# Estación meteorológica con conexión a internet

Complementos Sistemas Electrónicos Digitales Avanzados – MUIT

Adrián Gallego Sánchez

Boby Nicusor Constantin

# Índice

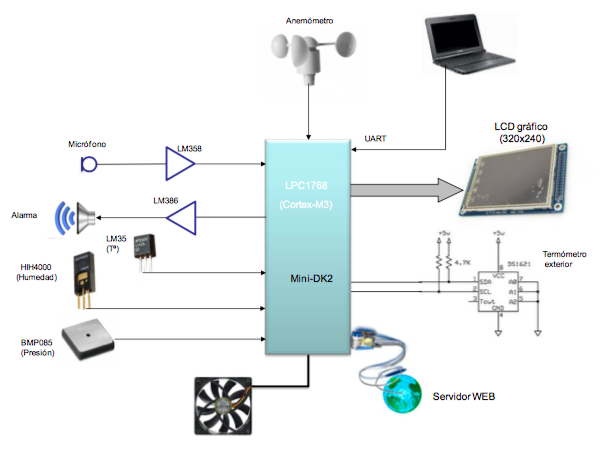
# Introducción

La propuesta de proyecto consiste en diseñar un sistema empotrado basado en el

microcontrolador LPC1768 (Cortex-M3). El objetivo es implementar una estación meteorológica que ofrezca la posibilidad de ser monitorizada de forma remota. El sistema contará con los siguientes elementos:

* Sensor analógico de temperatura LM35
* Termómetro-termostato digital DS1621
* Sensor analógico de humedad HIH4000
* Sensor digital de presión BMP180
* Micrófono
* Anemómetro
* Ventilador
* Altavoz
* Módulo de visualización (HY28B)

El sistema mostrará sobre un display los datos obtenidos de los diferentes sensores. Además, también podrán ser monitorizados mediante un entorno WEB o una interfaz serie asíncrona.



# Descripción del proyecto

A continuación, se procede a describir los elementos hardware que componen el proyecto.

### Sensor LM35

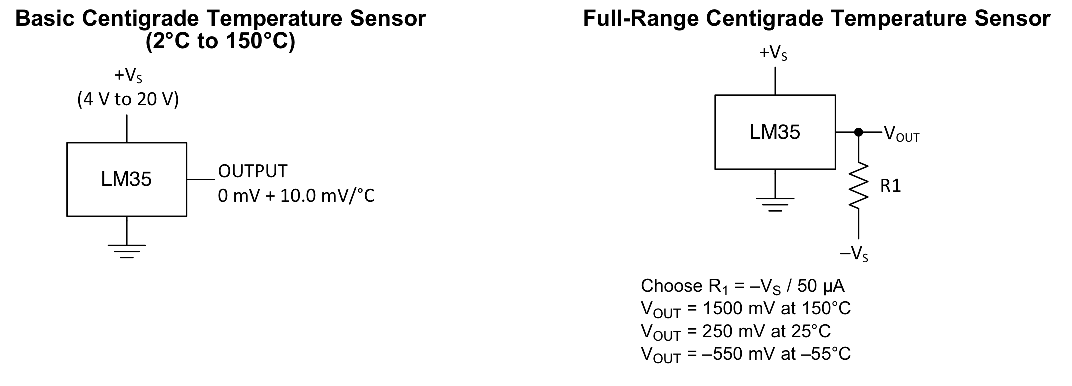
El LM35 es un dispositivo de circuito integrado que mide temperatura con una salida en tensión proporcional a la temperatura en grados centígrados. La ventaja de los dispositivos LM35 sobre los sensores de temperatura calibrados en Kelvin es que el usuario no necesita restar a la salida una constante en tensión pera obtener un escalado apropiado de los grados centígrados. Los dispositivos LM35 no necesitan un calibrado externo para ofrecer una precisión de ¼ de grado a temperatura ambiente y hasta ¾ de grado cuando mide temperaturas en el rango de -55ºC a 150ºC. La salida de baja impedancia, la salida lineal y la calibración inherente hacen que la lectura y circuitería de control resulten especialmente sencillas.

Características principales:

* Calibrado directamente en grados centígrados (Celsius)
* Factor lineal de +10 mV/ºC
* Precisión de 0.5ºC a temperatura ambiente (25ºC)
* Funciona con una alimentación desde 4 a 30 V



Modos de funcionamiento:

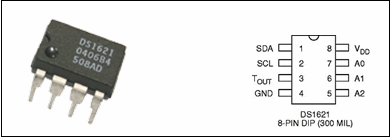


### Sensor DS1621

El temómetro-termostato digital proporciona lecturas de temperatura de 9 bit. La salida de alarma Tout se activa cuando la temperatura del dispositivo supera el umbral definido por el usuario (TH). La salida permanece activa hasta que la temperatura se sitúa por debajo de un umbral (TL), también definido por el usuario.

Los ajustes definidos por el usuario se almacenan en memoria no volátil por lo que puede ser programado antes de ser introducido en un sistema. Los ajustes y lecturas de temperatura se comunican hacia/desde el DS1621 mediante un interfaz serie de dos hilos (I2C).

Características principales:

* El DS1621 es capaz de medir temperaturas en el rango de -55ºC a 125ºC con incrementos de 0.5ºC
* La configuración de termostato es definida por el usuario y guardada en memoria no volátil
* Rango de alimentación de 2.7 a 5.5 V
* La comunicación se establece mediante I2C
* No requiere de componentes externos para medir la temperatura

Comandos:

* Leer temperatura [**AAh**]: lee el resultado de la última conversión de temperatura. EL DS1621 envía 2 bytes
* Acceso a TH [**A1h**]: si el valor campo R/W es ‘0’ se escribe en el registro TH (Alta Temperatura). Los dos siguientes bytes que se escriban/envíen al DS1621 configurarán el valor del umbral superior de temperatura para la alarma. Si el valor es ‘1’ entonces se lee el valor del registro TH
* Acceso a TL [**A2h**]: si el valor campo R/W es ‘0’ se escribe en el registro TL (Baja Temperatura). Los dos siguientes bytes que se escriban/envíen al DS1621 configurarán el valor del umbral inferior de temperatura para la alarma. Si el valor es ‘1’ entonces se lee el valor del registro TL
* Acceso a Config [**ACh**]: si el valor campo R/W es ‘0’ se escribe en el registro de configuración. El siguiente byte se escribirá en el registro. Si el valor es ‘1’ se lee el valor del registro de configuración
* Leer Contador [A8h]: este comando lee el valor *Cout\_Remain*. El comando es válido únicamente si R/W es ‘1’
* Leer *Slope* (pendiente)[**A9h**]:este comando lee el valor *Count\_Per\_C.* El comando es válido únicamente si R/W es ‘1’
* Iniciar conversión T [**EEh**]: inicia la conversión de temperatura. En el modo disparo una vez que se realice la conversión el DS1621 quedará ocioso. En el modo continuo, el DS1621 el comando iniciará conversiones continuamente
* Parar la conversión T [**22h**]: interrumpe la conversión de temperatura. Se utiliza para detener un DS1621 en modo conversión continua. Tras enviar este comando, el sensor finalizará la conversión en curso y me mantendrá ocioso hasta nueva orden

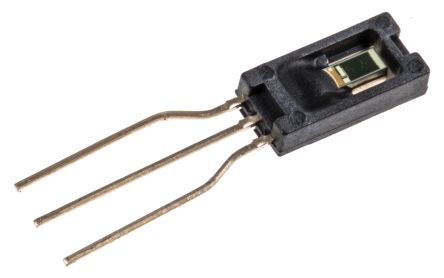
### Sensor HIH4000

La serie HIH-4000 de sensores de humedad están específicamente diseñados para un alto volumen de usuarios OEM (Original Equipment Manufacturer).

La salida casi lineal en tensión permite la conexión directa con un controlador u otro dispositivo. Con un consumo de corriente de 200 μA, típicamente, la serie HIH-4000 es ideal para sistemas de bajo consumo.

Características principales:

* Encapsulado de plástico termoendurecido
* Salida en tensión casi lineal frente a % de humedad relativa
* Diseño de bajo consumo
* Tiempo rápido de respuesta
* Precisión mejorada
* Calibración de fábrica



### Sensor BMP180

Se trata de un sensor de presión diseñado para ser conectado directamente a un microcontrolador a través de un bus I2C. Su electrónica de ultra-bajo consumo lo hacen óptimo para teléfonos móviles, PDAs, navegadores GPS y equipamiento de campo.

El BMP180 está formado por un sensor piezo-resistivo, un conversor analógico digital y una unidad de control con E2PROM e interfaz serie I2C. El dispositivo entrega el valor de presión y temperatura sin calibrar. La memoria E2PROM contiene 176 bits de calibración que se utilizan para calibrar la temperatura y la presión entre otros parámetros.

La memoria E2PROM está dividida en 11 palabras de 16 bits cada una. Cada palabra conforma un parámetro o coeficiente de calibración. Estos coeficientes son individuales para cada sensor, por lo tanto, el *master* debe leer los parámetros antes de la la primera conversión.

Características principales:

* Rango de alimentación de 1.8 a 3.6 V
* Rango de medida de presión desde 300 hasta 1100 hPa
* Bajo consumo: 5 μA por muestra/seg en modo estándar
* Precisión de 0.03 hPa en modo de alta precisión

### 

### Micrófono

### Altavoz

### Ventilador

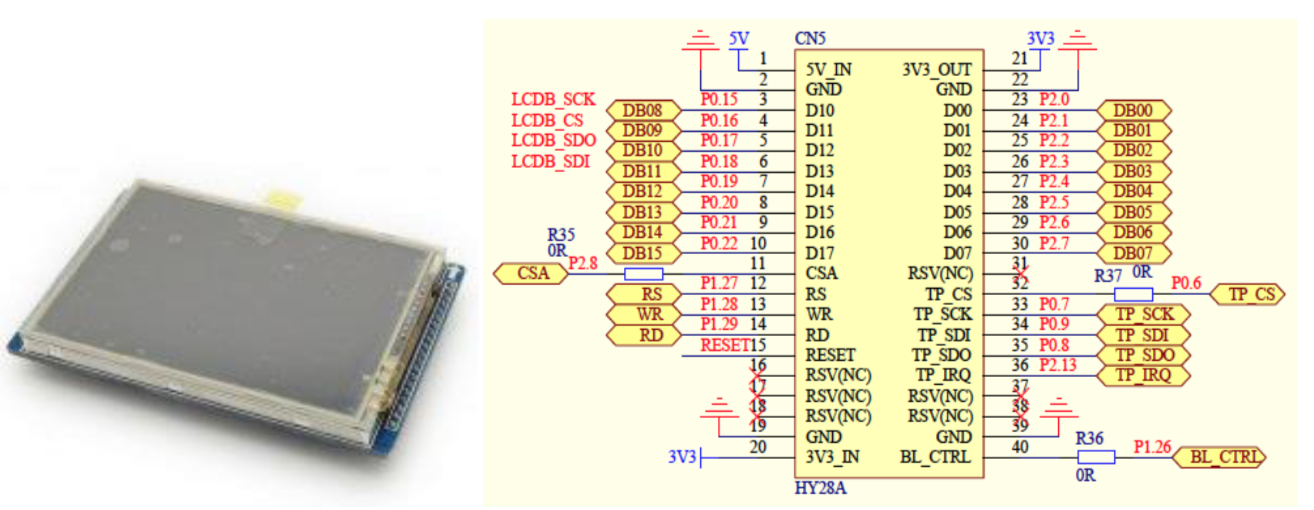
### Anemómetro

### Módulo de visualización (HY28B)

El módulo de visualización consta de un display TFT de 2.8” a color táctil y con una resolución de 320x240 píxeles. Utiliza el driver ILI9325, un driver integrado que proporciona hasta 262.144 colores. Además, cuenta con una RAM de 172.800 bytes de memoria gráfica. El display está implementado en un PCB que incorpora los circuitos de la interfaz táctil XPT2046 y de la retroiluminación del panel.

El driver ILI9325 permite 3 modos de funcionamiento de alta velocidad: 8 bits en paralelo, 16 bits en paralelo o mediante SPI.

El controlador XPT2046 es un controlador de pantalla táctil resistiva que incorpora un conversor A/D de 12 bit y 125 KHz de muestreo. Es capaz de detectar la presión en la pantalla mediante dos conversiones A/D. Además de la posición, también mide la presión de la pulsación.



# Software utilizado

A continuación, se describen las bibliotecas más destacables que se han empleado para desarrollar el proyecto así como la función que desempeñan dentro de este.

### HTTP

Esta biblioteca es proporcionada por KEIL y nos permite generar páginas web dinámicas que es uno de los objetivos del proyecto. Los ficheros más destacables que incluye son:

* TCP\_CM3.lib: proporciona las funciones de comunicaciones de KEIL RL-TCPnet.
* EMAC\_LPC17xx\_LAN8720.c: implementa la interfaz entre las funciones de la biblioteca de comunicaciones con el hardware, tanto del subsistema hardware del microcontrolador como con el chip externo que implementa el nivel físico.
* Net\_Config.c: fichero a través del cual se configura la biblioteca de funciones de la comunicación TCP/IP. Entre otros, se puede configurar la dirección IP, la máscara de red, el servidor DNS, el Gateway o la dirección MAC.
* Net\_Debug.c: permite depurar las comunicaciones TCP/IP.
* TCPD\_CM3.lib: biblioteca de funciones de comunicaciones de Keil RL-TCPnet que permiten depuración. Se debe mencionar que no es compatible con la librería TCP\_CM3.lib

### GLCD

Esta biblioteca se encarga de gestionar el display y su configuración. Las librerías principales son:

* GLCD.c: contiene las funciones de configuración y acceso al display.
* AsciiLib.c: contiene la tabla de codificaciones de los píxeles gráficos que corresponde a cada carácter ASCII.

### I2c\_lpcxx

Esta biblioteca se encarga de implementar las funciones necesarias para simular el funcionamiento de i2c mediante los pines P0.0 y P0.1 que simulan SDA y SCL respectivamente. Las principales funciones son:

* I2CSendAddr(): genera la condición de START, y establece la comunicación con el dispositivo.
* I2CGetByte(): se encarga de leer la información procedente del dispositivo o esclavo.
* I2CSendByte(): se encarga de enviar información al dispositivo.
* I2CSendStop() se encarga de dejar el bus en reposo.
* I2CDelay(): se encarga de garantizar los tiempos de retardo mínimos.
* Pulso\_SCL(): se encarga de generar de simular los pulsos del reloj.

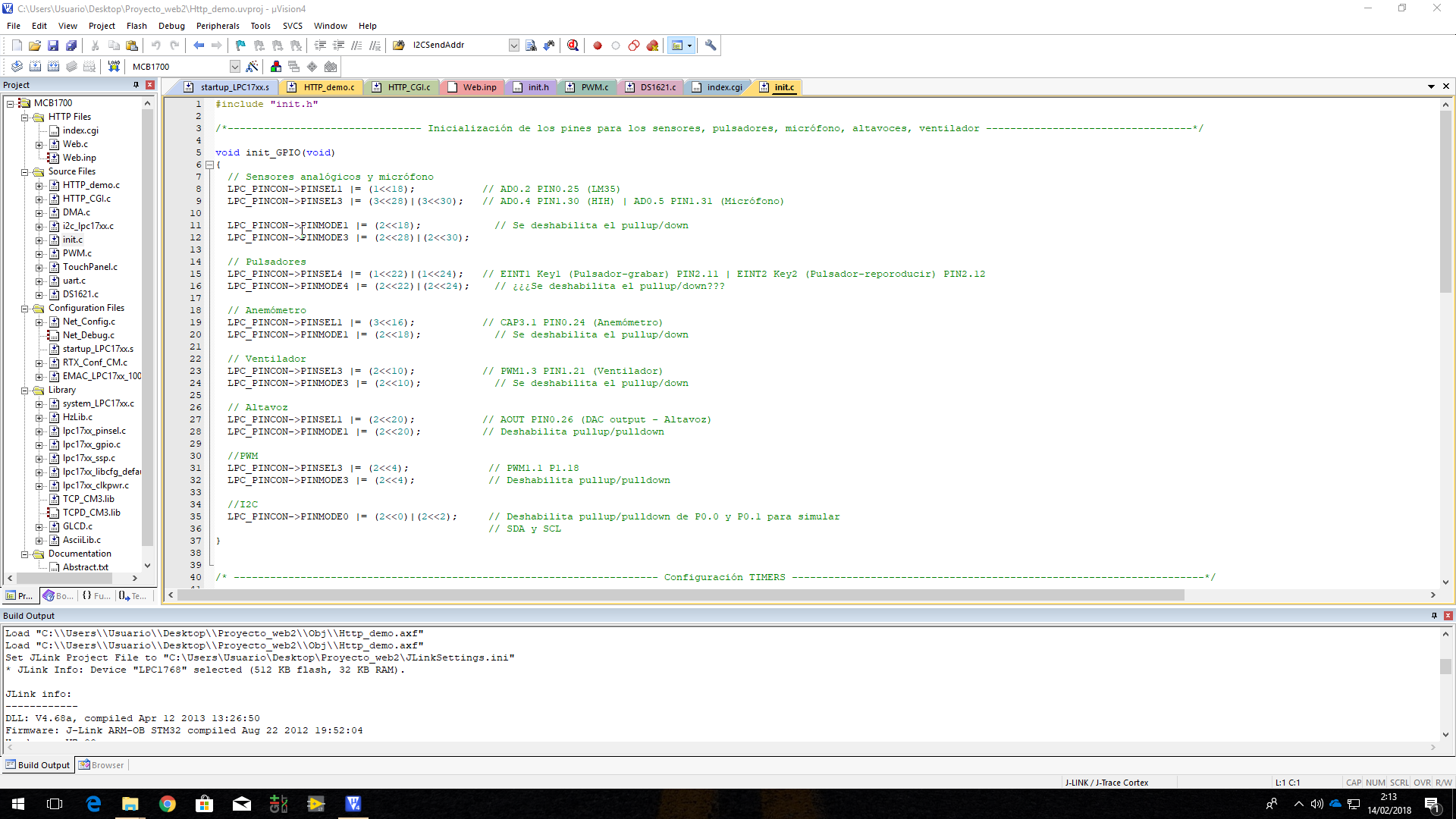
# Código generado

El código generado se ha dividido en varios archivos. A continuación, se describe cada uno de ellos.

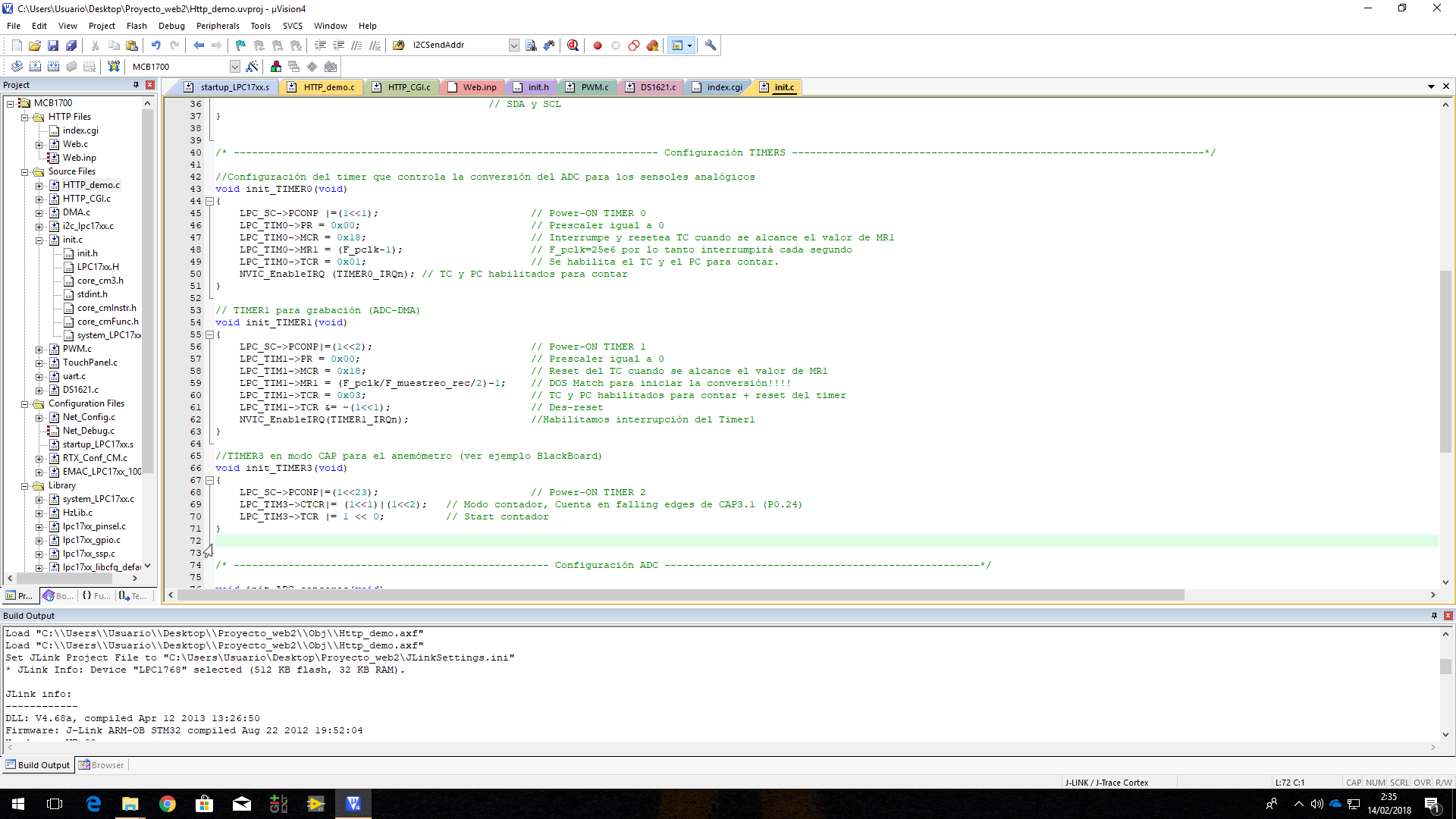
### init.c

Este archivo es uno de los más importantes del proyecto ya que se encarga de realizar la configuración de los elementos de la placa MINI-DK2. Las funciones que se implementan son:

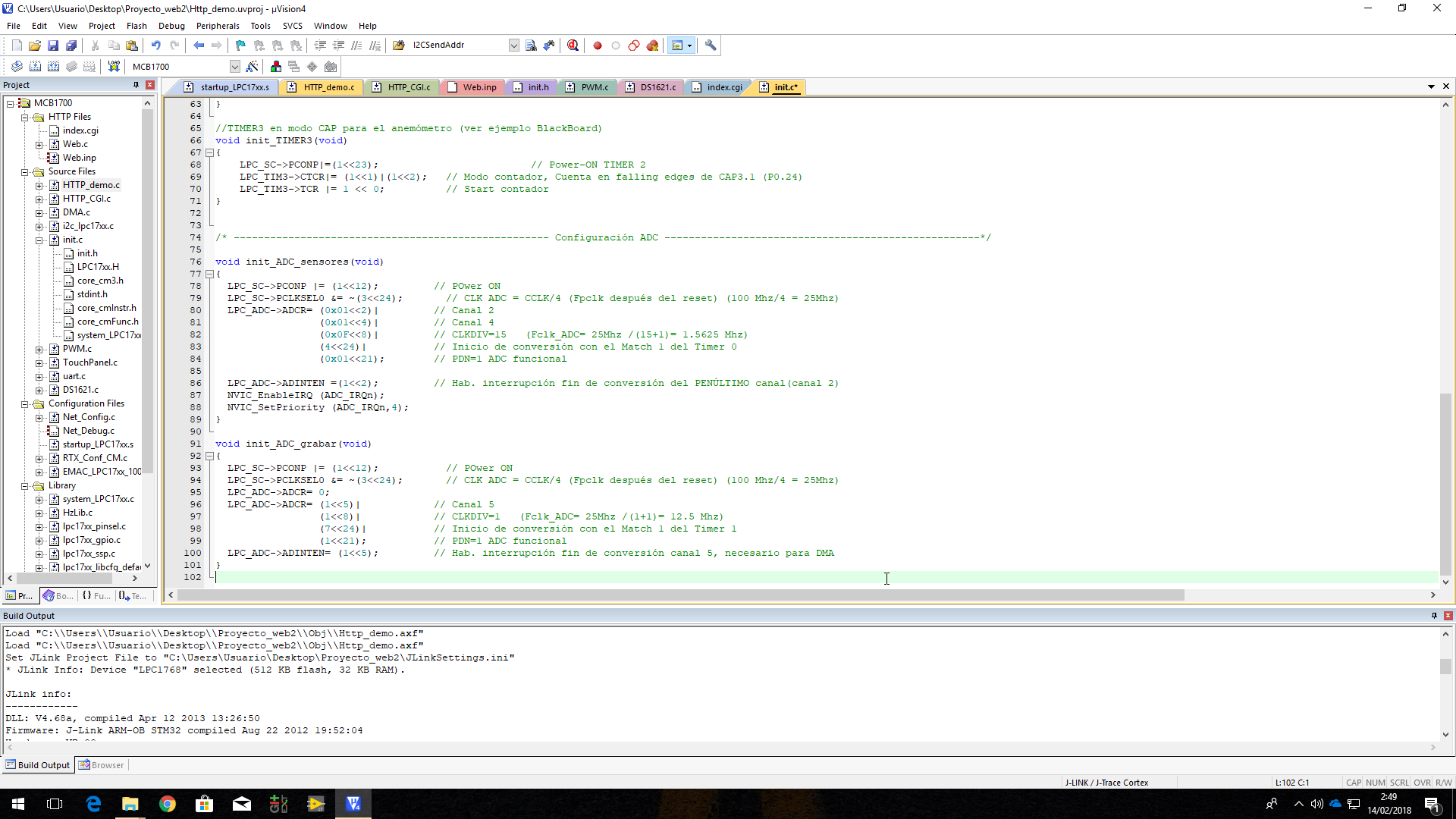
* Init\_GPIO(): se encarga de inicializar los pines para los sensores, micrófono, pulsadores, anemómetro, ventilador, altavoz, PWM y para los GPIO que se utilizan como SDA y SCL para simular I2C



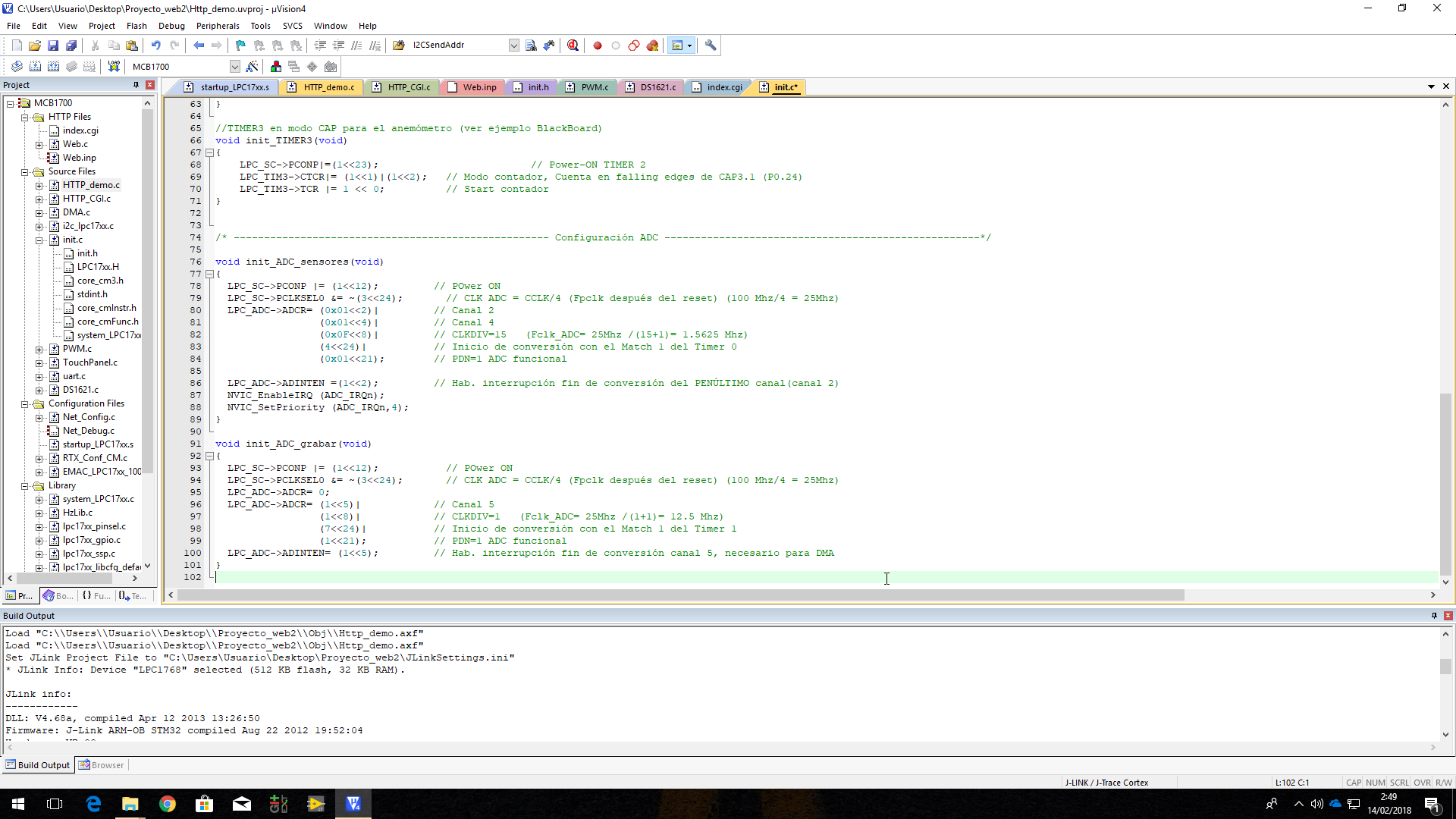
* Init\_TIMER0: configura el TIMER0 para que interrumpa cada segundo. Su función de interrupción se utilizará para gestionar diferentes tareas del sistema.



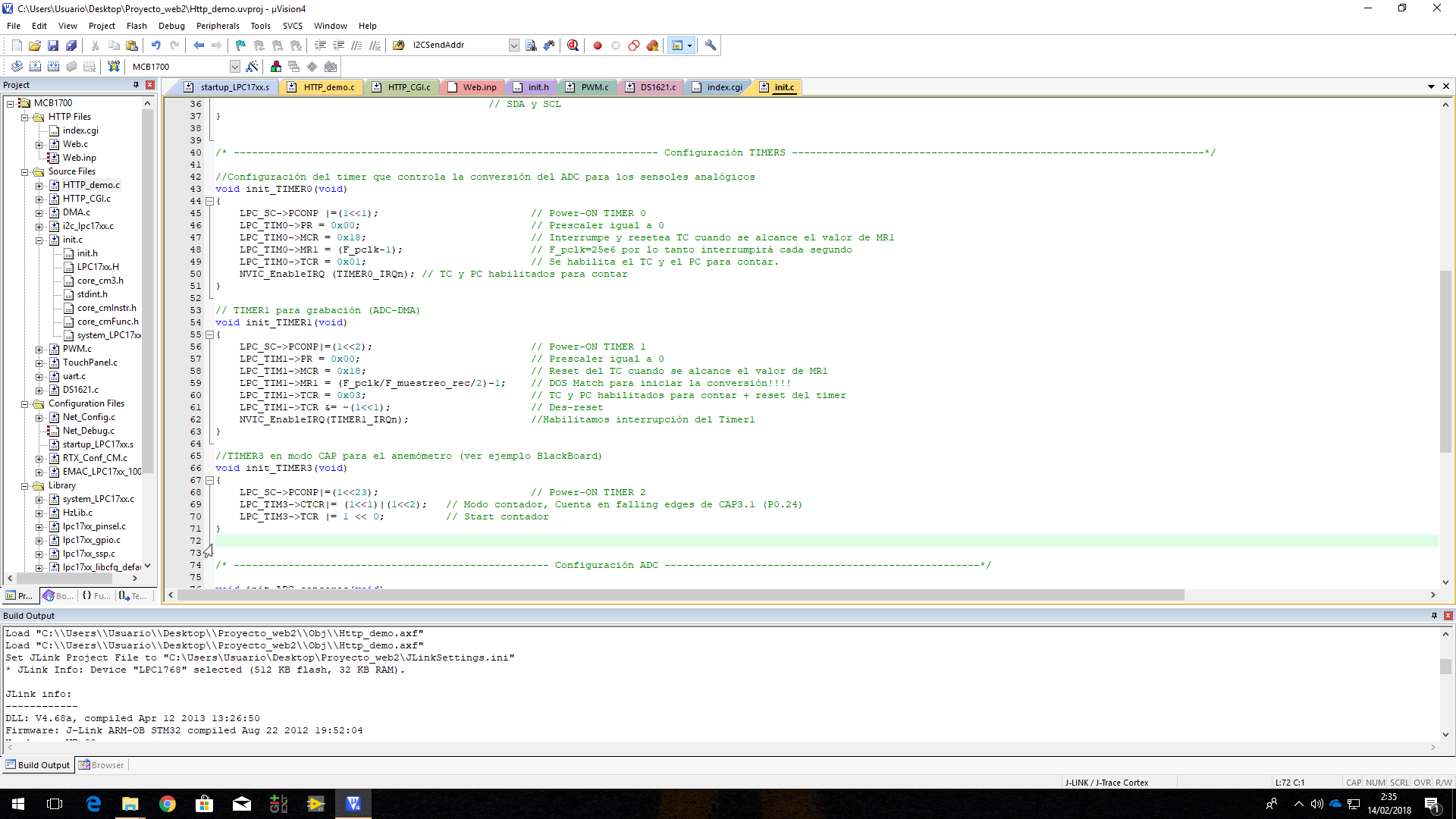
* Init\_ADC\_sensores(): configura el ADC para que muestree el canal 2 y 4 del ADC donde están conectados los sensores analógicos. Se configura para comenzar la conversión en función del Match1 del Timer0 por lo tanto muestrea cada 2 segundos.



* Init\_ADC\_grabar(): configura el ADC para grabar el sonido del micrófono y convertir las muestras de la alarma.



* Init\_TIMER1(): configura el TIMER1 para la grabación de sonido mediante el micrófono y la conversión mediante el ADC. Se configura para que el ADC muestree a 8 KHz.



* Init\_TIMER3(): configura el TIMER3 en modo CAP para calcular la velocidad del anemómetro.

