada
1138 02b4 0000C842	.word	1120403456
1139 02b8 00000000	.word	buff1
1140 02bc 00000000	.word	menu1
1141 02c0 00000000	.word	Humedad
1142 02c4 00000000	.word	buff0
1143 02c8 00200C40	.word	1074536448
1144 02cc 00000000	.word	dataADC_1
1145 02d0 00000000	.word	temp0
1146 02d4 00C07F44	.word	1149222912
1147 02d8 0000A543	.word	1134886912
1148 02dc 00000000	.word	menu0
1149 02e0 00000000	.word	Temperatura
1150	.L41:	
165:../src/Final.c ****		}else{
166:../src/Final.c ****		menu0[20] = 48;
1151	.loc 2 166 0	
1152 02e4 1A4B	ldr	r3, .L45

```

1153 02e6 3022      movs    r2, #48
1154 02e8 1A75      strb    r2, [r3, #20]
167:../src/Final.c ****      Temperatura[3] = 48;
1155              .loc 2 167 0
1156 02ea 1A4B      ldr     r3, .L45+4
1157 02ec 3022      movs    r2, #48
1158 02ee DA70      strb    r2, [r3, #3]
1159      .L42:
168:../src/Final.c ****      }
169:../src/Final.c ****      buff0 = (uint8_t)(temp0) + 48;
1160              .loc 2 169 0
1161 02f0 194B      ldr     r3, .L45+8
1162 02f2 D3ED007A   vldr.32 s15, [r3]
1163 02f6 FCEEE77A   vcvt.u32.f32    s15, s15
1164 02fa C7ED017A   vstr.32 s15, [r7, #4]    @ int
1165 02fe 3B79      ldrb    r3, [r7, #4]
1166 0300 DBB2      uxtb    r3, r3
1167 0302 3033      adds    r3, r3, #48
1168 0304 DAB2      uxtb    r2, r3
1169 0306 154B      ldr     r3, .L45+12
1170 0308 1A70      strb    r2, [r3]
170:../src/Final.c ****      menu0[21] = buff0;
1171              .loc 2 170 0
1172 030a 144B      ldr     r3, .L45+12
1173 030c 1A78      ldrb    r2, [r3] @ zero_extendqisi2
1174 030e 104B      ldr     r3, .L45
1175 0310 5A75      strb    r2, [r3, #21]
171:../src/Final.c ****      Temperatura[4] = buff0;
1176              .loc 2 171 0
1177 0312 124B      ldr     r3, .L45+12

```

```

1178 0314 1A78          ldrb    r2, [r3] @ zero_extendqisi2
1179 0316 0F4B          ldr     r3, .L45+4
1180 0318 1A71          strb    r2, [r3, #4]
172:../src/Final.c **** Chip_UART_SendBlocking(LPC_USART3, &menu0, sizeof(menu0));
1181                  .loc 2 172 0
1182 031a 1C22          movs    r2, #28
1183 031c 0C49          ldr     r1, .L45
1184 031e 1048          ldr     r0, .L45+16
1185 0320 FFF7FEFF       bl      Chip_UART_SendBlocking
173:../src/Final.c **** Chip_UART_SendBlocking(LPC_USART3, &Temperatura,
sizeof(Temperatura));
1186                  .loc 2 173 0
1187 0324 0A22          movs    r2, #10
1188 0326 0B49          ldr     r1, .L45+4
1189 0328 0D48          ldr     r0, .L45+16
1190 032a FFF7FEFF       bl      Chip_UART_SendBlocking
1191                  .L38:
174:../src/Final.c **** }
175:../src/Final.c **** flagADC = 0;
1192                  .loc 2 175 0
1193 032e 0D4B          ldr     r3, .L45+20
1194 0330 0022          movs    r2, #0
1195 0332 1A70          strb    r2, [r3]
176:../src/Final.c **** NVIC_PORT->ISER0 |= (1 << 17); //Habilitacion de la interrupcion del ADC0
en el NVIC
1196                  .loc 2 176 0
1197 0334 4FF0E023       mov     r3, #-536813568
1198 0338 D3F80031       ldr     r3, [r3, #256]
1199 033c 4FF0E022       mov     r2, #-536813568
1200 0340 43F40033       orr     r3, r3, #131072
1201 0344 C2F80031       str     r3, [r2, #256]

```

```

177:../src/Final.c **** }
1202                .loc 2 177 0
1203 0348 00BF                nop
1204 034a 0837                adds    r7, r7, #8
1205                .cfi_def_cfa_offset 8
1206 034c BD46                mov     sp, r7
1207                .cfi_def_cfa_register 13
1208                @ sp needed
1209 034e 80BD                pop     {r7, pc}
1210                .L46:
1211                .align    2
1212                .L45:
1213 0350 00000000            .word    menu0
1214 0354 00000000            .word    Temperatura
1215 0358 00000000            .word    temp0
1216 035c 00000000            .word    buff0
1217 0360 00200C40            .word    1074536448
1218 0364 00000000            .word    flagADC
1219                .cfi_endproc
1220                .LFE488:
1222                .section.text.main,"ax",%progbits
1223                .align    1
1224                .global    main
1225                .syntax    unified
1226                .thumb
1227                .thumb_func
1228                .fpu    fpv4-sp-d16
1230                main:
1231                .LFB489:
178:../src/Final.c ****

```

```

179:../src/Final.c ****
180:../src/Final.c **** int main(void){
1232                .loc 2 180 0
1233                .cfi_startproc
1234                @ args = 0, pretend = 0, frame = 0
1235                @ frame_needed = 1, uses_anonymous_args = 0
1236 0000 80B5                push    {r7, lr}
1237                .cfi_def_cfa_offset 8
1238                .cfi_offset 7, -8
1239                .cfi_offset 14, -4
1240 0002 82B0                sub     sp, sp, #8
1241                .cfi_def_cfa_offset 16
1242 0004 02AF                add     r7, sp, #8
1243                .cfi_def_cfa 7, 8
181:../src/Final.c ****
182:../src/Final.c **** ConfLED(2, 0, 5, 0, 4); //Configuramos el SCU P2_0, GPIO5 puerto 5 pin 0 con
la funcion 4
1244                .loc 2 182 0
1245 0006 0423                movs   r3, #4
1246 0008 0093                str    r3, [sp]
1247 000a 0023                movs   r3, #0
1248 000c 0522                movs   r2, #5
1249 000e 0021                movs   r1, #0
1250 0010 0220                movs   r0, #2
1251 0012 FFF7FEFF          bl     ConfLED
183:../src/Final.c **** ConfLED(2, 1, 5, 1, 4); //Configuramos el SCU P2_1, GPIO5 puerto 5 pin 1 con
la funcion 4
1252                .loc 2 183 0
1253 0016 0423                movs   r3, #4
1254 0018 0093                str    r3, [sp]
1255 001a 0123                movs   r3, #1

```

```

1256 001c 0522          movs   r2, #5
1257 001e 0121          movs   r1, #1
1258 0020 0220          movs   r0, #2
1259 0022 FFF7FEFF      bl      ConfLED

184:../src/Final.c **** ConfLED(2, 2, 5, 2, 4); //Configuramos el SCU P2_2, GPIO5 puerto 5 pin 2 con
la funcion 4

1260                  .loc 2 184 0
1261 0026 0423          movs   r3, #4
1262 0028 0093          str     r3, [sp]
1263 002a 0223          movs   r3, #2
1264 002c 0522          movs   r2, #5
1265 002e 0221          movs   r1, #2
1266 0030 0220          movs   r0, #2
1267 0032 FFF7FEFF      bl      ConfLED

185:../src/Final.c **** ConfLED(2, 10, 0, 14, 0); //Configuramos el SCU P2_10, GPIO0 puerto 0 pin
14 con la funcion 0

1268                  .loc 2 185 0
1269 0036 0023          movs   r3, #0
1270 0038 0093          str     r3, [sp]
1271 003a 0E23          movs   r3, #14
1272 003c 0022          movs   r2, #0
1273 003e 0A21          movs   r1, #10
1274 0040 0220          movs   r0, #2
1275 0042 FFF7FEFF      bl      ConfLED

186:../src/Final.c **** ConfLED(2, 11, 1, 11, 0); //Configuramos el SCU P2_11, GPIO1 puerto 1 pin
11 con la funcion 0

1276                  .loc 2 186 0
1277 0046 0023          movs   r3, #0
1278 0048 0093          str     r3, [sp]
1279 004a 0B23          movs   r3, #11
1280 004c 0122          movs   r2, #1

```

```

1281 004e 0B21          movs   r1, #11
1282 0050 0220          movs   r0, #2
1283 0052 FFF7FEFF      bl      ConfLED

187:../src/Final.c **** ConfLED(2, 12, 1, 12, 0); //Configuramos el SCU P2_12, GPIO1 puerto 1 pin
12 con la funcion 0

1284                  .loc 2 187 0
1285 0056 0023          movs   r3, #0
1286 0058 0093          str     r3, [sp]
1287 005a 0C23          movs   r3, #12
1288 005c 0122          movs   r2, #1
1289 005e 0C21          movs   r1, #12
1290 0060 0220          movs   r0, #2
1291 0062 FFF7FEFF      bl      ConfLED

188:../src/Final.c ****

189:../src/Final.c **** ConfButton(1, 2, 0, 9, 0, 2); //Direccion 1_2; Puerto 0; Pin 9; Funcion 0
//Puls2 Pinout //interru

1292                  .loc 2 189 0
1293 0066 0223          movs   r3, #2
1294 0068 0193          str     r3, [sp, #4]
1295 006a 0023          movs   r3, #0
1296 006c 0093          str     r3, [sp]
1297 006e 0923          movs   r3, #9
1298 0070 0022          movs   r2, #0
1299 0072 0221          movs   r1, #2
1300 0074 0120          movs   r0, #1
1301 0076 FFF7FEFF      bl      ConfButton

190:../src/Final.c **** ConfButton(1, 6, 1, 9, 0, 3); //Direccion 1_6; Puerto 1; Pin 9; Funcion 0
//Puls3 Pinout //interru

1302                  .loc 2 190 0
1303 007a 0323          movs   r3, #3
1304 007c 0193          str     r3, [sp, #4]

```

```

1305 007e 0023      movs   r3, #0
1306 0080 0093      str     r3, [sp]
1307 0082 0923      movs   r3, #9
1308 0084 0122      movs   r2, #1
1309 0086 0621      movs   r1, #6
1310 0088 0120      movs   r0, #1
1311 008a FFF7FEFF    bl      ConfButton
191:../src/Final.c ****
192:../src/Final.c **** CalculosHumedad();
1312              .loc 2 192 0
1313 008e FFF7FEFF    bl      CalculosHumedad
193:../src/Final.c **** UART_Config();
1314              .loc 2 193 0
1315 0092 FFF7FEFF    bl      UART_Config
194:../src/Final.c **** confADC();
1316              .loc 2 194 0
1317 0096 FFF7FEFF    bl      confADC
1318              .L49:
195:../src/Final.c ****
196:../src/Final.c **** while (1) {
197:../src/Final.c ****     if(flagADC){
1319              .loc 2 197 0
1320 009a 034B      ldr     r3, .L50
1321 009c 1B78      ldrb    r3, [r3] @ zero_extendqisi2
1322 009e 002B      cmp     r3, #0
1323 00a0 FB00      beq     .L49
198:../src/Final.c ****     transmision_datos();
1324              .loc 2 198 0
1325 00a2 FFF7FEFF    bl      transmision_datos
197:../src/Final.c ****     transmision_datos();

```



```
1326          .loc 2 197 0
1327 00a6 F8E7          b      .L49
1328          .L51:
1329          .align  2
1330          .L50:
1331 00a8 00000000      .word  flagADC
1332          .cfi_endproc
```