



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Título del trabajo: Automatización de una  
granja porcina

English tittle: Automation of a swine farm

*Autor/es*

Estibaliz Muñoz Bermejo

*Director/es*

Roberto Casas Nebra

ESCUELA DE INGENIERÍA Y ARQUITECTURA  
2016



## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Estibaliz Muñoz Bermejo,

con nº de DNI 18455060K en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado \_\_\_\_\_, (Título del Trabajo)

Automatización de una granja porcina

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 25 de Noviembre del 2016

Fdo: Estibaliz Muñoz Bemejo



# “Automatización de una granja porcina”

## RESUMEN

---

El trabajo que a continuación se presenta trata sobre la automatización de una granja porcina a través de diferentes sensores, los cuales permitirán que el ganadero conozca las condiciones en las que están los animales. El principal objetivo que se pretende conseguir es disminuir el trabajo de los ganaderos en dichas granjas.

Para llevar a cabo el proyecto se ha recopilado información acerca de cómo se automatizan actualmente las granjas porcinas y qué sensores se debían usar. Una vez obtenida esa información se han propuesto las especificaciones que tendrá el proyecto. Con las especificaciones marcadas y los sensores que formarán parte del proyecto elegidos, se ha diseñado el circuito esquemático. Cuando se ha acabado de diseñar esta parte se ha pasado a diseñar la PCB, para posteriormente fabricarla y soldarle los componentes necesarios para que funcione el prototipo. A la par de esta tarea se ha ido programando el firmware prototipo para que cumpla con las especificaciones marcadas en un principio.


El prototipo que se ha desarrollado en este proyecto es capaz de controlar sendos parámetros tales como la temperatura, la velocidad del viento, la humedad o la concentración de gases y actuar en consecuencia sobre ellos para que los animales estén en las mejores condiciones posibles dentro de la granja. También posee un interfaz sencillo y fácil de manejar, donde el usuario puede ver y cambiar la temperatura del interior de la granja y otros parámetros.

Finalmente, se ha logrado conseguir el dispositivo que se buscaba, un prototipo capaz de controlar los sensores necesarios para automatizar una granja porcina de una forma educativa. Este dispositivo se puede controlar de dos formas, a través del interfaz que lleva incorporado o bien inalámbricamente a través de Wi-Fi, esto le da cierta libertad al usuario ya que no es necesario que acuda a la granja a seleccionar la temperatura que quiere que haya en el interior de su granja.



## TABLA DE CONTENIDOS


1.	INTRODUCCIÓN.....	1
1.1.	Introducción .....	1
1.2.	Objetivos y alcance.....	1
1.3.	Planificación del trabajo .....	2
1.4.	Estructura de la memoria .....	4
2.	DESCRIPCIÓN GENERAL.....	5
2.1.	Introducción .....	5
2.2.	Condiciones ambientales.....	5
2.2.1.	Temperatura .....	6
2.2.2.	Gases .....	7
2.2.3.	Ventilación.....	7
2.3.	Sistema de comida.....	8
2.4.	Especificaciones .....	9
3.	DISEÑO DEL HARDWARE .....	10
3.1.	Introducción .....	10
3.2.	Diseño.....	10
3.2.1.	Actuadores.....	12
3.2.2.	Sensores.....	14
3.2.2.1.	Temperatura y humedad .....	14

 <b>Escuela de Ingeniería y Arquitectura</b> <b>Universidad Zaragoza</b>	Automatización de una granja porcina	
		Fecha revisión: 25/11/2016

3.2.2.2.	Gases.....	16
3.2.2.3.	Ultrasónico.....	17
3.2.2.4.	Final de carrera.....	18
3.2.2.5.	Anemómetro .....	19
3.2.3.	Interfaz.....	20
3.2.4.	Microcontrolador y comunicación.....	21
3.3.	Alimentación .....	22
3.4.	Diseño de PCB .....	25
4.	DISEÑO DEL FIRMWARE .....	28
4.1.	Introducción .....	28
4.2.	Estructura del código de programación.....	28
4.3.	Funcionamiento del dispositivo .....	29
4.3.1.	Inicio.....	30
4.3.2.	Modo usuario.....	30
4.3.3.	Modo programación .....	31
4.4.	Wi-Fi.....	33
5.	CONCLUSIONES .....	34
6.	BIBLIOGRAFÍA.....	36
ANEXO II		
	Anexo I.I. Plano cara top.....	I



Anexo I.II. Plano cara bottom .....	II
Anexo I.III. Esquemático .....	III
ANEXO II. COSTE DE LOS COMPONENTES .....	V
ANEXO III. CÓDIGO DE PROGRAMACIÓN .....	VI

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	LISTAS	Fecha revisión: 25/11/2016

## TABLA DE ILUSTRACIONES

Ilustración 1: Diagrama de Gantt	3
Ilustración 2: Esquema granja porcina	5
Ilustración 3: Diagrama de bloques	11
Ilustración 4: Configuración ULN2003	12
Ilustración 5: Esquemático actuadores	13
Ilustración 6: Telaire T9604	15
Ilustración 7: DHT 22	15
Ilustración 8: Esquemático sensor de temperatura y humedad	16
Ilustración 9: Sensor de gas.	16
Ilustración 10: Funcionamiento sensor electroquímico.	17
Ilustración 11: Esquemático del sensor de gases	17
Ilustración 12: Sensor de ultrasonidos.	18
Ilustración 13: Esquemático final de carrera	19
Ilustración 14: Anemómetro.	19
Ilustración 15: Esquemático anemómetro	20
Ilustración 16: Esquemático pulsador	21
Ilustración 17: Esquemático pantalla LCD	21
Ilustración 18: Flyport Wi-Fi.	22
Ilustración 19: LDO NCP5504DTRKG	22
Ilustración 20: Esquemático alimentación	23
Ilustración 21: Diseño PCB en 3D	26
Ilustración 22: Cara <i>bottom</i> soldada	26




Ilustración 23: Cara <i>top</i> soldada	27
Ilustración 24: Capas del código de programación	28
Ilustración 25: Máquina de estados general	29
Ilustración 26: Diagrama de flujo modo usuario	30
Ilustración 27: Diagrama de flujo modo programación	32
Ilustración 28: Plano cara <i>top</i>	I
Ilustración 29: Plano cara <i>bottom</i>	II

## ÍNDICE DE TABLAS

---

Tabla 1: Temperaturas de confort.	6
Tabla 2: Temperaturas según la época del año	6
Tabla 3: Concentración máxima de gases.	7
Tabla 4: Comparativa sensores de temperatura y humedad	15
Tabla 5: Consumos y tensiones de los componentes	23
Tabla 6: Consumos alimentación modelo estudiante	24
Tabla 7: Coste de los componentes	V




 <b>Escuela de Ingeniería y Arquitectura</b> <b>Universidad Zaragoza</b>	Automatización de una granja porcina	
	LISTAS	Fecha revisión: 25/11/2016


## LISTA DE SIGLAS Y ABREVIATURAS

---

<b>CI</b>	Circuito Integrado
<b>cm</b>	Centímetros
<b>CO2</b>	Dióxido de carbono
<b>EEPROM</b>	Erasable Programmable Read-Only Memory
<b>GPIO</b>	General Purpose Input / Output
<b>H2S</b>	Ácido sulfhídrico
<b>HTML</b>	HyperText Markup Language
<b>kB</b>	Kilobyte
<b>LCD</b>	Liquid Cristal Display
<b>LDO</b>	Low Dropout regulator
<b>m/s</b>	Metros dividido por segundo
<b>mA</b>	Miliamperios
<b>mm</b>	Milímetros
<b>NH3</b>	Amoníaco

 <b>Escuela de Ingeniería y Arquitectura</b> <b>Universidad Zaragoza</b>	Automatización de una granja porcina	
	LISTAS	Fecha revisión: 25/11/2016

<b>°C</b>	Grados Celsius
<b>PCB</b>	Printed Circuit Board
<b>ppm</b>	Partes Por Millón
<b>RTC</b>	Real Time Clock
<b>RTD</b>	Resistance Temperature Detector
<b>TCi</b>	Temperatura Crítica Inferior
<b>TCs</b>	Temperatura Crítica Superior
<b>TFG</b>	Trabajo de Final de Grado
<b>THL</b>	Through Hole Device
<b>V</b>	Voltios
<b>Wi-Fi</b>	Wireless Fidelity
<b>μA</b>	Microamperios

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

# 1. INTRODUCCIÓN

## 1.1. Introducción

El proyecto que se presenta trata sobre la automatización de una granja porcina, en concreto las granjas de engorde o también llamadas granjas de cebo. También se podría extender a las granjas de maternidad, destete y gestación, pero debido a que hay que tener muchos más cuidados es más complicado realizar un control correcto. Con este proyecto se intenta hacer el trabajo de los ganaderos más fácil ya que estos no tienen que estar tan pendientes del bienestar de los animales.

Este proyecto se ha realizado junto con el Departamento de Electrónica y Comunicaciones. Además parte desde cero, aunque se basa en la línea de trabajo que se siguió en la asignatura de Laboratorio de Diseño Electrónico.


En un principio este proyecto se pensó para que automatizara una granja entera, pero debido a la repetición de la mayoría de los sensores, se pensó, junto con el director del proyecto, que lo más óptimo para no repetir tantos sensores, ya que la Universidad no los podía proporcionar, era programar una versión de estudiante, con un solo sensor de cada tipo. Por lo tanto se tuvo que modificar la PCB, como se puede observar en el apartado 3.4 colocando un nuevo microcontrolador.

## 1.2. Objetivos y alcance

El objetivo de este TFG es desarrollar un prototipo funcional para que una granja porcina pueda ser automatizada, basándose en el uso de numerosos sensores los cuales harán que los animales estén en las mejores condiciones posibles.

Para llevar a cabo este objetivo se han planteado varios objetivos más pequeños:

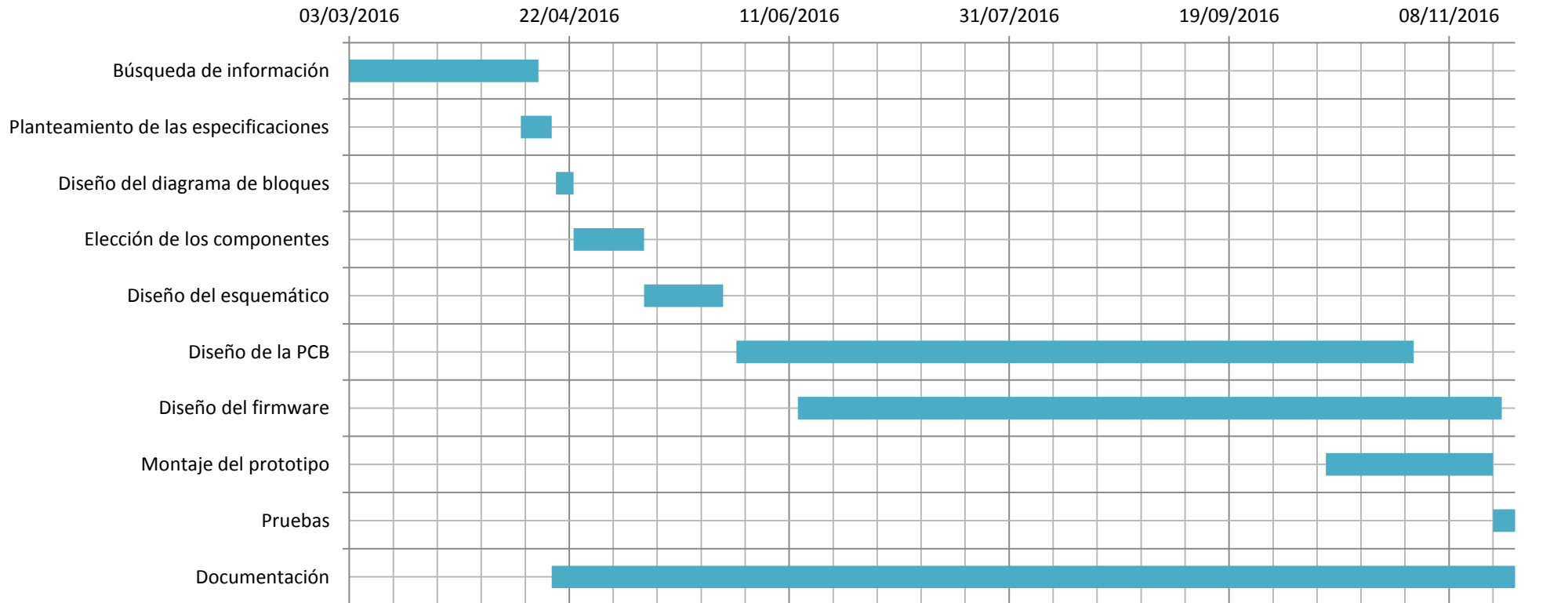
- Búsqueda de información: Se ha tratado de buscar información acerca de cómo se puede automatizar una granja porcina y los parámetros de confort de los animales.
- Planteamiento de las especificaciones: Una vez obtenida toda la información relevante, se sintetiza para conseguir unas especificaciones claras para poder desarrollar el proyecto.
- Diseño electrónico: Este objetivo va a tener subapartados más pequeños que permitan obtener un buen diseño:

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

- Diseño del diagrama de bloques: En primer lugar se va a diseñar el diagrama de bloques, con el cual se asegura que se cumplen todas las especificaciones marcadas.
  - Elección de los componentes: Una vez realizado el diagrama de bloques, se elegirán los componentes principales de los que está compuesto el proyecto.
  - Diseño del esquemático: Cuando se han elegido los componentes principales, se diseña el esquemático de los diferentes bloques teniendo en cuenta las características de los componentes elegidos y del proyecto.
  - Diseño de la PCB: Por último, en esta fase, se diseña la PCB. Hay que tener en cuenta la colocación de los componentes, tamaño de las pistas, etc.
- Diseño del firmware: A la vez que se va diseñando la PCB, se trabaja con el Flyport para ir programando los diferentes sensores y así poder probarlos.
  - Montaje del prototipo: Una vez se ha fabricado la PCB, se monta, siguiendo las pautas de la asignatura de Laboratorio de Diseño Electrónico.
  - Pruebas: Cuando ya está montada la PCB y diseñado el firmware, se hacen una serie de pruebas para comprobar que se ha programado correctamente y corregir los posibles fallos.
  - Documentación: A lo largo de todo el proyecto se va preparando la documentación pertinente para finalmente poder realizar una memoria completa.


### 1.3. Planificación del trabajo

Como forma de asegurar el cumplimiento de plazos y así tener el trabajo a punto para la fecha de entrega se establece un diagrama de Gantt (Ilustración 1), en el que se aprecian los plazos que hay que cumplir para completar cada fase del proyecto. La duración de cada fase es aproximada.



	Documentación	Pruebas	Montaje del prototipo	Diseño del firmware	Diseño de la PCB	Diseño del esquemático	Elección de los componentes	Diseño del diagrama de bloques	Planteamiento de las especificaciones	Búsqueda de información
Fecha de inicio	18/04/2016	18/11/2016	11/10/2016	13/06/2016	30/05/2016	09/05/2016	23/04/2016	19/04/2016	11/04/2016	03/03/2016
Duración	219	5	38	160	154	18	16	4	7	43

Ilustración 1: Diagrama de Gantt

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

## 1.4. Estructura de la memoria

Para poder documentar este TFG de una forma completa se va a desarrollar una memoria en la que se mostrarán las tareas realizadas, así como toda la información correspondiente a este proyecto. También se incluyen unos anexos en los que se contempla la información menos relevante de cara a comprender el proyecto.

Los apartados o capítulos en los que se estructura la memoria son:

- Capítulo 1: En este capítulo se exponen los objetivos y el alcance del proyecto, la planificación temporal del mismo y una breve descripción de los capítulos o apartados de los que se compone la memoria.
- Capítulo 2: En este segundo capítulo se describe de una forma general una granja porcina, como se distribuyen sus sensores, los parámetros de confort de los cerdos, cómo funciona el sistema de alimentación más común en dichas granjas y por último las especificaciones que regirán el proyecto.
- Capítulo 3: En este apartado se muestra como se ha realizado el diseño del hardware, desde el diseño del diagrama de bloques, pasando por la realización del esquemático hasta acabar con el diseño de la PCB.
- Capítulo 4: En este apartado se plantea el diseño del firmware, donde se expone cómo funciona el dispositivo y la estructura que tiene la programación del mismo.
- Capítulo 5: En este capítulo se plantean las conclusiones finales del trabajo, para ello se contrastan las especificaciones que se habían planteado al principio del proyecto con lo que se ha conseguido finalmente. También se abordan las líneas futuras de trabajo.
- Capítulo 6: En este último apartado se muestra tanto la bibliografía como la linkografía de los trabajos y páginas web citadas en la memoria.
- Anexo I: En este anexo se incluyen los planos del diseño del hardware.
- Anexo II: En este anexo se describe de forma detallada el coste de los componentes que forman el dispositivo.
- Anexo III: En este anexo se detalla el código del programación del dispositivo.



## 2. DESCRIPCIÓN GENERAL

### 2.1. Introducción

En este capítulo se explica cómo se reparten los sensores en las granjas, además también se habla de los parámetros de confort de los cerdos y del sistema de alimentación más extendido entre las granjas. Por último, se plasman las especificaciones que deberá tener el proyecto.

### 2.2. Condiciones ambientales

Mantener a los animales con unas condiciones ambientales correctas es lo ideal para que estos no enfermen. De esta forma el ganadero se ahorra mucho tiempo y dinero en medicamentos u otros cuidados que requiere un animal enfermo.

Actualmente, la mayoría de las granjas porcinas dedicadas al cebo de los animales, se dividen en 3 para colocar los sensores, es decir, hay un sensor de temperatura y humedad y otro de gases cada 1/3 de la nave, como se puede apreciar en la Ilustración 2. Esto es así ya que puede que la temperatura dentro de la nave no sea la misma en un tercio que en otro. Para la apertura y cierre de las ventanas, las granjas se suelen dividir en 4 lados, para que los motores que las abren y cierran no soporten tanta carga.

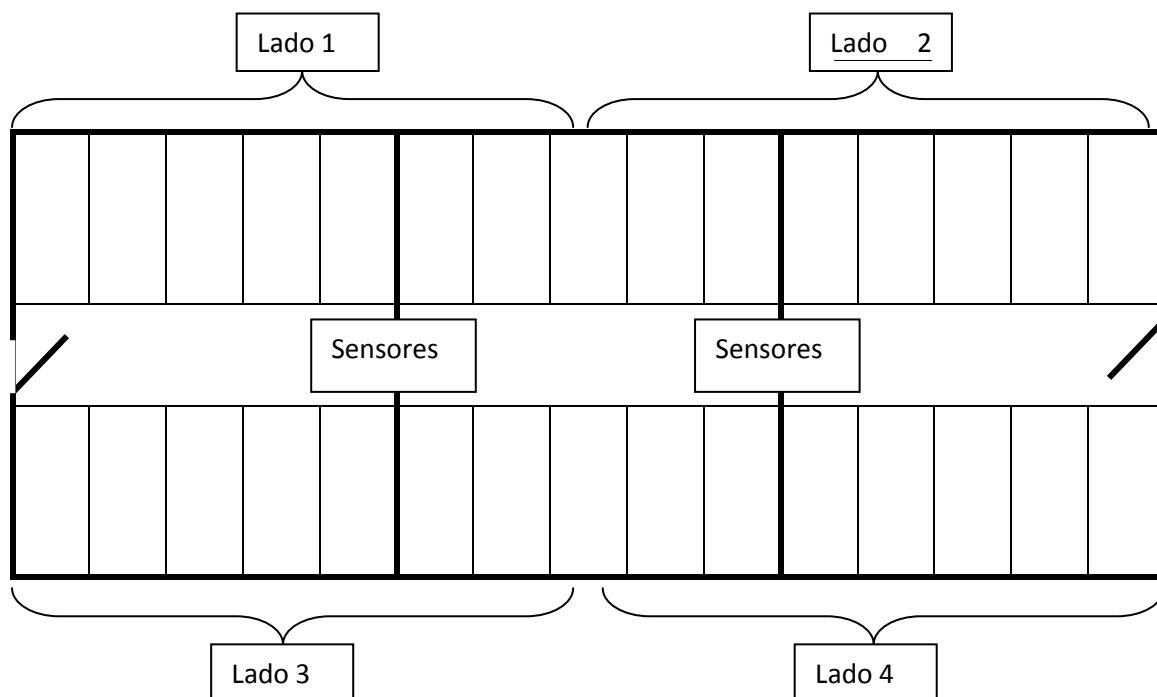


Ilustración 2: Esquema granja porcina



## 2.2.1. Temperatura

En cuanto a los parámetros de temperatura, varían mucho si se habla de una granja de cebo o de una granja de maternidad y destete. En las granjas de cebo o engorde, que son para las que está destinado este proyecto, hay que tener cuidado con el frío en la entrada de los cerdos lechones en la granja, y posteriormente, cuando van engordando tienen calor, por lo que hay que controlar bien tanto la temperatura, como se aprecia en la Tabla 1, como la ventilación. Así mismo, también hay que variar la temperatura depende de la época de año en la que estén (Tabla 2).

Peso (Kg)	TCi (°C)	TCs(°C)
20	21	27
30	20	26
60	17	23
110	16	22

Tabla 1: Temperaturas de confort.  
Fuente: <http://www.itgganadero.com/>

TCi es la temperatura mínima que permite el mayor crecimiento, mientras que TCs es la temperatura máxima que permite el mayor crecimiento, por lo que es muy importante que la temperatura del interior de la granja esté dentro de este rango.

Enero - Febrero	Marzo	Abril - Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre - Diciembre
22 °C	23 °C	24 °C	25 °C	27 °C	26 °C	25 °C	23 °C	22 °C

Tabla 2: Temperaturas según la época del año  
Fuente: <http://www.itgganadero.com/>

En la Tabla 2 se muestran las temperaturas ideales según el mes del año en el que nos encontremos, para que no se produzcan infraventilaciones en invierno o sobreventilaciones en verano.





## 2.2.2. Gases

El control de la calidad del aire es importante, no solo para los animales sino también para los ganaderos. Una mala ventilación o exceso de gases en la granja puede afectar a los cerdos provocando daños en los ojos, la mucosa o la piel.

Los gases más peligrosos son los que provienen de los desperdicios de los animales, tal y como son el NH<sub>3</sub>, el CO<sub>2</sub> o el H<sub>2</sub>S:

- El amoníaco es un gas tóxico que se genera cuando se mezclan las heces con la orina, por lo que está muy presente en este tipo de explotaciones.
- El dióxido de carbono se produce en los pulmones de los cerdos, pero si hay una buena ventilación la concentración de este gas se mantiene por debajo de los niveles de riesgo.
- El ácido sulfhídrico es un gas tóxico que puede producir la muerte de los animales, este gas se forma por la descomposición anaerobia de los excrementos, pero si la granja posee de un buen sistema de evacuación de los excrementos no habrá problemas de mortandad a priori.


En la Tabla 3 se muestra la concentración máxima de gases que puede haber en las explotaciones de animales.

Gas	Concentración, ppm
Amoníaco (NH <sub>3</sub> )	20
Dióxido de carbono (CO <sub>2</sub> )	3000
Monóxido de carbono (CO)	10
Ácido sulfhídrico (H <sub>2</sub> S)	0,5

Tabla 3: Concentración máxima de gases.  
Fuente: <https://www.3tres3.com/>

## 2.2.3. Ventilación

Las granjas porcinas deben tener una correcta ventilación, si no es así los gases se acumulan en el interior de la granja produciendo enfermedades a los animales como se ha

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

visto en la sección 2.2.2. En verano apenas hay problemas de acumulación de gases ya que para mantener una buena temperatura se ventila constantemente, pero en invierno, cuando las temperaturas exteriores son muy bajas, se produce suele producir una mala ventilación, por lo que hay que atender a la concentración de gases para que esto no ocurra.

Hay diferentes tipos de ventilación [1]:


- Ventilación natural: Este tipo de ventilación funciona por la velocidad del viento, el inconveniente principal es que no se puede controlar la cantidad de aire que se ventila. La ventilación natural está formada por una abertura en el techo y de las ventanas.
- Ventilación forzada: Estos sistemas se pueden dividir en 3:
  - o Sistema de presión positiva: En este tipo de ventilación se impulsa el aire hacia el interior de la granja a través de ventiladores saliendo por diferentes aberturas que posee la granja.
  - o Sistema de presión neutra: En este tipo de sistemas de ventilación el aire es impulsado hacia el interior de la granja con unos ventiladores y sale de la misma a través de extractores.
  - o Sistema de presión negativa: En este tipo de ventilación se hacen unas entradas de aire en las paredes de la granja, que dirigen el aire hacia el centro de la granja, el cual se mezcla con el aire caliente que hay en el interior.

Este proyecto se programará para una ventilación natural, aunque también se podría programar para cualquier tipo de ventilación forzada puesto que lo que se controla son los relés que activan los motores como se muestra en la sección 3.2.1.

## 2.3. Sistema de comida

El sistema de comida que se implementa actualmente en las granjas porcinas es muy sencillo, un motor unido a un tornillo sinfín impulsa la comida a través de las tuberías y esta va cayendo a los silos hasta llegar al último de los silos, en el cual hay un final de carrera que activa o desactiva dicho motor.

Para saber si queda o no comida se usarán sensores ultrasónicos, los cuales se colocarán en el interior del silo. Debido a que según el tipo de pienso que haya en el silo, este


 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

cae de una forma u otra, se colocarán tres sensores para saber cómo está cayendo el pienso y cuanto queda.

## 2.4. Especificaciones

Una vez se han analizado los parámetros de confort de los animales y se ha diseñado el sistema de comida, se deben plasmar las especificaciones que ha de cumplir el proyecto, las cuales son:

- **Control de la ventilación** a través de la apertura y cierre de las ventanas, dependiendo del sensor de gases, del sensor de temperatura y humedad y de la velocidad del viento tal y como se muestra a continuación:
  - o Cuando la concentración de amoníaco sea superior a 20 ppm y/o la concentración de ácido sulfhídrico sea superior a 0.5 ppm.
  - o Cuando la temperatura sea superior a la introducida por el usuario.
  - o Cuando la humedad interior supere el límite superior, siempre y cuando la humedad exterior sea menor.
  - o Solamente se podrán abrir las ventanas cuando la velocidad del viento no sea superior a la indicada por el usuario.
- **Control de los comederos**, si el último comedero se vacía, se debe activar el mecanismo que proporciona la comida a los comederos.
- **Detectar cuando hay poca comida en el silo** y avisar al ganadero si la comida que hay en el silo es inferior a 1/3 del silo.
- **Medir temperaturas** en un rango de 0 a 40°C y humedad relativa entre el 10 al 90 %.
- **Medir velocidad** del viento entre 0 m/s y 32 m/s.
- **Sistema intuitivo y fácil de usar.**

 <b>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

## 3. DISEÑO DEL HARDWARE

### 3.1. Introducción

En este capítulo se aborda la parte correspondiente al diseño del hardware de este proyecto. Para ello se proponen diferentes bloques, dando solución a cada uno de ellos por separado. Dicha solución consiste en plantear el esquemático de cada bloque y seleccionar los componentes que mejor se ajusten al proyecto.

Una vez se ha realizado el esquemático del proyecto y se han seleccionado los componentes, se diseña la PCB y posteriormente se fabrica, lo que permite tener un prototipo funcional del proyecto.

### 3.2. Diseño

En este apartado se afronta el diseño del diagrama de bloques. Como se ha comentado con anterioridad, se propone un diseño de forma modular, identificando cada bloque y proponiendo una solución a cada uno de ellos.

En la Ilustración 3 se pueden apreciar los diferentes bloques por los que está compuesto el proyecto:

- Bloque de Actuadores: Es el encargado de ejecutar las acciones que le solicita el microcontrolador. Está compuesto por diversos relés que tienen como misión activar y desactivar los motores para subir y bajar las persianas, así como activar y desactivar el mecanismo para echar comida en los comederos.
- Bloque de Sensores: Proporciona la información necesaria para que el microcontrolador ejecute sus tareas. Este bloque está formado por finales de carrera, sensores de temperatura y humedad, de gases, ultrasónicos y un anemómetro.
- Bloque de Interfaz: Permite al usuario que pueda interactuar con el proyecto. Está compuesto por una pantalla LCD y pulsadores.
- Bloque de Comunicaciones: Está compuesto por un módulo Wi-Fi y permite una comunicación bidireccional del usuario y el dispositivo.
- Bloque de Microcontrolador: Es el núcleo del proyecto y se encarga de gestionar cada uno de los bloques restantes.
- Bloque de Alimentación: Este bloque se encarga de suministrar la alimentación a todo el dispositivo.

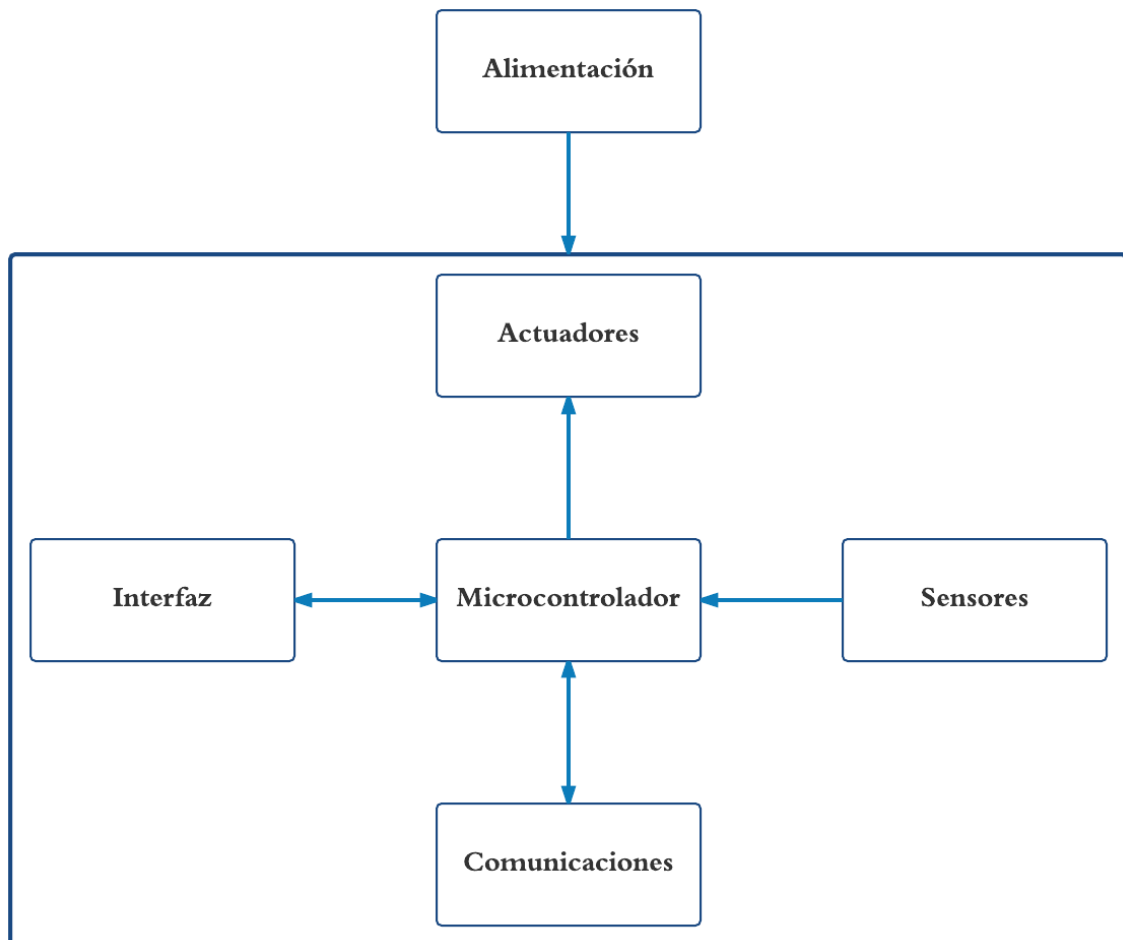


Ilustración 3: Diagrama de bloques

A continuación se desglosan cada uno de los bloques. Cada sección comenzará planteando las razones por las que se ha incluido dicho bloque en el proyecto, posteriormente se definirán tanto las características de cada bloque como los componentes que lo integran, y por último se expondrán los requisitos de cada bloque en lo referente a la elección del microcontrolador.



### 3.2.1. Actuadores

Este es un bloque bastante importante, pues controla dos partes fundamentales en lo referente al bienestar de los animales. En primer lugar controla la ventilación de la granja (junto con los sensores de gases, de temperatura y humedad y anemómetro), esto es vital, ya que con una mala ventilación los cerdos podrían morir, y por otra parte controla la alimentación (junto a los finales de carrera).

En cuanto a la ventilación, se ha provisto el proyecto con 4 relés, los cuales controlan los motores que abren y cierran las ventanas. Y en lo que se refiere a la alimentación se han colocado 2 relés, que activan y desactivan los motores que hacen que vaya el pienso desde el silo hasta el comedero.

Los relés que se han elegido han sido de propósito general ya que son de los más baratos que se pueden encontrar en el mercado y una de sus aplicaciones es el control y la automatización de procesos. Estos relés son de la marca FINDER cuya referencia es 36.11.9.012.4011.

Por otra parte, para la activación de los relés, en primer lugar se barajó poner transistores y diodos de protección a cada uno de ellos, pero ocupaba bastante espacio en la PCB. Por lo que se decidió colocar un circuito integrado de 7 transistores tipo Darlington para su activación, este CI también lleva incorporados los diodos de protección como se puede ver en la Ilustración 4. El integrado que se eligió fue el ULN2003 [2] ya que se puede conectar a 3.3 V

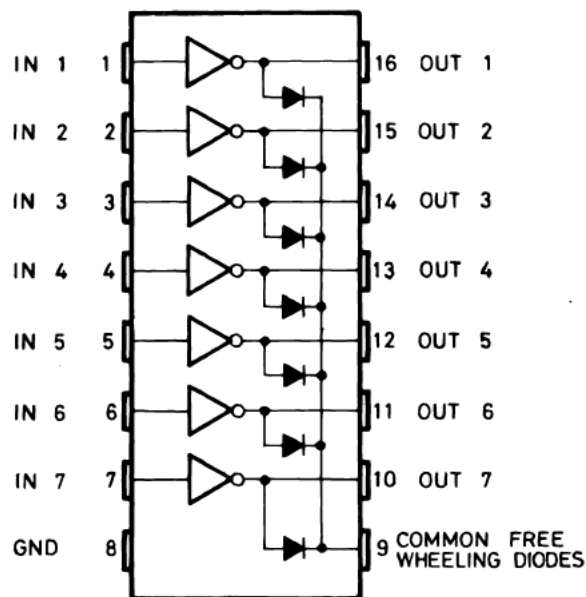


Ilustración 4: Configuración ULN2003

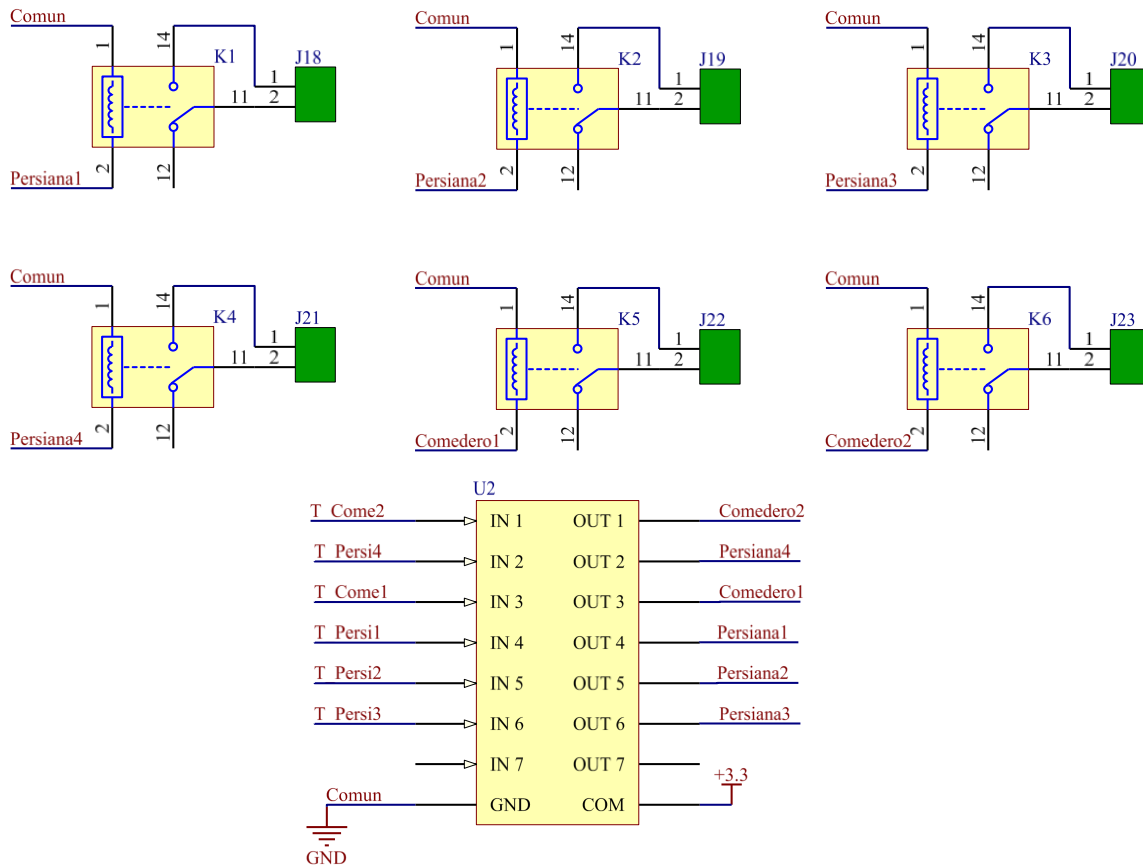



Ilustración 5: Esquemático actuadores

En la Ilustración 5, se aprecia la parte de los actuadores del el esquemático. Para que los relés puedan conmutar se necesitan transistores que le den suficiente corriente al microcontrolador, para ello se usa el CI ULN2003 (U2). Además como se ha comentado con anterioridad este CI también posee los diodos que protegen al resto del circuito frente a la corriente generada por la bobina. Su conexión con el microcontrolador es muy sencilla, los pines IN van al microcontrolador, mientras que los pines OUT se conectan a los relés.

A su vez, también se muestra la conexión de los relés con los motores, esto se hace a través del conector; en el pin 2, el cual es el común, conectaremos la masa, mientras que al pin 1 se conectará el motor. El motor se alimentará externamente a 12 V, los cuales procederán de las placas solares o baterías de las que este dotada la explotación porcina.

 <b>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

Como lo que se programa finalmente, es un prototipo educativo, solo se incluirán en la PCB dos relés, uno para simular el control de las ventanas y otro para simular el control de los comederos.

### 3.2.2. Sensores

Este bloque, junto con el bloque de sensores componen la parte fundamental para que se realice un buen control de la explotación, por lo que hay que elegir muy bien cada sensor para que no se produzcan fallos.

Como finalmente lo que se programa es un proyecto más educativo, como se ha comentado en la sección 1.1, se han usado los sensores que ya poseía la Universidad. Si este proyecto se llevara a la realidad habría que proteger correctamente estos sensores para que no se ensuciaran y sus medidas fueran lo más fiables posible.

Los sensores sobre los que se hablará a continuación son: sensor de temperatura y humedad, sensor de gases, sensor ultrasónico, anemómetro y final de carrera. Estos son sensores con los que va a contar el proyecto.

#### 3.2.2.1. *Temperatura y humedad*

Se necesita conocer la temperatura y humedad que hay en el interior de la granja, y así poder actuar cuando estas sobrepasan los valores límite, los cuales se encuentran en la sección 2.2.1. Además se debe conocer la temperatura exterior para poder actuar en consecuencia sobre los motores que abren y cierran las ventanas de la explotación porcina. También hay que conocer que humedad hay en el exterior, ya que si la humedad del interior supera los límites habría que abrir las ventanas para que esta disminuyese.

Hay una gran cantidad de sensores y tecnologías diferentes para medir temperatura, las más conocidas son los termistores, RTD o termopares. También hay varias tecnologías para medir humedad, como por ejemplo la psicometría, por condensación o sensores mecánicos o capacitivos. Además de todos estos tipos de sensores, hay uno que es capaz de medir temperatura y humedad en un mismo sensor, este es el sensor de temperatura y humedad integrado, el cual se usará en el proyecto debido a que es mucho más compacto.

En primer lugar se propuso el sensor Telaire T9602<sup>1</sup> (Ilustración 6) ya que su índice de protección es muy alto, pero se descartó debido a su elevado precio, además posee unas

---

<sup>1</sup> <http://www.mouser.com/ds/2/18/AAS-920-638F-Telaire-T9602-060316-web-850549.pdf>





características similares a las del sensor DHT22 [3] (Ilustración 7) en cuanto a rangos de medida de temperatura y humedad. Por lo que finalmente se concluyó que el sensor DHT22 era el que más encajaba en el proyecto debido a sus características y precio y además ya lo poseía la universidad.



Ilustración 6: Telaire T9604  
Fuente: <http://www.mouser.es/>



Ilustración 7: DHT 22  
Fuente: <http://www.mouser.es/>

A continuación, en la Tabla 4 se muestran las diferencias entre ambos sensores.

	Telaire T9602	DHT22
Rango de temperatura	-20 °C a 70 °C	-40 °C a 80 °C
Rango de humedad	0 % a 100 %	0 % a 100 %
Resolución temperatura/ humedad	0.01 °C / 0.01 %	0.1 °C / 0.1 %
Tipo de salida	I2C o analógica	1-wire
Voltaje	3.3 V o 5 V	3 V a 5 V
Consumo	750 $\mu$ A	2.5 mA
Índice de protección	IP67	-

Tabla 4: Comparativa sensores de temperatura y humedad

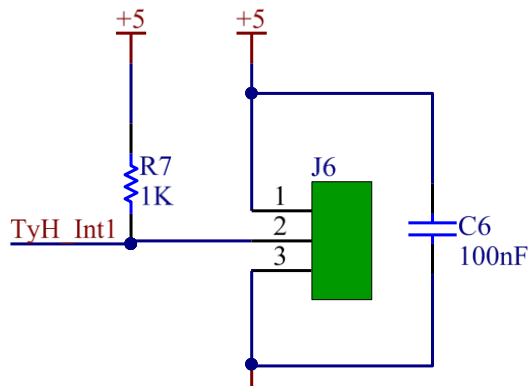


Ilustración 8: Esquemático sensor de temperatura y humedad

En la Ilustración 8, se puede ver el esquemático de este sensor, al cual se le ha incorporado una resistencia, R7, entre la línea de datos, pin 2, y masa tal y como se especifica en el datasheet del componente, además se incorpora un condensador, C6, entre masa y alimentación. En cuanto al resto de pines, se conectan fácilmente, el pin 1 se conecta a la alimentación, mientras que el pin 3 se conecta a masa.

### 3.2.2.2. Gases

El sensor de gases es muy importante en una granja porcina, pues los mismos animales y sus desechos desprenden gases que pueden ser nocivos para la salud. Los gases que más se controlan en este tipo de explotaciones son amoníaco (NH<sub>3</sub>), dióxido de carbono (CO<sub>2</sub>) y ácido sulfhídrico (H<sub>2</sub>S), como se puede observar en la sección 2.2.2.

Debido a que no se encontró un sensor que integrara la detección de los 3 gases principales que hay en una explotación porcina, se optó por elegir un sensor que detectara los gases más perjudiciales para la salud, los cuales son el NH<sub>3</sub> y H<sub>2</sub>S. Por este motivo se eligió el sensor TGS 2609-B00 [4] de Figaro.

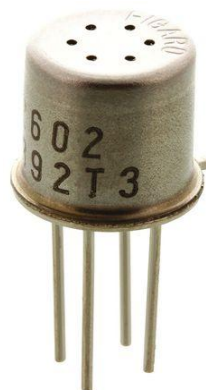


Ilustración 9: Sensor de gas.  
Fuente: <http://es.rs-online.com/>

En el mercado actual hay una gran variedad de tipos de sensores de gases [5], como por ejemplo, sensores electroquímicos, sensores catalíticos o sensores semiconductores. El sensor que se ha elegido es de tipo electroquímico (Ilustración 9), los cuales tienen dos

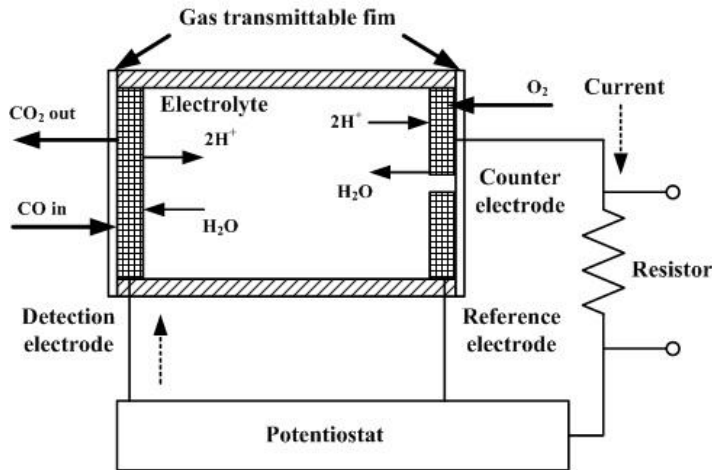


Ilustración 10: Funcionamiento sensor electroquímico.  
Fuente: <https://www.330ohms.com/>

En la Ilustración 11, se puede observar el esquemático de dicho sensor, en el cual se aprecia una resistencia, R13, entre el pin 3 y masa, tal y como indica el fabricante, además de un condensador, C10, entre masa y alimentación. El pin 1, es el pin de alimentación, el pin 2, alimenta a la resistencia *heater*, el pin 3 es la entrada al microcontrolador, la cual es analógica, y por último, el pin 4 se conecta a masa.

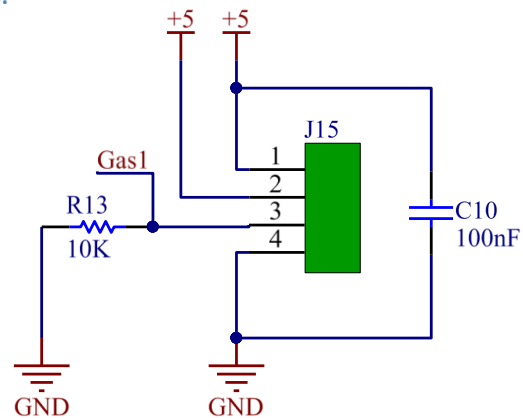


Ilustración 11: Esquemático del sensor de gases

### 3.2.2.3. Ultrasonico

El sensor ultrasónico se usa para medir la cantidad de comida queda en el silo y así, avisar al ganadero cuando quede poca comida. En un principio, para llevar a cabo esta tarea se buscaron sensores de peso, como por ejemplo células de carga.

Las células de carga miden la compresión a la que está sometida dicha célula, esta compresión hace que varíe la longitud del hilo metálico del que están compuestas las galgas, lo cual provoca una variación de la resistencia eléctrica. Esta variación de resistencia nos proporcionaría, a través de cálculos, el peso que soporta la célula. Por lo que es un buen sensor para conocer el peso del silo, pero estas células tienen un precio muy elevado, y además es complicado colocarlas de forma correcta para poder tomar buenas medidas en los silos.



Entonces se propuso otra forma con la que ver cuanta comida queda en los silos. Esta forma es midiendo la distancia que hay desde lo alto del silo hasta la comida, con un sensor ultrasónico. El sensor que se eligió fue el HC-SR04 [6], ya que tiene un rango muy amplio de medida, de 2 cm a 400 cm.

Su conexión es muy simple, como se aprecia en la Ilustración 12, tiene 4 pines, el primero de ellos, Vcc, se conecta directamente a la alimentación, el segundo, Trig, es el emisor, el tercero, Echo, es el receptor y por último, el cuarto pin se conecta a masa.



Ilustración 12: Sensor de ultrasonidos.  
Fuente: <http://electrotec.pe/>

#### 3.2.2.4. Final de carrera

El sistema de comida, como se ha explicado en la sección 2.3, se compone de los motores que junto con los tornillos sinfín hacen avanzar la comida a lo largo de las tuberías y esta va cayendo a los comederos. Los finales de carrera se usan para saber cuando están llenos los comederos y así poder parar el motor.

En primer lugar se propusieron finales de carrera especiales<sup>2</sup> para granjas porcinas, pero finalmente se eligieron finales de carrera que ya poseía la universidad, ya que funcionan de la misma forma.

El final de carrera se conecta al microcontrolador a través de una resistencia *pull-up* y un filtro para evitar los posibles rebotes, como se puede apreciar en la Ilustración 13.

---

<sup>2</sup> <https://tiendaganadera.com/final-de-carrera/>

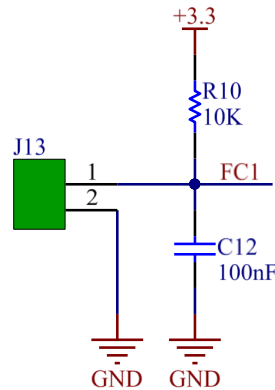


Ilustración 13: Esquemático final de carrera

### 3.2.2.5. Anemómetro

El último sensor que incorpora el trabajo es el anemómetro, el cual usaremos para medir la velocidad del viento. Lo idóneo sería tener una estación meteorológica, con la que poder medir, a parte de la velocidad del viento, la dirección, para saber qué persianas debemos subir o bajar para que los cerdos no salgan de su zona de confort.

Debido a que las estaciones meteorológicas son muy caras, se propuso medir únicamente la velocidad del viento, la cual es muy importante, pues si esta es muy elevada y el aire es frío, los cerdos se pueden resfriar lo que repercute de forma económica en los ganaderos.

El anemómetro que se ha elegido ha sido el 1773 de Adafruit, el cual se puede apreciar en la Ilustración 14. Su salida es lineal y va de 0.4 V a 2 V, indicando con 0.4 V que no hay viento y con 2 V que el viento es de 32.4 m/s.

A continuación, en la Ilustración 15 se puede observar el esquemático del anemómetro. Su conexión es simple, pues solamente hay que alimentarlo y llevarlo a masa. También se ha colocado un divisor resistivo, en la línea de datos, para adecuar su señal, pero puesto que la salida del mismo solamente va de 0.4 V a 2.0 V no haría falta adecuar dicha señal. Por lo que en futuras versiones del proyecto se debería de quitar el divisor resistivo formado por las resistencias R12 y R15.



Ilustración 14: Anemómetro.  
Fuente: <https://www.adafruit.com>

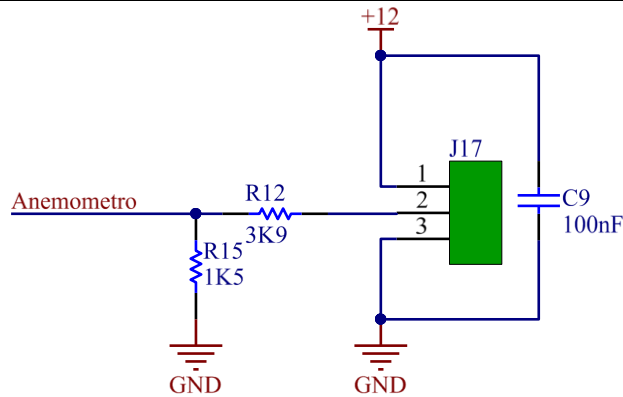


Ilustración 15: Esquemático anemómetro

### 3.2.3. Interfaz

Este bloque se propuso para que el usuario pudiera tener constancia de los parámetros que regulan el confort de los cerdos, tal y como es la temperatura y humedad tanto exterior como interior, la velocidad del viento y la cantidad de comida que queda en los silos. Además también se pensó para que el ganadero pudiera establecer la temperatura de la granja, dependiendo de la cantidad de cerdos que hay en ella y la edad de los mismos, y también el resto de parámetros.

Para que esto se pudiera llevar a cabo se propuso colocar una pantalla LCD, en la que se pudieran ver los parámetros y además varios pulsadores con los que aumentar o disminuir la temperatura.

La pantalla LCD que se eligió fue la 16244 de Datavision [7], ya que estaba en la universidad y se controla fácilmente. Su conexión no es compleja puesto que solo hay que alimentarla, ajustar el brillo con un potenciómetro y conectarla al micro como se puede ver en la Ilustración 17. El pin RS sirve para seleccionar el registro de datos, pin R/W lo llevamos a masa, puesto que siempre se va a escribir sobre la pantalla, el pin E es el que va a permitir que se pueda habilitar o deshabilitar el *display* y por último los pines DB son las señales de datos, en este caso solamente se usa un bus de datos de 4 bits (DB7-DB4).

El proyecto cuenta con 3 pulsadores, uno de ellos es para seleccionar el parámetro que se va a cambiar y los otros dos pulsadores restantes son para aumentarlo o disminuirlo. En la Ilustración 16 se aprecia el esquemático del pulsador, el cual se conecta al microcontrolador a través de una resistencia *pull-up* y un filtro para evitar los posibles rebotes, al igual que los finales de carrera.

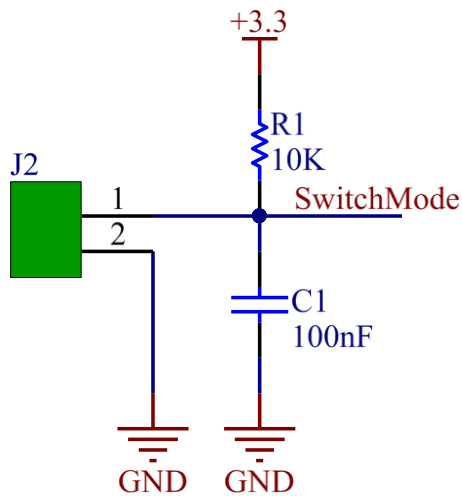


Ilustración 16: Esquemático pulsador

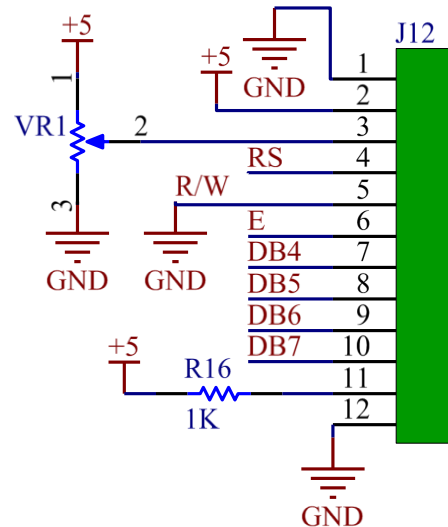


Ilustración 17: Esquemático pantalla LCD

### 3.2.4. Microcontrolador y comunicación

Este es el bloque más importante de todos, pues es el que controla y gestiona toda la información que recibe a través de los diferentes componentes que forman el proyecto. Dado que en el departamento en el que se ha realizado el proyecto se suele trabajar con módulos Flyport, se optó por estudiar las opciones que tenían este tipo de módulos y elegirlo para el proyecto si encajaba.

En un principio se optó por elegir el módulo FlyportPRO Wi-Fi<sup>3</sup>, ya que tiene 40 pines disponibles, contando con masa y alimentación. Para poder controlar todos los componentes del proyecto se necesitan 29 pines GPIO, por lo que tenemos pines suficientes para conectar todos los componentes. Además se tienen dos pines de alimentación y otros dos de masa, puesto que este módulo consta de dos conectores, como se puede apreciar en su datasheet. Pero debido a que se va a programar un modelo de estudiante, se va a usar el módulo Flyport Wi-Fi [8], ya que no se necesitan tantos pines.

La principal diferencia entre ambos módulos, a parte del número de patillas, es que el FlyportPRO Wi-Fi cuenta con una memoria EEPROM de 64 kB y el Flyport Wi-Fi no posee este tipo de memoria, pero esto no influye en el proyecto, puesto que no se van a guardar datos en esa memoria.

<sup>3</sup> [http://space.openpicus.com/u/ftp/datasheet/datasheet\\_flyportpro\\_wifi.pdf](http://space.openpicus.com/u/ftp/datasheet/datasheet_flyportpro_wifi.pdf)



Ilustración 18: Flyport Wi-Fi.  
Fuente: [store.openpicus.com/](http://store.openpicus.com/)

El Flyport Wi-Fi cuenta con un microcontrolador PIC 24 de Microchip, además tiene 18 GPIO y 4 entradas analógicas, de las cuales se usarán 17 para conectar los diferentes componentes con el microprocesador. También se usará una UART para programar el microcontrolador y no tener que usar la plataforma de programación. A parte de todo esto, también posee un módulo Wi-Fi, cuyo modelo es MRF24WG0MA/MB, que permitirá controlar el dispositivo de forma inalámbrica.

Se incorpora también un pulsador de *reset* para poder resetear el microprocesador.

### 3.3. Alimentación

La alimentación de las granjas porcinas suele ser por paneles solares y baterías de 12 V, por lo que se alimentará al dispositivo a 12 V. Los componentes que forman el proyecto se alimentan a 5 V o 3.3 V, lo que implica que se use un LDO para bajar el nivel de tensión 12 V a 5V y a 3 V.

El LDO que se eligió al principio fue el NCP5504DTRKG [9] de ON Semiconductor, el cual es un regulador dual, cuya salida se puede ajustar de 1.25 V a 5 V y puede dar 250 mA en cada salida. Esto encaja perfectamente con los consumos de los componentes principales del proyecto como se puede apreciar en la Tabla 5.

En la Ilustración 20 se aprecia el esquemático de la alimentación. Se ha colocado un diodo para que no deje pasar la corriente en caso de que se conectaran los cables de alimentación al revés y así actuar a modo de protección. Las resistencias R5 y R6 son para ajustar la tensión de salida a 5 V, tal y como indica el datasheet, los condensadores se han colocado por recomendación del fabricante.

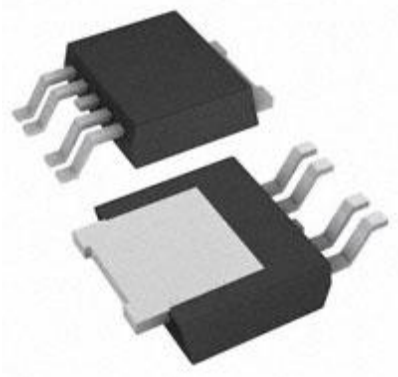


Ilustración 19: LDO NCP5504DTRKG



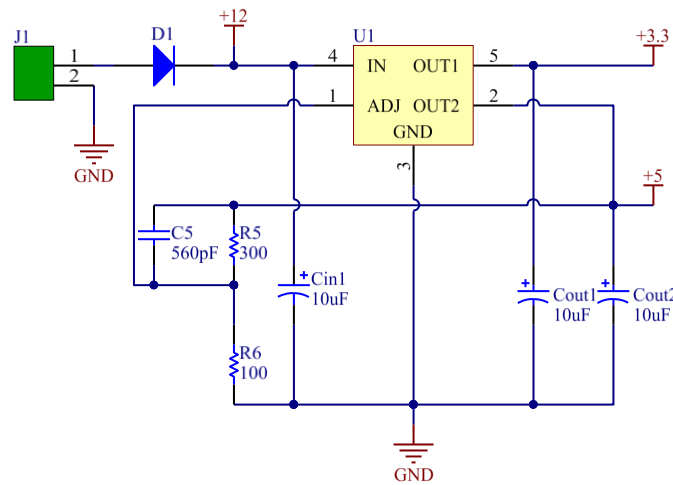


Ilustración 20: Esquemático alimentación

A continuación se muestran los consumos de los componentes y la tensión a la que van conectados cada uno de ellos

	Componente	Consumo			Tensión conectada
				Total	
Microcontrolador	FlyportPRO WI-FI	Sleep mode (Wi-Fi and microcontroller OFF)	0,18 mA	0,18 mA	3,3 V
		Hibernation (Wi-Fi OFF)	20 mA	20 mA	
		Wi-Fi ON and connected	150 mA	150 mA	
		Wi-Fi ON and not connected	35 mA	35 mA	
Actuadores	Relé	$I_{coil}$	30 mA	180 mA	12 V
	Controlador de puerta	$I_{ion}$	1,35 mA	8,1 mA	3,3 V
Sensores	Gases	$I_{heater}$	61 mA	122 mA	5 V
	Temp y Humedad	$I_{max}$	2,5 mA	7,5 mA	3,3 V
	Ultrasónico	$I_{on}$	15 mA	45 mA	5 V
	Anemómetro	$I_{max}$	10 mA	10 mA	12 V
Interfaz	Pantalla	$I_{on}$	1,8 mA	1,8 mA	5V
<b>Consumo a 3,3 V</b>				<b>166 mA</b>	
<b>Consumo a 5 V</b>				<b>169 mA</b>	
<b>Consumo a 12 V</b>				<b>190 mA</b>	

Tabla 5: Consumos y tensiones de los componentes




Debido a que se programa una versión de estudiante, como se ha comentado anteriormente, se optó por cambiar la alimentación ya que no se disponía del LDO NCP5504DTRKG en la universidad. Por tanto la alimentación que se diseñó fue con un LDO cuyo modelo es LM7805<sup>4</sup>, la ventaja de este regulador es que es *through hole*, por lo que no habrá problemas en el momento de soldar el componente, además puede proporcionar corrientes de hasta 1 A. Su uso es más sencillo, puesto que el fabricante únicamente recomienda colocar dos condensadores los cuales podrán suministrar corriente en caso de que se produzca algún pico.

El LM7805 proporciona 5 V de tensión, pero hay componentes que se alimentan a 3.3 V, esta tensión se obtendrá del Flyport Wi-Fi, que puede entregar hasta 100 mA, corriente suficiente, como se aprecia en la Tabla 6.

	Componente	Consumo			Tensión conectada
				Total	
Microcontrolador	Flyport WI-FI	Sleep mode	1,44 mA	1,44 mA	5 V
		Hibernation (Wi-Fi OFF)	20 mA	28,21 mA	
		Wi-Fi connected	150 mA	162,7 mA	
		Wi-Fi not connected	35 mA	39,75 mA	
Actuadores	Relé	$I_{coil}$	30 mA	60 mA	12 V
	Controlador de puerta	$I_{ion}$	1,35 mA	2,7 mA	3,3 V
Sensores	Gases	$I_{heater}$	61 mA	61 mA	5 V
	Temp y Humedad	$I_{max}$	2,5 mA	5 mA	3,3 V
	Ultrasónico	$I_{on}$	15 mA	15 mA	5 V
	Anemómetro	$I_{max}$	10 mA	10 mA	12 V
Interfaz	Pantalla	$I_{on}$	1,8 mA	1,8 mA	5V
Consumo a 3,3 V				8 mA	
Consumo a 5 V				240 mA	
Consumo a 12V				70 mA	

Tabla 6: Consumos alimentación modelo estudiante

<sup>4</sup> [http://www.farnell.com/datasheets/2161191.pdf?\\_ga=1.69096785.1796405222.1443540081](http://www.farnell.com/datasheets/2161191.pdf?_ga=1.69096785.1796405222.1443540081)

 <b>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

### 3.4. Diseño de PCB

En este apartado se va a describir el diseño de la PCB, así como las herramientas y los criterios que se han usado para diseñarla.

Para el diseño de la PCB se ha usado CircuitMaker<sup>5</sup>, este programa es una versión gratuita de Altium. Ofrece múltiples posibilidades a la hora de diseñar tanto el esquemático como la PCB, pues posee una comunidad de usuarios la cual sube sus diseños al servidor de Altium. Por tanto hay una gran variedad de componentes creados por los diferentes usuarios que puede usar cualquier persona con una cuenta de CircuitMaker.

Por otra parte, en lo que a las reglas de diseño se refiere, se ha usado una clase 3 ya que solamente incorpora componentes THL. Las pistas de alimentación se han colocado con una anchura de 1 mm, mientras que las de señal se han colocado con una anchura de 0.5 mm a excepción de las que pasan a través del conector del Flyport, que se han colocado con una anchura de 0.4 mm, para que no hubiera problemas a la hora de fabricar.

Los componentes se han colocado por bloques, intentado que los que se alimentan a 5 V estén en la parte superior de la PCB y los que se alimentan a 3.3 V en la parte inferior, dejando el centro libre para colocar el microcontrolador y los relés, puesto que ocupan gran parte de la PCB. También se ha intentado que los componentes estuvieran lo más alineados posible, pues si en un futuro se fabricara esta PCB de forma automática, el proceso sería más rápido. El primer prototipo de este proyecto se ha fabricado en el taller del Departamento de Electrónica y Comunicaciones de la Universidad de Zaragoza.

En esta versión de la PCB se ha dejado espacio para dos microcontroladores, puesto que como se ha comentado en numerosas ocasiones a lo largo de la memoria, se programa una versión estudiante del proyecto. Pero en una futura versión solamente habría que colocar FlyportPRO Wi-Fi, por lo tanto se eliminaría el exceso de vías que actualmente hay en la PCB haciendo que su fabricación fuese más rápida. También se ha dejado espacio para la alimentación con el FlyportPRO Wi-Fi, como se puede ver en las ilustraciones siguientes.

A continuación se muestra como ha quedado finalmente la PCB, así como su diseño en 3D (Ilustración 21), el cual es muy útil en lo que al posicionamiento y colocación de los componentes se refiere.

---

<sup>5</sup> [circuitmaker.com](http://circuitmaker.com)

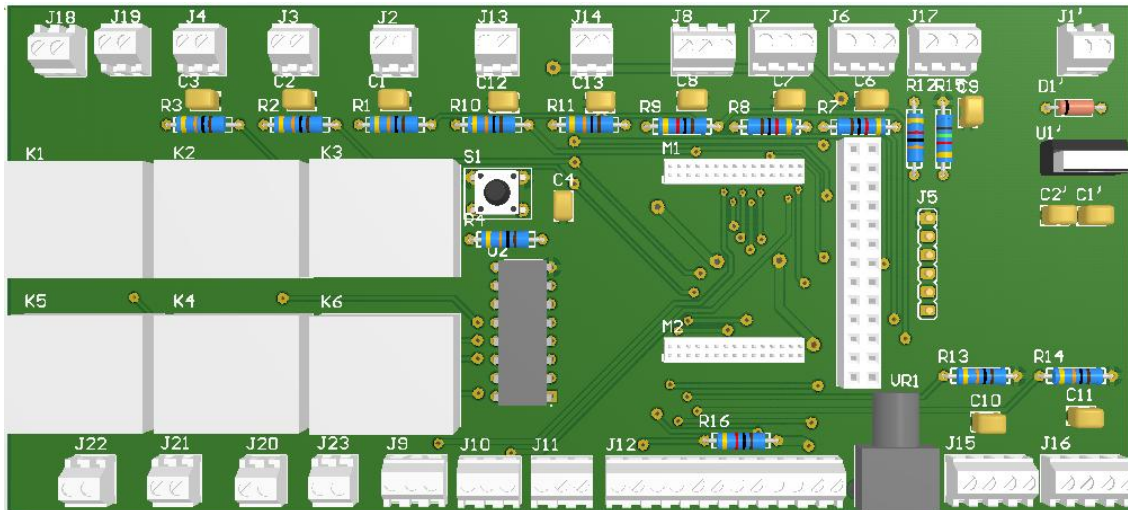


Ilustración 21: Diseño PCB en 3D

Finalmente, la PCB soldada ha quedado de la siguiente forma:

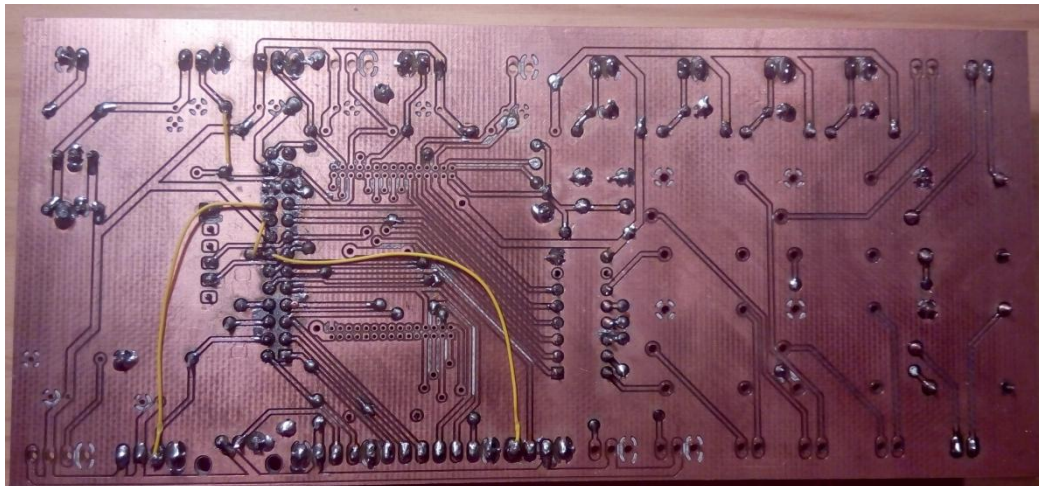


Ilustración 22: Cara *bottom* soldada

En la Ilustración 22 se muestra como ha quedado la cara *bottom* terminada de soldar. Se aprecian los errores que ha habido en el sensor de gases y en el ultrasónico, de estos errores se habla en la sección 5. En las futuras versiones de esta PCB habría que solucionar estos errores, el problema del sensor de gases solo afecta al Flyport Wi-Fi ya que en el FlyportPRO Wi-Fi este fallo está solucionado. Pero el fallo del ultrasónico es debido a una mala elección de la huella del mismo.

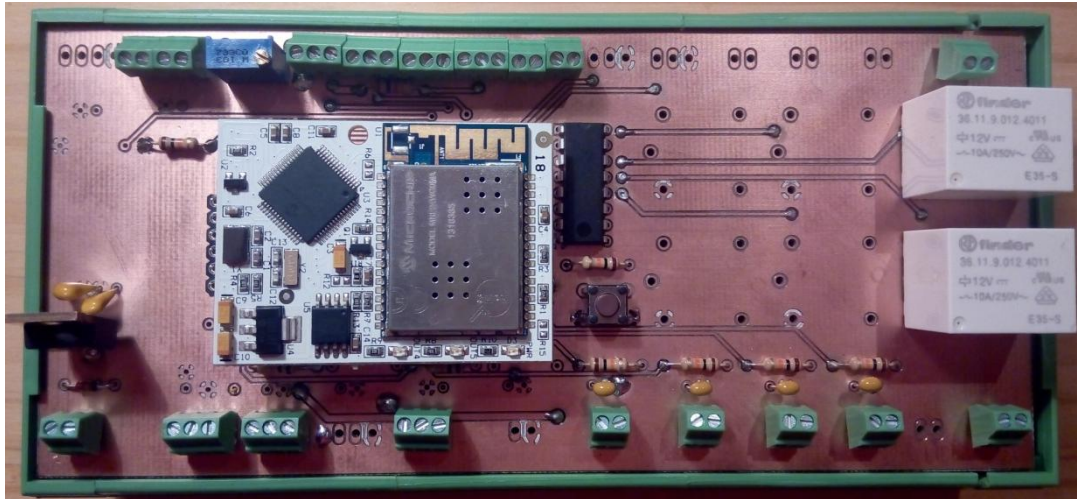



Ilustración 23: Cara *top* soldada

En la Ilustración 23 se ve como queda la PCB dentro de la caja en la que iría albergada, posteriormente esta caja habría que cubrirla. En esta cara se pueden apreciar algunas pistas horizontales que no se han podido realizar por la cara *bottom*, también se ven las vías y las huellas del resto de componentes que por falta de los mismos no se han programado. Por último se puede apreciar el espacio que queda alrededor de la alimentación, esto es debido a que cuando se coloque el FlyportPRO Wi-Fi la alimentación es diferente y ocupa más espacio, como se ha comentado en la sección 3.3.

 <b>Escuela de Ingeniería y Arquitectura</b> <b>Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

## 4. DISEÑO DEL FIRMWARE

### 4.1. Introducción

En este bloque se habla sobre el diseño del firmware. La programación se ha realizado de forma modular, en diferentes capas de programación para que se pueda actualizar de una forma más rápida y sencilla, como se puede apreciar en el Anexo III. Como se ha comentado en la sección 3.2.4 se usa el Flyport Wi-Fi para programar el dispositivo, este módulo se programa en lenguaje C.

Primero se tratará sobre cómo se ha estructurado el código de programación, posteriormente se hablará sobre los diferentes modos de los que se compone el sistema y por último se explicará la comunicación usada.

### 4.2. Estructura del código de programación

La programación del dispositivo se ha realizado de forma modular, en concreto se ha estructurado en 3 capas como se muestra en la Ilustración 24.

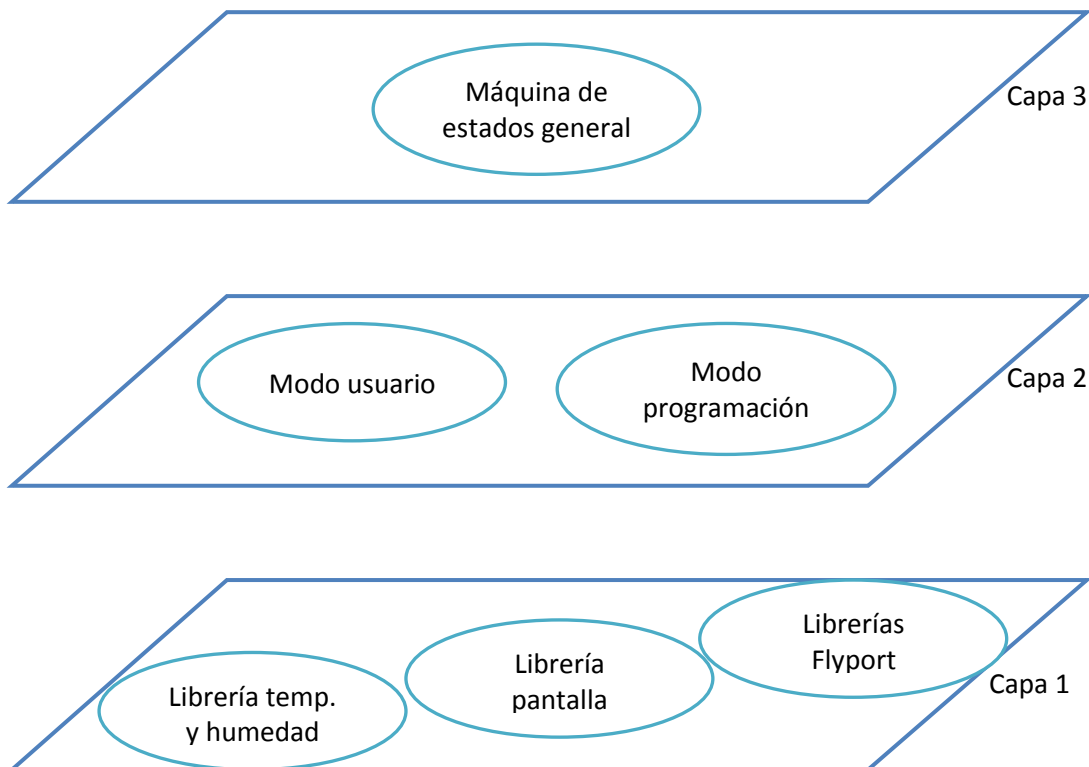



Ilustración 24: Capas del código de programación

 <b>Escuela de Ingeniería y Arquitectura</b> <b>Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

En la capa 1 se han definido las librerías que controlan los sensores más difíciles de programar, además de la librería que se necesita para poder programar en el Flyport. En la capa 2 se encuentran los dos modos posibles en los que puede estar el dispositivo, sobre estos modos se habla en la sección 4.3. Y para finalizar, en la capa 3, está la máquina de estados, la cual es la que controla todos los cambios de estado que se producen.

### 4.3. Funcionamiento del dispositivo

En la Ilustración 25 se puede apreciar la máquina de estados general, la cual está compuesta por 3 estados.

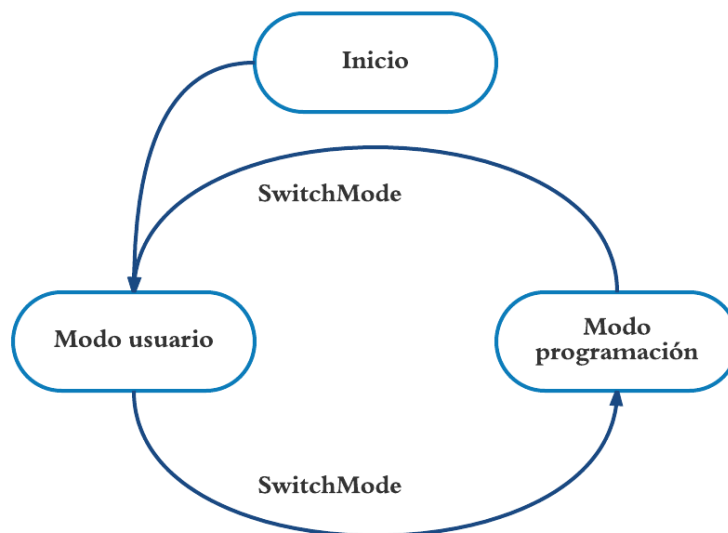


Ilustración 25: Máquina de estados general

Se comienza en el estado de inicio, donde se configuran los parámetros que necesita usar el sistema. Una vez realizada esta tarea se pasa al modo usuario, este es el modo en el que va a pasar la mayor parte del tiempo. Por último se entra en el modo de programación, en este modo se programan todos los máximos y mínimos de los diferentes sensores para proporcionar unas buenas condiciones ambientales a los animales. El dispositivo se podrá controlar de manera inalámbrica, a través de Wi-Fi, y también de manera presencial a través de la interfaz que proporciona.

A continuación se describen con detalle los diferentes modos de operación del dispositivo.



### 4.3.1. Inicio

El modo de inicio es el proceso principal, al cual se llega cada vez que se pone en marcha por primera vez el dispositivo o se reinicia. En este modo se crean las variables globales del dispositivo tales como la temperatura, la humedad, la velocidad del viento, etcétera, así mismo se configuran los pines de los diferentes sensores, como por ejemplo los pines del sensor de temperatura y humedad, también se configuran los pines de la pantalla y se configura la RTC. Todo esto se puede apreciar en el código, el cual está en el Anexo III.

Una vez que se ha inicializado todo correctamente se pasa de forma directa al modo de usuario el cual se detalla a continuación.

### 4.3.2. Modo usuario

Después de establecer todas las variables globales y de que se hayan inicializado todos los componentes necesarios, se pasa al modo usuario.

Con este modo se pretende que el usuario únicamente pueda visualizar la temperatura de confort de los animales, sin miedo a cambiar o tocar ningún otro parámetro del dispositivo. En este modo estará el dispositivo la mayor parte del tiempo.

En la Ilustración 26 se puede ver el diagrama de flujo correspondiente a este modo. El funcionamiento es el siguiente:

Se muestra por pantalla la temperatura y humedad del interior de la granja. A la misma vez se comprueba que todos los parámetros estén dentro de los límites marcados en el modo programación, en cual se explica en la sección 4.3.3, para ver si hay o no que actuar. Si algún parámetro está fuera de estos límites establecidos se actuará en consecuencia de la siguiente manera:

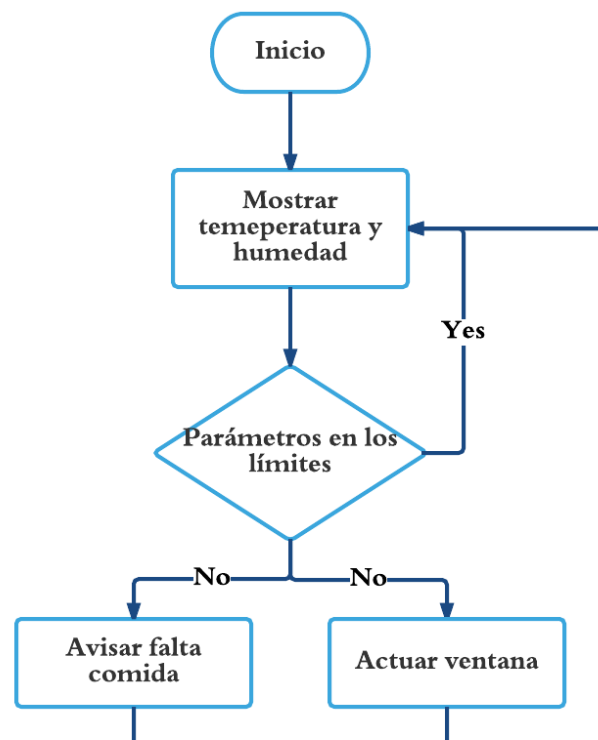



Ilustración 26: Diagrama de flujo modo usuario



 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

- Si la temperatura del interior de la granja es superior a la del ambiente y el usuario desea bajar la temperatura se abrirán las ventanas.
- Si la humedad del interior la granja supera los límites establecidos y la humedad del ambiente es inferior se abrirán las ventanas.
- Si la concentración de gases es superior a la permitida se abrirán las ventanas. En este caso se abrirán siempre las ventanas aunque haya alguna otra contradicción, ya que una alta concentración de gases es muy peligrosa.
- Cuando estos parámetros vuelvan a estar dentro de los límites se cerrarán las ventanas.
- Si la velocidad del viento es superior a la establecida se cerrarán las ventanas.
- Si la comida en el silo es inferior a la establecida se avisará al ganadero con un mensaje en la pantalla o por la página web.
- Si no queda comida en el último comedero se activará el motor para que se rellenen los comederos.

Una vez se ha establecido el modo de usuario se pasa a explicar el modo de programación.

### 4.3.3. Modo programación

Se entrará a este modo pulsando el botón *SwitchMode* una vez se haya entrado en el modo de usuario.

En este modo se van a poder cambiar todos los parámetros de confort de los animales, tal y como es la velocidad máxima del viento, la altura del pienso, etcétera. Se podrán cambiar todos los parámetros a excepción de la concentración máxima de gases debido a su peligro. Este modo se usará pocas veces, pues una vez que se programan los parámetros principales no hace falta volver a tocarlos y para cambiar la temperatura lo más cómodo para el ganadero es hacerlo a través de la página web.

En la Ilustración 27 se muestra el diagrama de flujo del modo programación, al entrar en este modo se pueden cambiar tres parámetros importantes del sistema. Se empieza cambiando la velocidad máxima del aire, cuando se supera este parámetro se cierran las ventanas, para poder subirla hay que apretar el botón *SwitchUp*, cada vez que lo apretemos se sube 1 m/s la velocidad, mientras que para bajar esta velocidad hay que pulsar *SwitchDown*, cada vez que pulsemos este pulsador disminuirá la velocidad en 1 m/s. Una vez se ha



cambiado la velocidad se puede cambiar la altura del pienso, si pulsamos el botón *SwitchUp* se aumentará la altura del pienso 10 cm, mientras que cada vez que pulsemos *SwitchDown* se disminuirá la altura del pienso 10 cm. Por último, se programa la temperatura de consigna de interior de la granja, para ello hay que apretar *SwitchUp* para subir la temperatura 1 °C y *SwitchDown* para disminuirla en 1 °C. Para pasar de cambiar un parámetro a cambiar otro parámetro se debe apretar el pulsador *SwitchMode*.

Una vez que se han programado todos los parámetros de este modo se vuelve de forma inmediata al modo usuario.

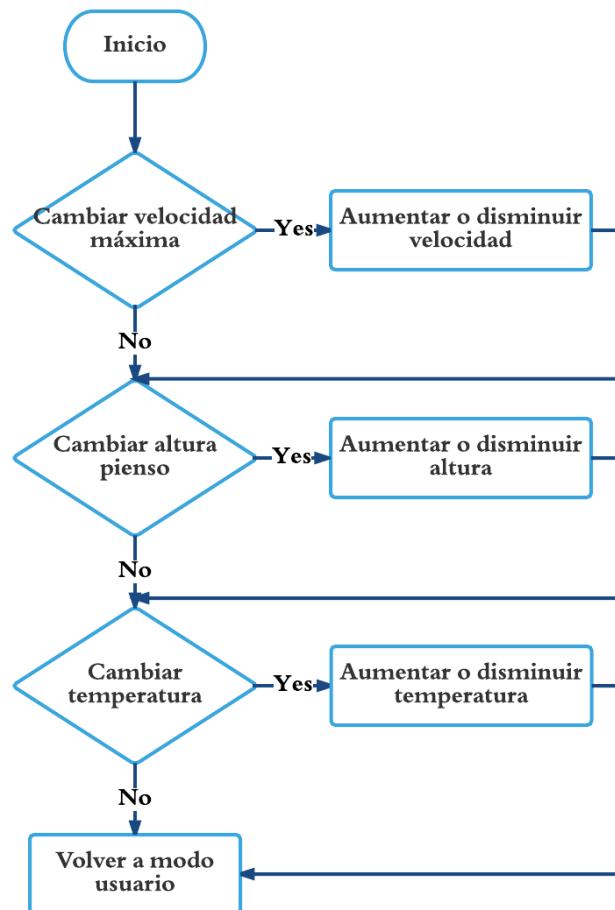



Ilustración 27: Diagrama de flujo modo programación


 <b>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

## 4.4. Wi-Fi

Para que el usuario se pueda comunicar con el dispositivo sin que este tenga que ir a la granja donde está situado el mismo, se ha pensado crear una red Wi-Fi en la que poder cambiar la temperatura de consigna del interior de la granja.

En esta versión del proyecto se va a crear una red Wi-Fi local a través del Flyport, esto se hace para que se hagan mucho más fáciles las pruebas del laboratorio. Pero si en algún momento este producto se colocara en alguna granja habría que crear una comunicación Wi-Fi a través del punto de Internet que hubiera en la granja.

El servidor web se ha realizando modificando el ejemplo que hay en la página de OpenPicus. Para ello se hace una página web en formato HTML, en la que se puede modificar la temperatura de consigna del interior de la granja y también se muestra la temperatura que hay en el interior de la misma. Además también nos muestra un aviso cuando la altura de la comida del silo es inferior a la seleccionada. Esto hace que se pueda controlar de la misma forma, la parte principal de la programación, desde el dispositivo colocado en la granja o desde la página web, lo que le da libertad al usuario y no tiene por que ir todos los días a modificar la temperatura o ver si no queda comida.

 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

## 5. CONCLUSIONES


El objetivo de este TFG se ha cumplido, ya que se ha desarrollado un prototipo funcional para que una granja porcina pueda ser automatizada. Aunque como se ha dicho en repetidas ocasiones a lo largo de la memoria, en la parte de programación se han usado menos sensores de los que se debieran debido a la falta de los mismos.

A continuación se expone el cumplimiento de los objetivos parciales que se marcaron al principio del proyecto


- Búsqueda de información: Se ha aprendido como es una granja porcina y los sensores y parámetros que hay que tener en dichas granjas para que los animales estén en buenas condiciones.
- Planteamiento de las especificaciones: Una vez con toda la información obtenida se plantearon las especificaciones, esto ha hecho que se sintetice correctamente la información y así poder plantear unas especificaciones acorde con la información que se había buscado.
- Diseño electrónico: Se ha conseguido realizar un diseño del hardware capaz de integrar todos componentes necesarios en una PCB del tamaño de la caja. En esta parte han aparecido algunos contratiempos, los cuales se han solucionado como se puede apreciar en la sección 3.4
- Diseño del firmware: Se ha diseñado el firmware, como se ha dicho en muchas ocasiones, para una versión de estudiante. Pero se ha creado un algoritmo que integra todos los sensores y actuadores para tener un control básico de los parámetros de confort de los animales, así como controlar la comida que queda en el silo.

Como resultado final se ha obtenido un prototipo funcional del dispositivo, aunque en posteriores versiones se deberían hacer una serie de medidas para mejorar en todo lo posible el dispositivo:

- Mejorar el sistema de pesado de la comida colocando células de carga, pues con los sensores ultrasónicos no se puede medir bien la cantidad de comida que queda ya que según el tipo de pienso que es varía su densidad.
- Mejorar la PCB poniendo bien la huella del sensor de ultrasonidos en la PCB y en el esquemático, así como cambiar el pin del sensor de gases de GPIO a uno analógico en el Flyport Wi-Fi.


 <p>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</p>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

- Mejorar la comunicación Wi-Fi creando una red con el punto de acceso a Internet de la granja.
- Mejorar la comunicación del dispositivo con el usuario, haciendo que se envíe un correo electrónico o similar al usuario cuando la comida del silo sea insuficiente.
- Mejorar el algoritmo haciendo un algoritmo más robusto que contemplase todos los sensores colocados en la PCB.

 <b>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

## 6. BIBLIOGRAFÍA

- [1] STMicroelectronics, «Datasheet CI Darlington,» [En línea]. Available: <http://www.mouser.com/ds/2/389/uln2001-957492.pdf>.
  
- [2] Adafruit, «Datasheet sensor DHT22,» [En línea]. Available: <http://www.mouser.com/ds/2/737/dht-932870.pdf>.
  
- [3] Figaro, «Datasheet sensor de gases,» [En línea]. Available: <http://es.rs-online.com/webdocs/065f/0900766b8065f325.pdf>.
  
- [4] U. Méndez, «Tipos de sensores de gas,» [En línea]. Available: <https://www.330ohms.com/blogs/blog/sensores-de-gas-como-funcionan>.
  
- [5] SparkFun, «Datsheet sensor ultrasónico,» [En línea]. Available: <http://www.mouser.com/ds/2/813/HCSR04-1022824.pdf>.
  
- [6] Datavision, «Información pantalla LCD,» [En línea]. Available: [http://www.datavision.com.tw/en/lcm\\_01\\_1.php?P\\_Id=10](http://www.datavision.com.tw/en/lcm_01_1.php?P_Id=10).
  
- [7] Flyport, «Datasheet Flyport Wi-Fi,» [En línea]. Available: [http://space.openpicus.com/u/ftp/datasheet/datasheet\\_flyport\\_wifi\\_g.pdf](http://space.openpicus.com/u/ftp/datasheet/datasheet_flyport_wifi_g.pdf).
  
- [8] ON Semiconductor, «Datasheet LDO Dual,» [En línea]. Available: <http://es.rs-online.com/webdocs/124f/0900766b8124f9e1.pdf>.
  
- [9] B. Martín del Brío y A. Bono Nuez, Transductores y Sistemas de Instrumentación.
  
- [10] ITG Ganadero, «Parámetros de confortabilidad ambiental en explotaciones porcinas,» [En línea]. Available: <http://www.itgganadero.com/itg/portal/documentos.asp?id=40>.

 <b>Escuela de Ingeniería y Arquitectura Universidad Zaragoza</b>	Automatización de una granja porcina	
	MEMORIA	Fecha revisión: 25/11/2016

[11] ITG Ganadero, «Instalaciones porcinas y bienestar,» [En línea]. Available: <http://www.itgganadero.com/itg/portal/documentos.asp?id=48>.



## ANEXO I

### Anexo I.I. Plano cara top

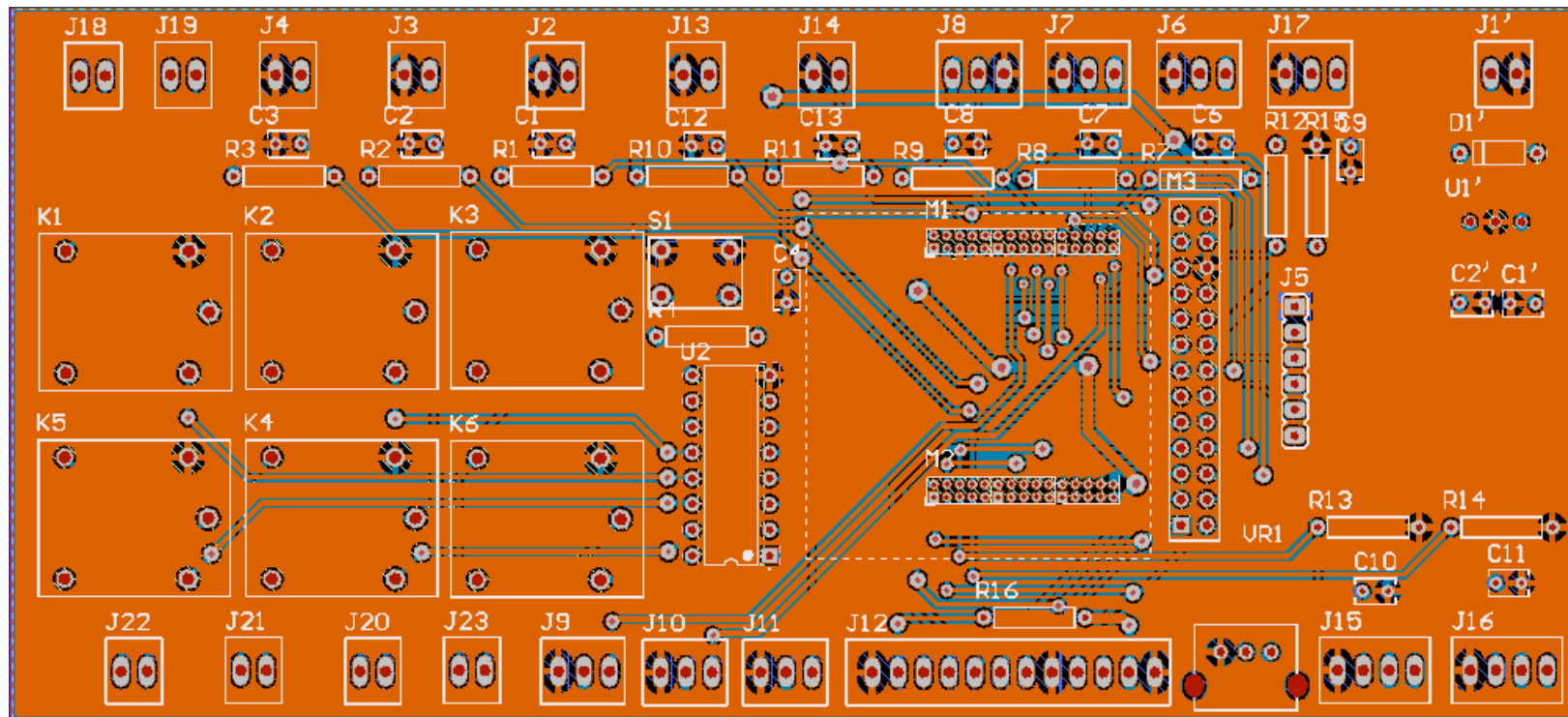


Ilustración 28: Plano cara top





## Anexo I.II. Plano cara bottom

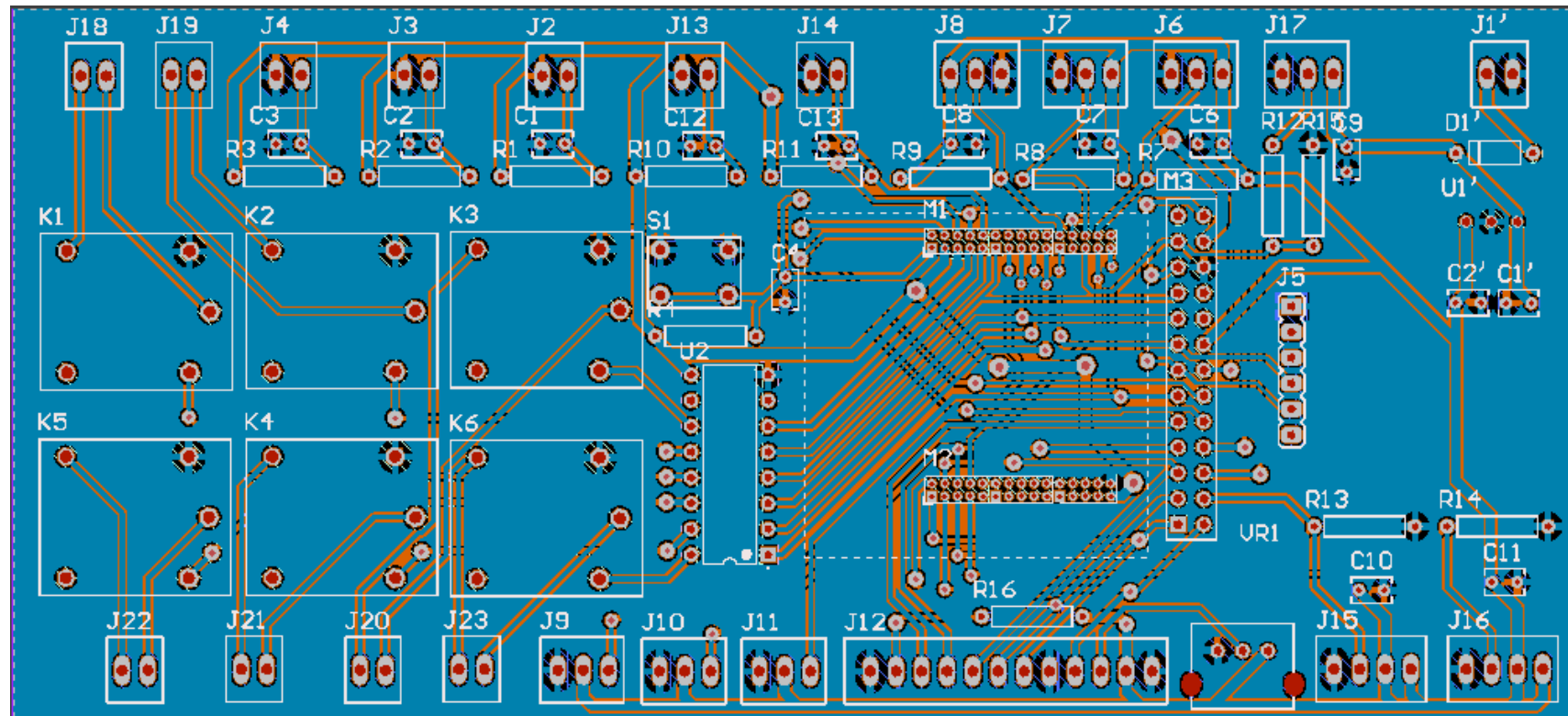

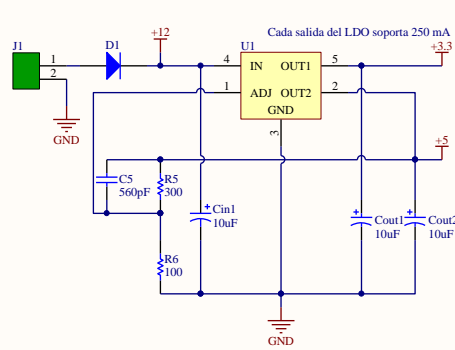


Ilustración 29: Plano cara bottom

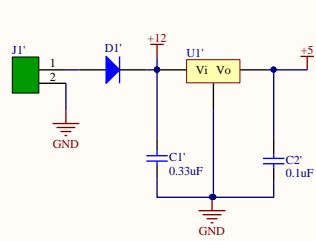
 <p>Escuela de Ingeniería y Arquitectura <b>Universidad Zaragoza</b></p>	Automatización de una granja porcina	
	ANEXOS	Fecha revisión: 25/11/2016

## Anexo I.III. Esquemático

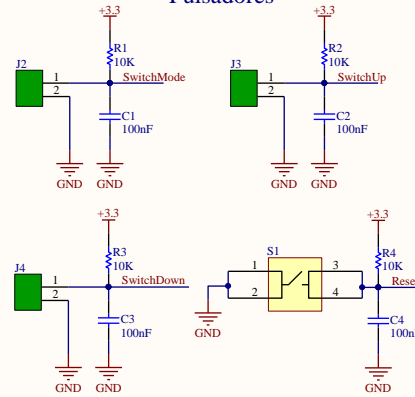
### Alimentación



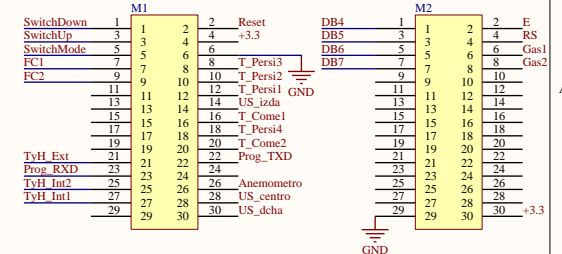
### Alimentación modelo prueba



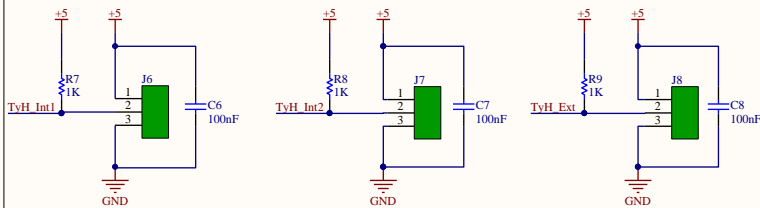
### Pulsadores



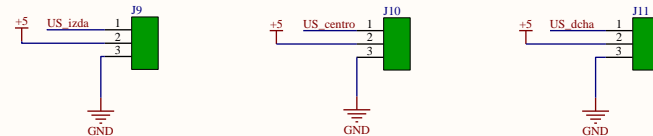
### Micro: FlyportPRO



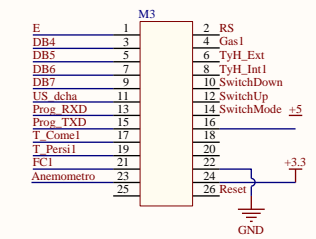
### Temp y Humedad



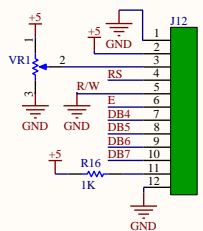
### Sensor Ultrasonico



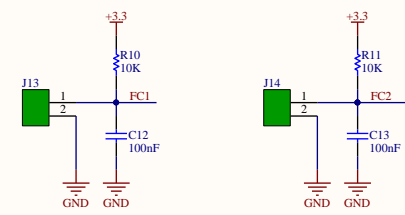
### Micro: Flyport



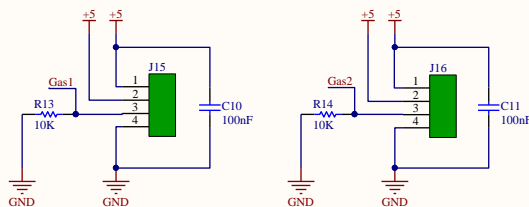
### Pantalla



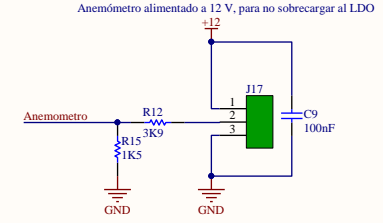
### Finales de carrera



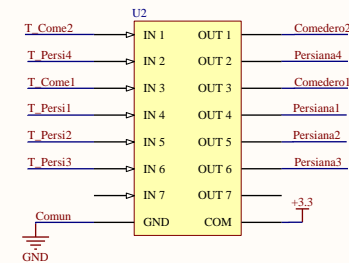
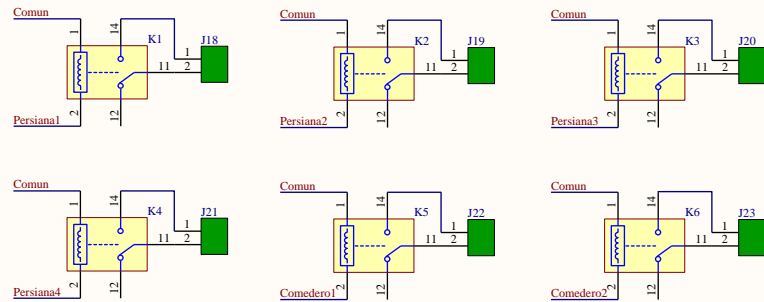
### Sensor Gases



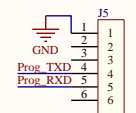
### Anemómetro



### Actuadores



### UART Programación



Title			Automatización de una granja porcina		
Size	Number			Revision	
A3	3.1			19/07/2016	
Date:	25/11/2016	Sheet of	1 to 1		
File:	Granja.SchDoc	Drawn By:	Estibalz Muñoz		



## ANEXO II. COSTE DE LOS COMPONENTES

Cantidad	Descripción	Precio					
		1 ud		100 ud		1000 ud	
		Unidad	Total	Unidad	Total	Unidad	Total
12	Condensador 0.1uF	0,235	2,82	0,055	0,66	0,045	0,54
1	Condensador 0.33F	0,299	0,299	0,119	0,119	0,072	0,072
1	Diodo	0,091	0,091	0,017	0,017	0,011	0,011
12	Conector 2 pin	0,489	5,868	0,421	5,052	0,413	4,956
1	Conector 6 pin	0,478	0,478	0,341	0,341	0,3	0,3
7	Conector 3 pin	1,18	8,26	0,929	6,503	0,762	5,334
1	Conector 12 pin	3,98	3,98	3,66	3,66	3,45	3,45
2	Conector 3 pin	2,11	4,22	1,28	2,56	1,09	2,18
6	Relés	1,35	8,1	1,1	6,6	0,989	5,934
2	Zócalo Flyport	2,54	5,08	2,04	4,08	1,48	2,96
8	Resistencia 10 K $\Omega$	0,109	0,872	0,109	0,872	0,019	0,152
4	Resistencia 1 K $\Omega$	0,109	0,436	0,032	0,128	0,02	0,08
4	Pulsador	0,119	0,476	0,092	0,368	0,066	0,264
1	LDO	1	1	0,623	0,623	0,434	0,434
1	C.I. relés	0,453	0,453	0,229	0,229	0,177	0,177
1	Potenciómetro 10 KW	0,759	0,759	0,472	0,472	0,342	0,342
1	Anemómetro	40,68	40,68	36,62	36,62	36,62	36,62
2	Final de carrera	6,52	13,04	4,2	8,4	4,2	8,4
3	Sensor ultrasónico	13,63	40,89	13,63	40,89	13,63	40,89
3	Sensor temp y hum	13,72	41,16	12,35	37,05	12,35	37,05
3	Sensor gas	13,8	41,4	13,8	41,4	13,8	41,4
1	Pantalla	12,02	12,02	12,02	12,02	12,02	12,02
	<b>Total PCB</b>		<b>45,68</b>		<b>33,636</b>		<b>28,114</b>
Opcionales	Anemómetro		40,68		36,62		36,62
	Final de carrera		13,04		8,4		8,4
	Sensor ultrasónico		40,89		40,89		40,89
	Sensor temp y hum		41,16		37,05		37,05
	Sensor gas		41,4		41,4		41,4
	Pantalla		12,02		12,02		12,02
	<b>Total proyecto</b>		<b>234,87</b>		<b>210,016</b>		<b>204,494</b>

Tabla 7: Coste de los componentes



## ANEXO III. CÓDIGO DE PROGRAMACIÓN

```
1. #include "taskFlyport.h"
2. #include "grovelib.h"
3. #include "OpenPicusLCD.h"
4. #include "rht03.h"
5.
6. // Variables globales
7. float TemInt = 0;
8. float TemExt = 20;
9. float HumInt = 0;
10. float HumExt = 60;
11. char temperatura[100];
12. char humedad[100];
13. char tempselec = 0;
14. char humselec = 0;
15. char alturaselec = 0;
16. char vientoselec = 0;
17. struct tm mytime;
18. int contUs = 0;
19. double altura = 0.0; //cm
20. char distancia[100];
21. int adcGas = 0;
22. double Rs = 0.0;
23. double Ro = 14437.92;
24. double Vout = 0.0;
25. double ratio = 0; //ratio entre rs/r0
26. int adcViento = 0;
27. double VoutViento = 0.0;
28. double VelViento = 0.0;
29. char Velocidad[100];
30.
31. // Flags interrupciones
32. char fsubir=0;
33. char fbajar=0;
34. char fcambiar=0;
35. char modo=0;
36. char fsensores=0;
37.
38. void Subir (){
39.     fsubir=1;
40. }
41.
42. void Bajar (){
```



```
43.         fbajar=1;
44.     }
45.
46.     void CambioModo (){
47.         fcambiar=1;
48.         modo++;
49.         if(modo ==4) modo=0;
50.     }
51.
52.     void leer_sensores(){
53.         fsensores=1;
54.     }
55.
56.     void ultrasonido(){
57.         IOPut(p14, off);
58.         vTaskDelay(1);
59.         IOPut(p14, on);
60.         Delay10us(2);
61.         IOPut(p14, off);
62.         while(!(IOGet(p11))); //Esperamos al escalon de respue
sta
63.         vTaskSuspendAll();
64.         while(IOGet(p11)){
65.             Delay10us(1);
66.             contUs=contUs+10;
67.         }
68.         xTaskResumeAll();
69.         altura=(contUs)*340.0*1.5/20000.0; // Cálculo de la al
tura
70.         contUs=0;
71.     }
72.
73.     void gases(){
74.         adcGas = ADCVal(3);
75.         Vout = adcGas*0.002; // Para convertirlo a voltaje
76.         Rs = ((5.0-Vout)/Vout)*10000.0;
77.         ratio = Rs/Ro; // rs/r0
78.     }
79.
80.     void viento(){
81.         adcViento = ADCVal(1);
82.         VoutViento = adcViento*0.002; // Para convertirlo a vo
ltaje
83.         VelViento = (VoutViento-0.4)*32/(2.0-
0.4); // Cálculo de la velocidad del viento
```



```
84.     }
85.
86.     void FlyportTask(){
87.         //Inicializacion interrupcio RTCC
88.         mytime.tm_hour = 0;
89.         mytime.tm_min = 0;
90.         mytime.tm_sec = 0;
91.         mytime.tm_mday = 0;
92.         mytime.tm_mon = 0;
93.         mytime.tm_year = 116;
94.         mytime.tm_wday = FRIDAY;
95.
96.         RTCCSet(&mytime); //Set time
97.
98.         RTCCAlarmConf(&mytime, REPEAT_INFINITE, EVERY_TEN_SEC, le
er_sensores); //set alarma
99.         RTCCAlarmSet(ON); //Enables alarma
100.
101.         // Inicializar ultrasónico
102.         IOInit(p11, indown); // Entrada ultrasonidos echo
103.         IOInit(p14, out); // Salida ultrasonidos trig
104.
105.         // Inicializar final de carrera
106.         IOInit(p21, inup);
107.
108.         // Inicializar reles
109.         IOInit(p17, out);
110.         IOInit(p19, out);
111.         // Inicializar pulsadores
112.         IOInit(p10, inup);
113.         IOInit(p12, inup);
114.         IOInit(p18, inup);
115.
116.         // Configurar interrupciones
117.         IOInit(p10, EXT_INT2); // SwitchDown
118.         IOInit(p12, EXT_INT3); // SwitchUp
119.         IOInit(p18, EXT_INT4); // SwitchMode
120.
121.         // Inicializar interrupciones
122.         INTInit(2, Subir , 1); //initialized the external interr
upt 2 and associate a function
123.         INTEnable(2);
124.         INTInit(3, Bajar , 1); //initialized the external interr
upt 2 and associate a function
125.         INTEnable(3);
```



```
126.      INTInit(4,CambioModo , 1);//initialized the external i
nterrupt 2 and associate a function
127.      INTEnable(4);
128.
129.      // Inicializar la pantalla (rs,rw,e,d0,d1,d2,d3,d4,d5,
d6,d7)
130.      init(p2, 255, p1, 0, 0, 0, 0, p3, p5, p7, p9);
131.
132.      // Dimensionar la pantalla (cols, filas, dotsize)
133.      begin(2, 16, 1);
134.
135.      // Configurar DHT22
136.      IOInit(p8, in);
137.      void *board = new(Board);
138.      void *rht03 = new(RHT03);
139.      attachToBoard(board,rht03,DIG1);
140.      configure(rht03, 3);
141.
142.      while(1){
143.
144.          if(modos==0){
145.              display(); // Encender la pantalla
146.              clear(); // Limpiar la pantalla
147.              TemInt = get(rht03, TEMP); //Leer temperatura
148.
              sprintf(temperatura,"Temp: %.1f", (double) Tem
Int);
149.              setCursor(0, 0); print(temperatura, 10);
150.              setCursor(10, 0); print(" grados", 10);
151.
152.              vTaskDelay(100);
153.              HumInt = get(rht03, HUMD); // Leer humedad
154.              sprintf(humedad,"Humedad: %.1f", (double) HumI
nt);
155.              setCursor(0, 1); print(humedad, 10);
156.              setCursor(13, 1); print(" % ", 10);
157.          }
158.
159.          if(modos==1){ // Seleccionar viento máximo
160.              clear(); // Limpiar la pantalla
161.              setCursor(0, 0); print("Sel. vel: ", 10);
162.              setCursor(9, 0); write(vientoselec);
163.              sprintf(Velocidad,"Velocidad: %.1f", (double)V
elViento);
164.              setCursor(0, 1); print(Velocidad, 10);
```





```
165.         setCursor(5, 1); print("m/s", 10);
166.
167.         if(fsubir){
168.             vientoselec = vientoselec + 1;
169.             fsubir=0;
170.         }
171.         if(fbajar){
172.             vientoselec = vientoselec - 1;
173.             fbajar=0;
174.         }
175.     }
176.
177.     if(modo==2){ // Seleccionar altura pienso
178.         clear(); // Limpiar la pantalla
179.         setCursor(0, 0); print("Sel. alt: ", 10);
180.         setCursor(9, 0); write(alturaselec);
181.         sprintf(distancia,"Altura: %.1f", (double)altu
ra);
182.         setCursor(0, 1); print(distancia, 10);
183.         setCursor(5, 1); print("cm", 10);
184.
185.         if(fsubir){
186.             alturaselec = alturaselec + 10;
187.             fsubir=0;
188.         }
189.         if(fbajar){
190.             alturaselec = alturaselec - 10;
191.             fbajar=0;
192.         }
193.
194.     }
195.
196.     if(modo==3){ // Seleccionar temperatura
197.         clear(); // Limpiar la pantalla
198.         setCursor(0, 0); print("Sel. temp: ", 10);
199.         setCursor(10, 0); write(tempselec);
200.         sprintf(temperatura,"Temp: %.1f", (double) Tem
Int);
201.         setCursor(0, 0); print(temperatura, 10);
202.         setCursor(10, 0); print(" grados", 10);
203.
204.         if(fsubir){
205.             tempselec = tempselec + 1;
206.             fsubir=0;
207.         }
```



```
208.         if(fbajar){
209.             tempselec = tempselec - 1;
210.             fbajar=0;
211.         }
212.
213.     }
214.
215.
216.     //Gestion de ventanas
217.     if(ratio < 0.7){ //Detectar gases peligro
218.         IOPut(p19, on); //abrir ventana
219.     }
220.     else
221.         if((tempselec > TemInt) && (TemInt < TemExt)){
222.             // Detectar temperatura seleccionada
223.             IOPut(p19 ,on); //abrir ventana
224.         }
225.         else
226.             if(vientoselec < VelViento){ // Detectar viento má
227.                 ximo
228.                 IOPut(p19, off); //cerrar ventana
229.             }else
230.                 if ((HumInt > humselec) && (HumInt > HumExt)){
231.                     // Detectar la humedad
232.                     IOPut(p19 ,on); //abrir ventana
233.                 }
234.                 else IOPut(p19,off); //cerrar ventana
235.
236.
237.     //Gestion del silo
238.     if(alturaselec < altura){ //Silo casi acabado
239.         clear();
240.         setCursor(0, 0); print("Aviso comida", 10);
241.     }
242.
243.     //Gestion comedero
244.     if(!IOGet(p21)) { //Comida acabada
245.         IOPut(p17,on); //Echar mas comida
246.     }
247.     else IOPut(p17,off); //Apagar comida
248.
249.
250.     //Actualizacion de valores sensores
251.     if(fsensores){
252.         //Ultrasonido
253.         ultrasonido();
254.     }
255. }
```



```
250.  
251.          //Gases  
252.          gases();  
253.  
254.          //Anemometro  
255.          viento();  
256.  
257.          fsensores=0;  
258.  
259.          }  
260.  
261.      }  
262.  }
```