



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

DISEÑO Y DESARROLLO DE UNA TRAMPA INTELIGENTE PARA EL CONTROL DE PLAGAS

Alumno: José Luis Fernández Collado

Tutor: Prof. D. Diego Manuel Martínez Gila
Dpto: Ingeniería Electrónica y Automática

Junio, 2017

Agradecimientos

En primer lugar, agradecer a mis padres ya que siempre han confiado en mí en todo momento y han estado luchando conmigo en los días más difíciles de la carrera.

Al resto de mi familia, que al igual que mis padres siempre se han preocupado por mí y por los valores que me han inculcado.

A Marta, por apoyarme durante esta toda la carrera y ayudarme cuando peor lo he pasado.

A mis amigos de siempre, gracias a ellos todos los años previos a mi entrada a la universidad hayan sido magníficos y me hayan influido a llegar hasta este punto.

Mis compañeros tanto de clase como de piso, en especial a Suca, Manu, Pascual y Francisco por todos los momentos buenos que hemos pasado juntos y sobre todo lo que me han ayudado durante todos los años de la carrera.

A mi profesor Diego Manuel Martínez Gila, porque ha hecho todo lo que ha estado en su mano para que yo pueda realizar mi trabajo de fin de grado.

Por último agradecer a Antonio Requena Ibáñez, ya que ha sido la persona que más me ha ayudado a lo largo de la carrera y sobre todo a la realización de este proyecto.

Índice

1. CONTEXTO	1
2. INTRODUCCIÓN AL PROYECTO	1
2.1. Mosca del olivo: Bactrocera (Dacus oleae)	2
2.1.1. Ciclo biológico	2
2.1.2. Desarrollo de la plaga	3
2.1.3. Daños	4
2.1.4. Seguimiento periódico de la población Bactrocera oleae	4
2.1.5. Medios de lucha	6
2.2. Objetivos del proyecto	6
3. METODOLOGÍA A DESARROLLAR	7
3.1. Pruebas de sensores	7
3.1.1. Sensor IR FC-51	8
3.1.2. Sensor emisor y receptor infrarrojos	12
3.2. Diseño y construcción del hardware	16
3.2.1. Diseño del primer prototipo	16
3.2.2. Diseño del segundo prototipo	18
3.2.2.1. Diseño en 3D de la placa base	23
3.3. Diseño e implementación del software para el microcontrolador	27
3.3.1. Arduino	27
3.4. Desarrollo de plataforma web	29
3.5. Implementación del software de comunicación con la plataforma web	31
3.5.1. Internet de las cosas	31
3.5.2. GSM	32
3.5.3. GPRS	34
3.5.4. GSM Arduino shield 2	34
3.5.4.1. Interconexion entre Arduino micro y shield gsm2	35
3.6. Flujograma del código utilizado	36
3.6.1. Programa principal	37
3.6.2. Código para resetear la memoria interna del microcontrolador	38
3.7. Autonomía de la trampa inteligente	39
4. RESULTADOS	42
4.1. Porcentaje de acierto del sensor receptor y emisor de infrarrojos	42
4.2. Pruebas de consumo	43
4.3. Montaje final	43
5. PRESUPUESTO DEL PROTOTIPO	45

5.1.	Estimación de costes	45
5.2.	Precios unitarios del producto	47
5.3.	Precio total del proyecto	48
6.	CONCLUSIONES Y TRABAJOS FUTUROS.....	48
6.1.	Posible diseño futuro.....	48
6.2.	Incorporación de sensores	52
8.	Bibliografía	54

INDICE DE ILUSTRACIONES

Ilustración 2.1. Larva de Bactrocera (Dacus oleae)	2
Ilustración 2.2. Bactrocera (Dacus oleae) adulta.	3
Ecuación 2.1. Mosca por Trampa y Día.....	5
Tabla 2.1. Ejemplo de capturas	5
Ilustración 3.1. Sensor detección y presencia de movimiento IR FC-51	8
Ilustración 3.2. Circuito electrónico del sensor FC-51	9
Ilustración 3.3. Tubo de cartón	10
Ilustración 3.4. Tubo translucido	10
Ilustración 3.5. Tubo negro	11
Ilustración 3.6. Emisor de infrarrojos	12
Ilustración 3.7. Receptor de infrarrojos	13
Ilustración 3.8. Transistor 2N2222	13
Ilustración 3.9. Potenciómetro lineal	14
Ilustración 3.10. Resistencias 1K Ω y 10K Ω	14
Ilustración 3.11. Sensor y receptor de infrarrojos.....	15
Ilustración 3.12. Pieza contenedora de componentes electrónicos.....	16
Ilustración 3.13. Tapa superior	17
Ilustración 3.14. Pieza contenedora de equipo electrónico	19
Ilustración 3.15. Pieza de estrechamiento	20
Ilustración 3.16. Pieza de almacenamiento de insectos.....	21
Ilustración 3.17. Tapa superior	22
Ilustración 3.18. Imagen exterior trampa inteligente	22
Ilustración 3.19. Corte transversal trampa inteligente	23
Ilustración 3.20. Circuito diseñado con proteus	24
Ilustración 3.21. Diseño de placa electrónica.....	25
Ilustración 3.22. Placa electrónica en 3D.....	26
Ilustración 3.23. Arduino micro	28
Ilustración 3.24. Página oficial ThingSpeak	30
Ilustración 3.25. Configuración canal creado	30
Ilustración 3.26. Internet de las cosas	31
Ilustración 3.27. GSM.....	33
Ilustración 3.28. Shield GSM 2	35
Ilustración 3.29. Conexión entre Arduino micro y shield gsm 2.....	36
Ilustración 3.30. Flujograma código principal.....	38
Ilustración 3.31. Divisor de tensión	39
Ecuación 3.1. Medida del divisor de tensión.....	40
Ecuación 3.2. Factor de conversión.....	40
Ilustración 3.32. Gráfica descarga de batería ThingSpeak	40
Ilustración 3.33. Flujograma para calcular descarga de la batería	41
Ecuación 4.1. Ecuación porcentaje de acierto	42
Ilustración 4.1. Componentes de trampa inteligente	44
Ilustración 4.2. Prueba con mosca de la fruta.....	44
Ilustración 4.3. Implementación final en olivo	45
Tabla 5.1. Costes indirectos	46
Tabla 5.2. Presupuesto del producto	48
Ilustración 6.1. Diseño futuro mosquero entrada y salida	49

Ilustración 6.2. Diseño futuro placa electrónica entrada y salida.....	50
Ecuación 6.1. Cálculo de potencia de placa fotovoltaica	51
Ilustración 6.3. Placa solar para el diseño futuro	51
Ilustración 6.4. Acelerómetro de 3 ejes ADXL 345.....	52
Ilustración 6.5. Módulo GPS Ublox NEO-6M	53

Resumen del proyecto

Este proyecto se ha escogido con el motivo de poder controlar plagas de la mosca *Bactrocera* (*Dacus oleae*) en la zona del interior, donde se producen los ataques de forma accidental y hay que llevar un control para empezar a intervenir realizando tratamientos. Con este proyecto se busca la solución para controlar estas plagas de una forma inteligente, ya que ahora se realiza de una forma manual. Este sistema estará colocado por distintas zonas para hacer un control por áreas.

El sistema de control de plagas inteligente, también está pensado para que pueda funcionar para controlar otros tipos de plagas de otros insectos que afecten a otros árboles o plantas modificando piezas del hardware y cambiando el código de programación del software, aunque este proyecto se centrara solo en el control de la mosca del olivo *Bactrocera* (*Dacus oleae*) en la zona del interior de la península.

1. CONTEXTO

Para hablar de la motivación de este proyecto, el lector de la memoria debe situarse en mi contexto. Soy un alumno de último curso del grado de ingeniería electrónica industrial, interesado, generalmente las tecnologías emergentes y en los nuevos equipo electrónicos.

Uno de los motivos de este proyecto, es completar mi formación universitaria, poniendo en práctica los conocimientos obtenidos durante todos los años que he estado en la carrera. Otro de los motivos, es enfrentarme a un problema real que me pueda encontrar fuera y poder utilizar estos conocimientos para poder resolverlo.

Por estos motivos, he aceptado la propuesta de mi tutor y del grupo de investigación GRAV (Grupo de Robótica, Automática y Visión por Computador) para diseñar una trampa inteligente a través de un microcontrolador, en mi caso el de Arduino, para el control de plagas del olivo, pero en el que se podría implementar para controlar las plagas de otras plantas.

2. INTRODUCCIÓN AL PROYECTO

Este Trabajo Fin de Grado (TFG) se ha escogido por el motivo de seguir poniendo en práctica todos los conocimientos que se han adquirido a lo largo del grado en Ingeniería Electrónica Industrial. Este proyecto se ha realizado en el Grupo de Robótica, Automática y Visión por Computador (GRAV) que se encuentra en la Universidad de Jaén.

El Trabajo Fin de Grado consiste en el diseño y desarrollo de una trampa inteligente para controlar insectos de tamaño reducido que distribuida en diferentes áreas de una plantación, permita monitorizar de forma online la magnitud de la infestación para así poder aplicar productos fitosanitarios. La trampa incorpora los sensores necesarios para cuantificar la cantidad de insectos que acceden a esta trampa, la cual estará programada con un software que permite que esta información sea mandada a una plataforma web para su monitorización remota.

2.1. Mosca del olivo: *Bactrocera (Dacus oleae)*

Es un insecto de la familia de los Tripétidos que se encuentra en el área mediterránea, en el occidente de Asia y en muchas zonas de África. Se trata de una mosca que en estado adulto suele medir de 4 a 5 milímetros de longitud. La hembra acaba su abdomen con el aparato ovipositor, de forma cónica y de un milímetro de longitud, pudiéndosela distinguir fácilmente del macho.

Los huevos son de color blanco lechoso y de longitud inferior a un milímetro, las larvas no tienen patas y su tamaño es de 6 a 8 milímetros de longitud por 1,3 a 1,4 milímetros de anchura en pleno desarrollo.

El control de plagas de este proyecto estará centrado en la hembra, ya que es quien causa el mayor daño al olivo como ya se explicará más adelante [1].

2.1.1. Ciclo biológico

Pasa el invierno en estado de letargo, bajo la tierra y con menos frecuencia en estado adulto en sitios resguardados. En primavera (marzo-abril) aparecen los adultos e inician un período en el que se alimentan de sustancias azucaradas y nitrogenadas, necesarias para el desarrollo de los genitales.

La puesta de huevos de las hembras se realiza en las aceitunas del olivo, tres días después de poner los huevos, la hembra elige los frutos de manera que tengan 8 o 10 milímetros de diámetro y que no hayan sido picados anteriormente.



Ilustración 2.1. Larva de *Bactrocera (Dacus oleae)*

En las últimas generaciones la hembra selecciona incluso frutos cuyo estado de madurez permita sincronizar su desarrollo con la evolución de la larva. Normalmente tiene 3 generaciones al año, aunque pueden llegar a tener 4 generaciones en circunstancias muy favorables. Las generaciones estivales completan su ciclo en 35 o 40 días, llegando hasta 60 días en las generaciones otoñales [1].



Ilustración 2.2. *Bactrocera (Dacus oleae)* adulta.

2.1.2. Desarrollo de la plaga

Los factores climáticos como son la temperatura y la humedad, influyen de forma decisiva en el desarrollo de la plaga, de manera que se limitan su área geográfica, regulando incluso su amplitud del ciclo biológico. En España, en la zona del litoral se da la mayor intensidad de plaga de este insecto ya que hay una elevada humedad relativa.

En el interior, los ataques de esta mosca son por accidente y sólo se produce cuando las condiciones climáticas de la zona se vuelven favorables para este insecto. Las altas temperaturas y la baja humedad relativa que hay en verano hacen imposible el desarrollo de este insecto ya que las larvas recién nacidas se desecan. La entonces no prospera hasta que se producen las primeras lluvias en otoño. En el

interior de la Península esta plaga no prospera por las condiciones climatológicas no son favorables.

Además de las condiciones climáticas citadas, los depredadores de la mosca también son importantes para que la plaga no se desarrolle. En verano la acción parasitaria de los depredadores de la mosca es muy fuerte, esto hace que estén controladas y no haya que actuar sobre ellas. En otoño, esta acción disminuye debido a que los depredadores en esta época se alimentan de otros insectos a parte de la larva de la mosca, por lo que ahora si se desarrolla la plaga y habría que tomar medidas [1].

2.1.3. Daños

La *Bactrocera* (*Dacus oleae*) se considera una de las plagas más importantes y temibles del olivo. La mosca adulta pone sus huevos en el interior de la aceituna, y la larva que nace, crece alimentándose del mesocarpio del fruto, provocando en la aceituna una disminución de peso del 20%, reduciendo el rendimiento de esta y la calidad del aceite considerablemente. Los frutos atacados en el peor de los casos, a la medida que van madurando pueden caer al suelo en función del daño recibido por la larva.

La disminución del peso que conlleva a que se reduzca el rendimiento de esta y la propia caída de la aceituna son los daños directos que produce la larva de *Bactrocera* (*Dacus oleae*) al olivo. Pero lo más importante es el daño indirecto que produce en la calidad del aceite de la aceituna atacada.

Otro daño que provoca esta larva es que en su desarrollo, deja agujeros y galerías en la aceituna, por lo que facilita la penetración de hongos (*Gloeosporium olivarum*) y bacterias que alteran gravemente la calidad de los aceites a causa del aumento de la acidez y el deterioro de las características organolépticas [1].

2.1.4. Seguimiento periódico de la población *Bactrocera oleae*

El control de los niveles de poblaciones se realiza mediante mosqueros de cristal (tipo McPhail) y de plástico. En este caso, el mosquero utilizado para este proyecto será de plástico. Dentro del mosquero se introduce una disolución de fosfato biamónico al 4% de proteína hidrolizada al 9% y 3% de bórax debido a su

bajo costo y mayor perdurabilidad en condiciones de campo, o una combinación en estado sólido de dos atrayentes, uno alimenticio y otro de feromona de la hembra de *Bactrocera* (*Dacus oleae*) para la atracción de hembras y machos que es el utilizado en este proyecto. La trampa se coloca en el interior del olivo con orientación sur y a media altura. En épocas de lluvia, además deberá colocarse en el exterior del árbol placas trampa amarillas con atrayente sexual.

Estas trampas se instalarán después del cuajado y antes del endurecimiento del hueso. Esto suele ocurrir a principios de junio.

Cuando se aproxima la época de presencia de la mosca, el conteo de las capturas debe realizarse semanalmente. El valor de las capturas se expresa como MTD (Mosca por Trampa y Día) y se calcula mediante la siguiente expresión:

$$MTD = \frac{N^{\circ} \text{ de adultos capturados}}{N^{\circ} \text{ de trampas} \cdot N^{\circ} \text{ días transcurridos}}$$

Ecuación 2.1. Mosca por Trampa y Día

Un ejemplo de las capturas que se han realizado en una semana sería el siguiente:

Trampa	Nº machos	Nº hembras	Total moscas
1	1	2	3
2	0	2	2
3	2	4	6
4	0	1	1
5	0	0	0
6	1	3	4
Total	4	12	16

Tabla 2.1. Ejemplo de capturas

Por lo tanto, el valor medio de capturas corresponde a:

$$MTD = \frac{16}{6 \cdot 7} = 0,38 \text{ moscas/día y trampa}$$

Cuando los valores de captura llegan a 0,6 mosca/día y trampa, es necesaria la colocación de las trampas para la captura masiva [2].

2.1.5. Medios de lucha

La forma de combatir este insecto es diferente según la zona en la que se encuentre. En el litoral mediterráneo se considera que es receptivo al ataque de mosca, mientras que más al interior donde los ataques son accidentales, habrá que determinar el nivel de población para iniciar el tratamiento. En esta zona es donde verdaderamente se muestra la eficacia de la trampa inteligente que se desarrollará más adelante.

El tratamiento puede hacerse con cebos aplicados en los árboles mojando la superficie de 1 a 2 m² en la parte orientada al sur con una solución de 600 cc de dimetoato, 1 Kg de proteína hidrolizable y 100 l de agua.

También se pueden hacer tratamientos en pulverización total y en este caso los insecticidas se utilizarán a dosis normales. También se recomienda el empleo de formation, troclorfon, metidation y formet. Se pueden realizar tratamientos aéreos aplicando dosis de 20 l por hectárea total de una disolución compuesta por 0,5 l de dimetoato, 0,5 kg de proteína hidrolizable y 20 l de agua [1].

2.2. Objetivos del proyecto

El objetivo general del proyecto es el diseño de una trampa inteligente para el cuantificar la cantidad de insectos de la mosca del olivo, de tamaño reducido que hay presentes en la superficie de una plantación, para poder realizar un control del nivel de plaga y usar el tratamiento contra esta plaga.

En este contexto, los objetivos específicos que se plantean dentro de este proyecto son:

- Seleccionar los sensores adecuados que permitan cuantificar estos insectos de tamaño reducido.
- Realizar un diseño en 3D del hardware de la trampa inteligente que irá conectada al mosquero de plástico adquirido anteriormente.
- Diseñar el software que permita registrar toda la información adquirida de los sensores conectados previamente en el mosquero y mandar toda

esta información a una plataforma web, para que en todo momento se pueda visualizar la cuantificación de los insectos.

- Validar todo el montaje de la trampa inteligente en el campo.

3. METODOLOGÍA A DESARROLLAR

La metodología que se desarrollará en este proyecto será:

- Realizar un diseño en 3D del hardware que se imprimirá en una impresora 3D y que irá atornillado al mosquero de plástico. Este diseño se ha hecho para realizar las pruebas convenientes con diferentes sensores y una vez comprobado su correcto funcionamiento, se diseñara una segunda pieza para el producto final.
- Probar diferentes sensores de detección de movimiento, para ver cuál funciona mejor a la hora de detectar insectos muy pequeños. Verificar que los sensores que se han puesto tienen un precio bajo y además tienen una buena precisión. En este proyecto se ha optado por la construcción de estos sensores, como se explicará más adelante.
- Diseñar el software a través de un microcontrolador, que sea el que gestione la cuantificación de los insectos y mande la información a la plataforma web, previamente creada. En este proyecto el software se ha realizado con el software libre Arduino.
- Una vez que todo el montaje esté listo y la plataforma web en funcionamiento, se pondrá el mosquero completo en el campo y se probará que todo funciona correctamente.

3.1. Pruebas de sensores

En primer lugar, se ha de probar el funcionamiento de sensores para detectar la presencia de la mosca del olivo en la trampa inteligente. Para esto se ha probado en primer lugar con un sensor digital para Arduino con diferentes tubos que imitarían la trampa inteligente.

3.1.1. Sensor IR FC-51

El sensor utilizado para la primera prueba fue el sensor de detección y presencia de movimiento, módulo sensor IR FC-51.

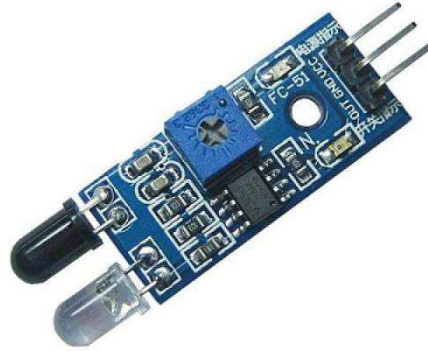


Ilustración 3.1. Sensor detección y presencia de movimiento IR FC-51

Este sensor es un sensor de proximidad por **infrarrojos IR** para detectar la presencia de movimiento o evitar obstáculos. Está compuesto por un transmisor que emite energía de **infrarrojos IR** y un receptor que detecta la energía **IR** reflejada por la presencia de cualquier objeto u obstáculo en la parte frontal del módulo. El módulo tiene un potenciómetro que permite ajustar el rango de detección. El sensor tiene una respuesta muy buena y estable con luz ambiente o en completa oscuridad.

Este sensor tiene un ángulo de cobertura de 35°, un voltaje de funcionamiento entre 3V y 6V, por lo que se puede alimentar con los 5V de tensión del Arduino. Tiene 3 pines, una para la alimentación del sensor, otra para tierra que se la da el propio Arduino y otra que se conecta a una entrada digital del Arduino para mandarle la información que este sensor recoja. Su rango de detección es desde 2cm hasta 30 cm. El nivel de salida que da este sensor cuando se detecta un obstáculo es de nivel lógico bajo y cuando no detecta ningún obstáculo dará salidas de nivel lógico alto [3].

El circuito electrónico de este sensor sería el siguiente:

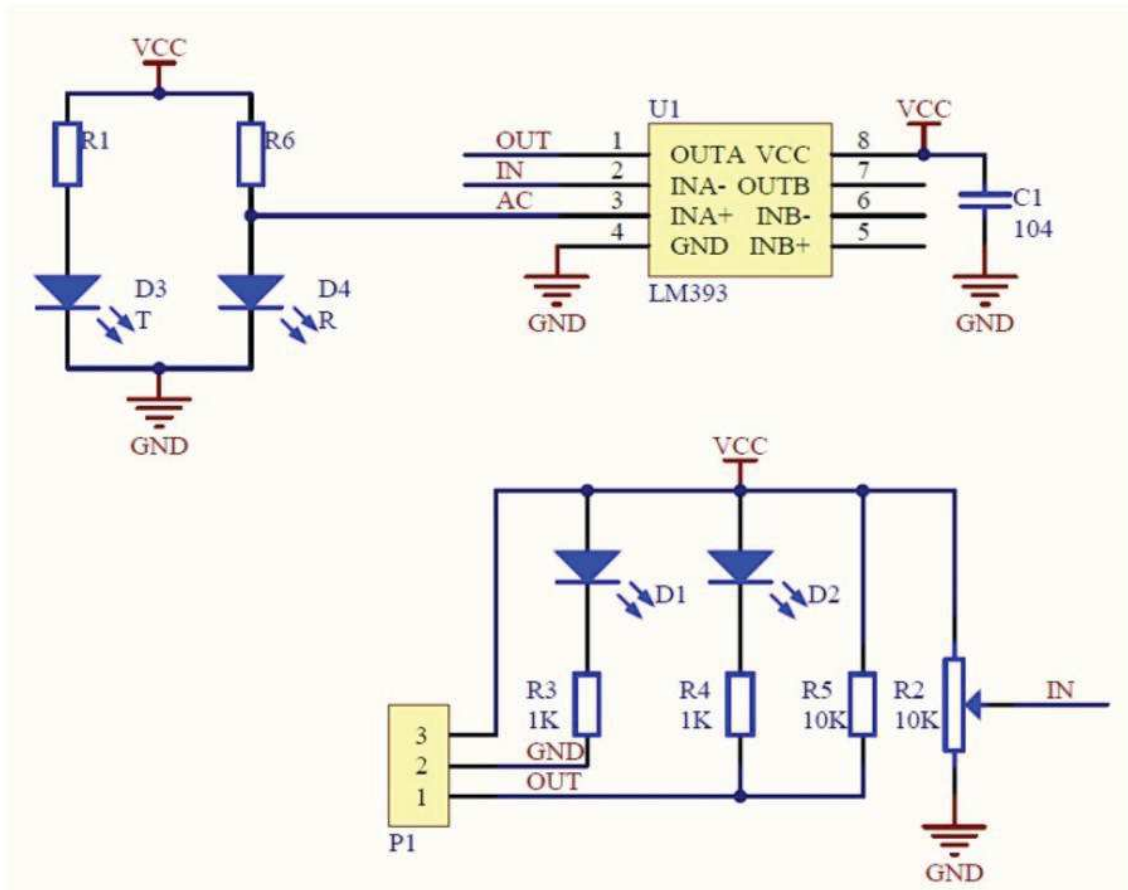


Ilustración 3.2. Circuito electrónico del sensor FC-51

Este sensor se ha probado con diferentes prototipos de tubos que serían similares a la entrada de la trampa inteligente con los siguientes resultados:

- **Tubo de cartón:**

En este tubo, el sensor FC-51 es capaz de detectar objetos de gran tamaño como por ejemplo el dedo de una persona cuando se pone delante del emisor y receptor. Tiene un funcionamiento correcto para detectar objetos grandes, pero para detectar la presencia de la mosca del olivo no sería suficiente, ya que los objetos pequeños pasan sin que el sensor sea capaz de detectarlos.



Ilustración 3.3. Tubo de cartón

- **Tubo de plástico translucido:**

En este tubo, el sensor FC-51 detectaba con dificultad los objetos grandes y no detectaba los objetos pequeños, ya que al ser este tubo translucido, la luz que le llega del exterior afecta a este sensor y pierde calidad a la hora de detectar.



Ilustración 3.4. Tubo translucido

- **Tubo de PVC negro:**

Una vez probados los dos tubos anteriores, se llega a la conclusión de que el color del tubo afecta a la precisión de este sensor. El sensor con el color negro siempre da una salida de nivel lógico alto, entonces en el momento que pase un objeto que tenga algo de color, lo detectará y dará una salida de nivel lógico bajo.

Este tubo, ha dado buenos resultados en la detección de objetos grandes que es capaz de detectarlos todos y con los objetos pequeños, los detecta en algunas ocasiones.



Ilustración 3.5. Tubo negro

Por este motivo el diseño que se hará del hardware de la trampa inteligente será de color negra, ya que en esa superficie es donde mejor trabaja el sensor.

Como este sensor no es muy preciso, ya que no tiene una alta probabilidad de acierto a la hora de detectar objetos muy pequeños como es la mosca de olivo se ha optado por realizar un circuito similar al que incorpora en el interior el FC-51, pero con un emisor frente a un receptor que se explicará en el siguiente punto.

3.1.2. Sensor emisor y receptor infrarrojos

El sensor diseñado se basa en un circuito en el que hay dos partes, un emisor y un receptor de infrarrojos. Utiliza básicamente la misma tecnología que el sensor nombrado anteriormente, a diferencia de que estos sensores se encuentran enfrentados uno a otro y no juntos como el anterior sensor en el que el sensor emisor y el receptor se encuentra uno pegado a otro. Esto hace que este sensor tenga una mayor precisión a la hora de detectar objetos pequeños como sería la mosca del olivo.

Este nuevo circuito constaría de:

- **Emisor de infrarrojos:** Es un componente que emite radiación infrarroja al aplicarle una tensión.

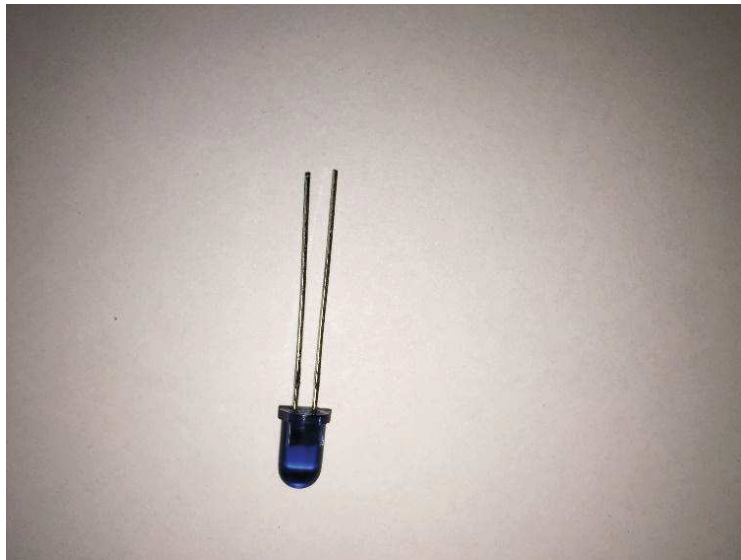


Ilustración 3.6. Emisor de infrarrojos

- **Receptor de infrarrojos:** Componente que capta la radiación infrarroja que le llega del emisor. Este sensor actúa como un circuito cerrado cuando recibe la radiación y cuando se corta esa radiación se abre el circuito.



Ilustración 3.7. Receptor de infrarrojos

- **Transistor 2222A:** también identificado como PN2222, es un transistor bipolar NPN de baja potencia de uso general. Sirve tanto para aplicaciones de amplificación como de conmutación (que es la que se utiliza en este circuito).

2N2222

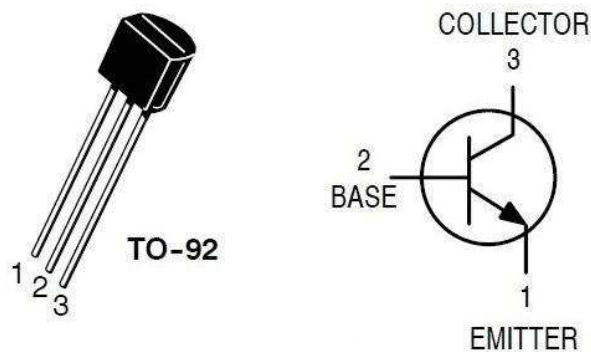


Ilustración 3.8. Transistor 2N2222

- **Potenciómetro:** Utilizado para regular la intensidad que le llega al receptor de infrarrojos. El potenciómetro cumple la función similar del

calibrador del otro sensor, que es ajustar la precisión de detección de los sensores.



Ilustración 3.9. Potenciómetro lineal

- **Resistencias de $1K\Omega$ y $10K\Omega$:** En el circuito se necesitan 2 resistencias para regular la tensión y no dañar el circuito.



Ilustración 3.10. Resistencias $1K\Omega$ y $10K\Omega$

Finalmente, el circuito creado será muy similar al sensor FC-51, pero con una mayor precisión para detectar objetos muy pequeños ya que el otro sensor se utiliza principalmente para detectar obstáculos y objetos grandes.

El funcionamiento de este circuito (Ilustración 2.11) se basa en la emisión de una radiación infrarroja la cual recibe el diodo receptor, este último al recibir la radiación se comporta como un circuito cerrado lo que da una tensión teórica de 0V en la base del transistor, lo que hace que el transistor no conduzca. Al pasar un

insecto a través del sensor, el receptor de infrarrojos cortará su circuito interno, por lo que en la base del transistor habrá un voltaje que lo excitará y en el colector se tendrá el mismo voltaje teórico del emisor que es 0V ya que este está conectado a tierra, y estos 0V son los que detectará el Arduino como paso de un insecto

Este circuito, al tener incorporado el potenciómetro que actúa como calibrador, hace que los sensores enfrentados (emisor y receptor) puedan tener una distancia entre ellos muy pequeña para detectar objetos muy pequeños o se pueda aumentar su distancia y calibrarlos para que sea capaz de detectar objetos más grandes.

El único inconveniente de este sensor, es que al haber una pequeña distancia entre los dos sensores enfrentados y al ser el tubo por donde pasan los insectos más grandes que la distancia que pueden detectar estos sensores, hay puntos ciegos en los laterales de estos sensores. Este problema se soluciona con la integración de piezas dentro del tubo formando una especie de cono en las zonas donde se encuentran los sensores, que se explicará en el siguiente punto.

El circuito final sería:

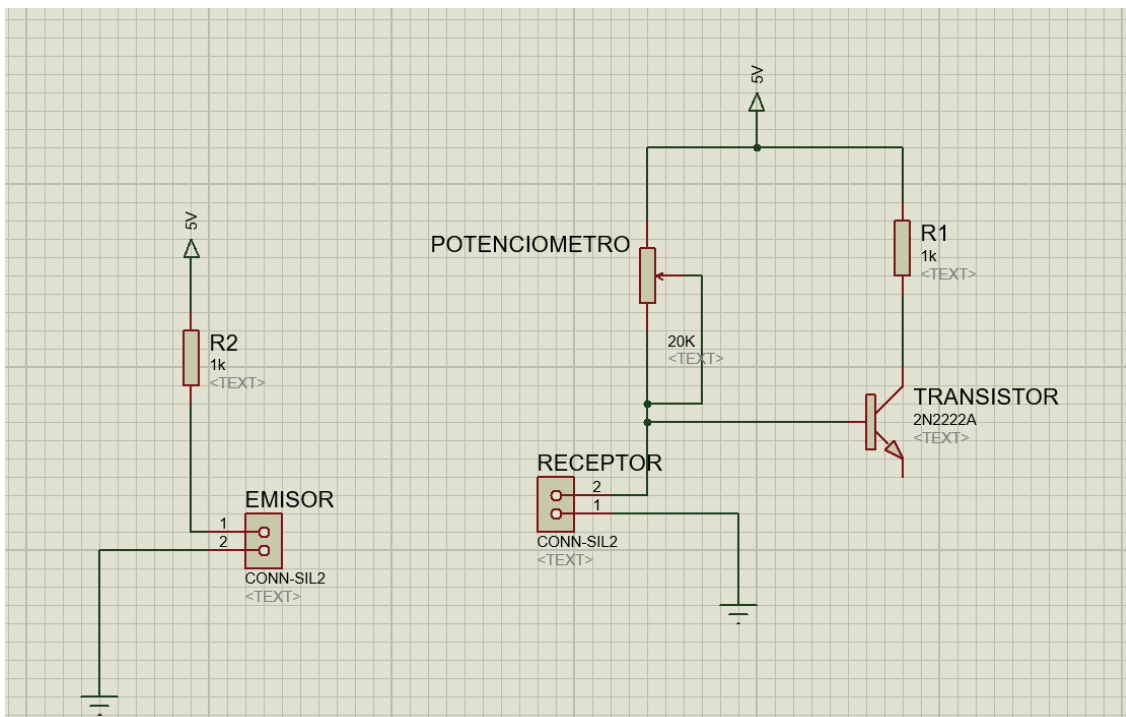


Ilustración 3.11. Sensor y receptor de infrarrojos

3.2. Diseño y construcción del hardware

Respecto al diseño hardware, se han realizado 2 tipos de diseños. Se dividirán en dos prototipos. El primer prototipo se desarrolla en primer lugar, al comienzo del proyecto para empezar a probar sensores y pruebas, aunque este será el prototipo que se pruebe en el campo por falta de tiempo. Se diseñara otro prototipo que será el producto final que se instalará en el campo una vez que esté validado.

3.2.1. Diseño del primer prototipo

El diseño de este prototipo se ha realizado con el programa de edición AutoCAD [4]. Es un programa de diseño en 2D y 3D, con el cual se pueden crear piezas fácilmente y se puedan imprimir posteriormente en una impresora 3D.

Este diseño cuenta con 2 partes:

- La primera parte es una pieza en la cual se meten todos los componentes electrónicos como son el Arduino, la batería, los sensores, etc. Esta parte contiene un tubo inicial por donde pasan las moscas en el primer momento y son detectadas por dos orificios donde se colocarían los sensores.

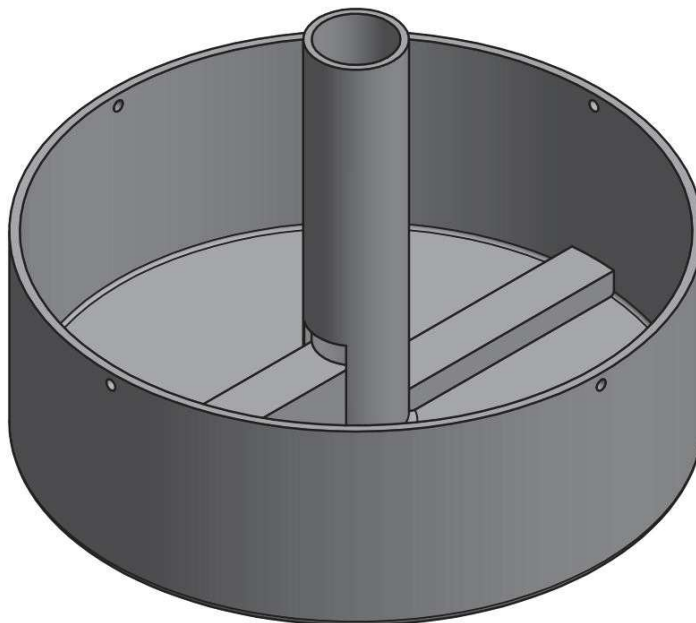


Ilustración 3.12. Pieza contenedora de componentes electrónicos

Esta pieza lleva dos soportes en la parte de abajo, uno más alto que otro, que son los lugares donde se pondrían los sensores. El orificio de abajo sería para la entrada de la mosca del olivo y el orificio de arriba sería para la salida. Pero solo se usará un soporte para contabilizar los insectos de entrada, el otro soporte quedará cerrado.

A los lados de los soportes se colocan los diferentes componentes como son el circuito de sensores emisor y receptor de infrarrojos, el microcontrolador Arduino y una batería para alimentar el sistema. Por último, en la parte superior se han hecho cuatro agujeros, para conectar la pieza 1 con la pieza 2.

- La segunda pieza, es una tapa que sirve para conectar la pieza 1 con el mosquero original de plástico que ya se mencionó anteriormente y para separar la zona de circuitería con la zona de almacenamiento de insectos.

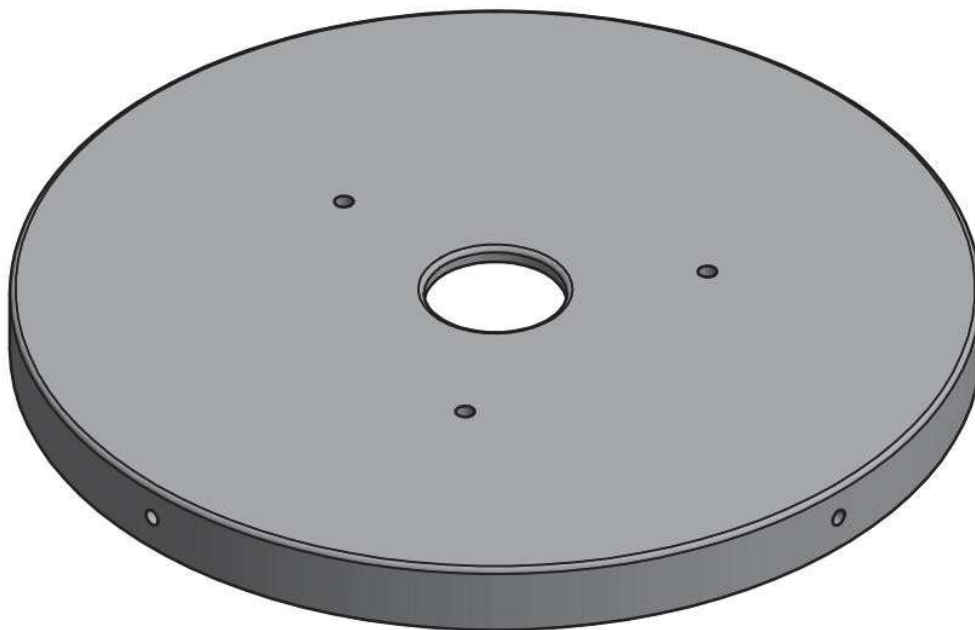


Ilustración 3.13. Tapa superior

Esta pieza tiene tres agujeros pequeños en la parte superior para poder unir esta tapa con el mosquero de plástico. Un orificio central, por el cual pasa el tubo de la pieza uno para que los insectos pasen de la pieza uno directamente al mosquero.

Por último hay cuatro agujeros en la parte lateral de esta pieza, que unen la pieza 1 con la pieza 2.

3.2.2. Diseño del segundo prototipo

El segundo prototipo se ha realizado con el programa de edición solidwork [5], de autodesk. Es un programa de edición en 2D y 3D, en el que te permite crear piezas fácilmente para poder ser impresas con una impresora 3D. Este prototipo se ha realizado una vez que se han probado los nuevos sensores y tiene un funcionamiento correcto.

De este prototipo, solo se realizará el diseño, ya que todas las pruebas se harán con el prototipo número 1, por falta de tiempo.

Es similar al primer prototipo, pero se ha ajustado al tipo de sensor que se usará finalmente y modificando sus piezas y formas, para que a la hora de imprimir estas piezas en una impresora 3D sea más fácil su impresión. Otro de los motivos por el que se elige el siguiente prototipo es que pueda contener los diferentes componentes electrónicos que se requieren para configurar totalmente la trampa inteligente. Este prototipo consta de las siguientes piezas.

- **Pieza contenedora de equipo electrónico.**

Esta pieza, es la pieza que contiene el equipo electrónico en su interior y es la primera pieza del nuevo diseño de la trampa inteligente. Esta, estará ubicada en la parte de abajo.

En esta pieza se puede observar que es de forma cuadrada. A diferencia de la primera que tenía forma circular, se ha pensado en esta forma para que sea más fácil trabajar con ella y haya espacio para los componentes electrónicos, que en este caso serían una batería para alimentar, un Arduino y una placa electrónica que son el circuito de los sensores de entrada.

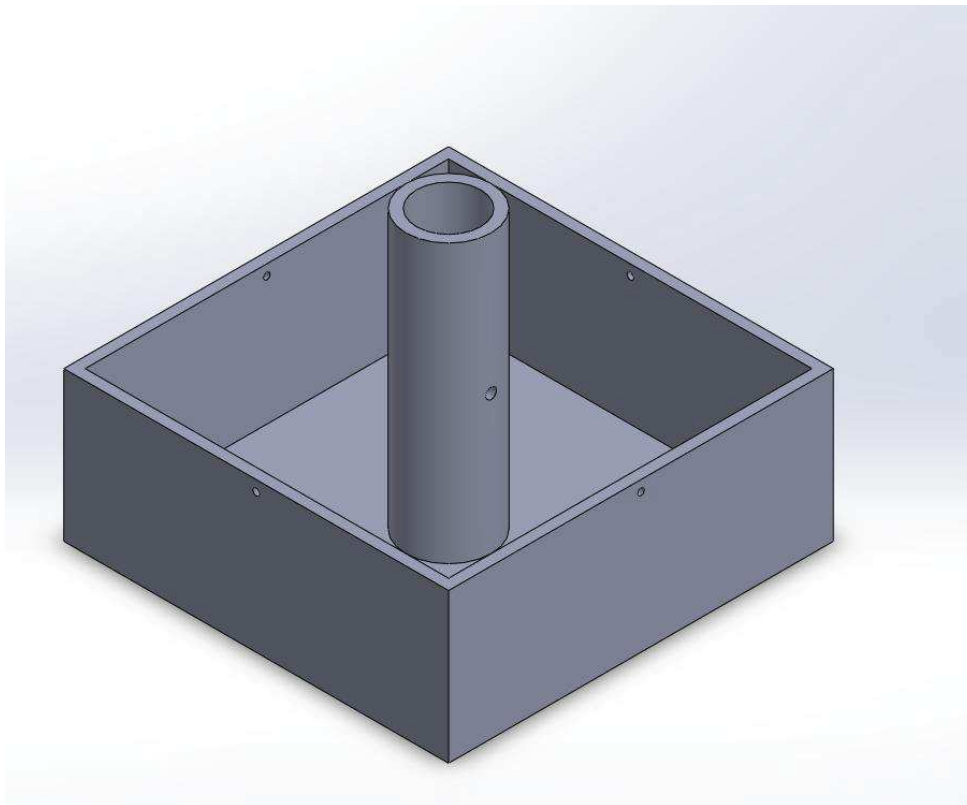


Ilustración 3.14. Pieza contenedora de equipo electrónico

Contiene un orificio central por el cual pasan los insectos al segundo compartimento y son detectados por los distintos sensores que irán conectados en los agujeros pequeños del tubo central. Para estos sensores se necesitan 2 agujeros, para controlar la entrada de los insectos. También dispone de 4 agujeros pequeños en los cuatro lados de esta pieza que la conecta con la siguiente pieza.

- **Pieza para estrechamiento del tubo central**

Como se ha explicado anteriormente, este nuevo sensor tiene puntos ciegos en los laterales de los sensores enfrentados, estos sensores al tener una distancia pequeña entre ellos, no alcanzan a detectar los objetos que pasan por el lateral de ellos. Para solucionar esto, se han pensado en la siguiente pieza.

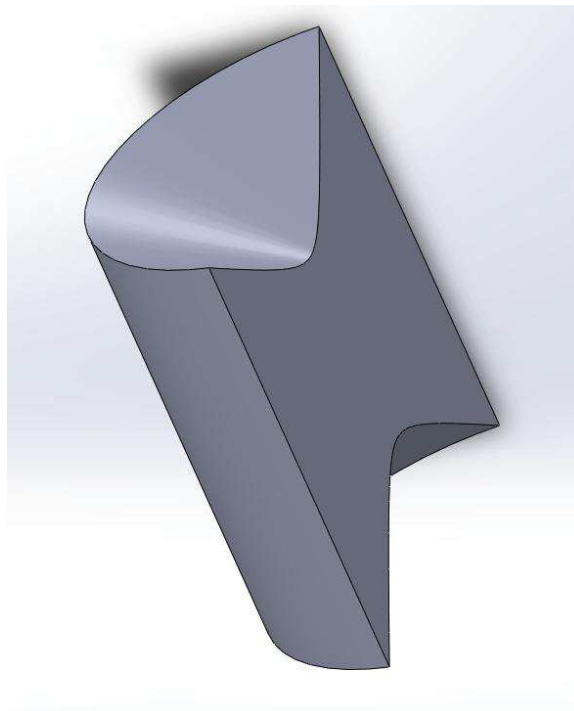


Ilustración 3.15. Pieza de estrechamiento

Se necesitarían dos piezas de este tipo. Estas dos piezas se colocarían justamente entre los sensores de entrada, para que el tubo se estreche solo en la parte interior donde se encuentran el emisor y receptor de infrarrojos y se puedan detectar fácilmente, así se elimina el punto ciego que tiene estos sensores. En el ensamblaje final se podrá observar mejor el funcionamiento de estas piezas.

- **Pieza contenedora de insectos**

Esta pieza, será el lugar donde se coloque el atrayente sexual, para atrapar estos insectos y después ser retirados. La pieza está unida a la pieza principal y su finalidad es separar los insectos de los componentes electrónicos, por lo que la trampa tendrá en su totalidad 2 compartimentos.

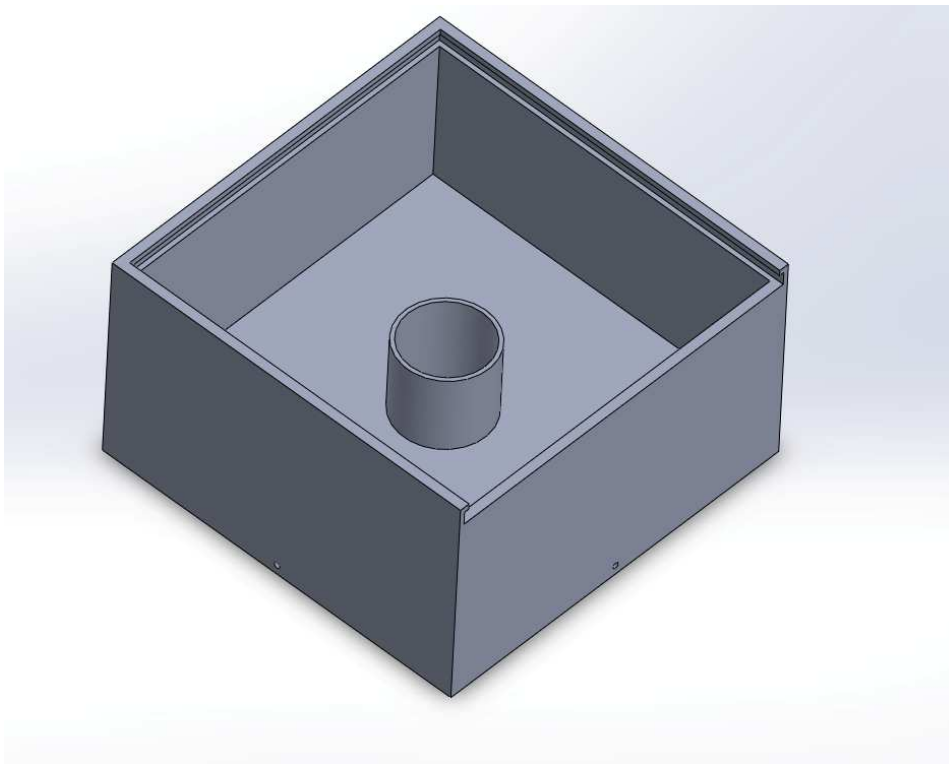


Ilustración 3.16. Pieza de almacenamiento de insectos

Se puede observar que hay un tubo central que será por el cual pasará el tubo del primer compartimento y donde saldrán los insectos una vez cuantificados. Tiene 4 agujeros, uno en cada lateral inferior de la pieza que servirán para unir esta pieza con la primera pieza. Por último dispone de unas guías en la parte superior en la cual se insertará una tapa transparente para que los insectos queden encerrados. La tapa se podrá retirar fácilmente una vez que se quiera limpiar este compartimento.

- **Tapa superior**

En último lugar, una tapadera que encaja en la parte superior de la trampa para que no escapen estos insectos, pero que se pueda retirar fácilmente una vez que se hayan contabilizado. Esta tapa será translúcida porque los insectos siempre se dirigen hacia la luz del sol. Esta penetra por la parte superior de la trampa y los insectos atrapados no podrán salir de la trampa.

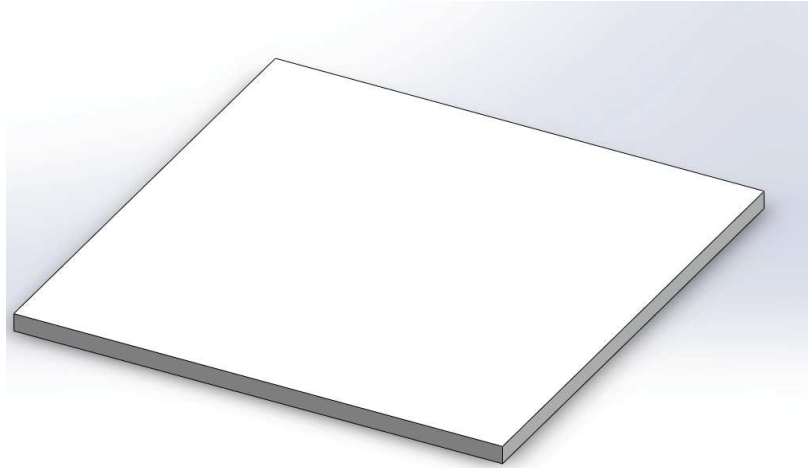


Ilustración 3.17. Tapa superior

Una vez que se han mostrado todas las piezas por separado, se muestra como sería el diseño del segundo prototipo de la trampa inteligente con todas las piezas conectadas.

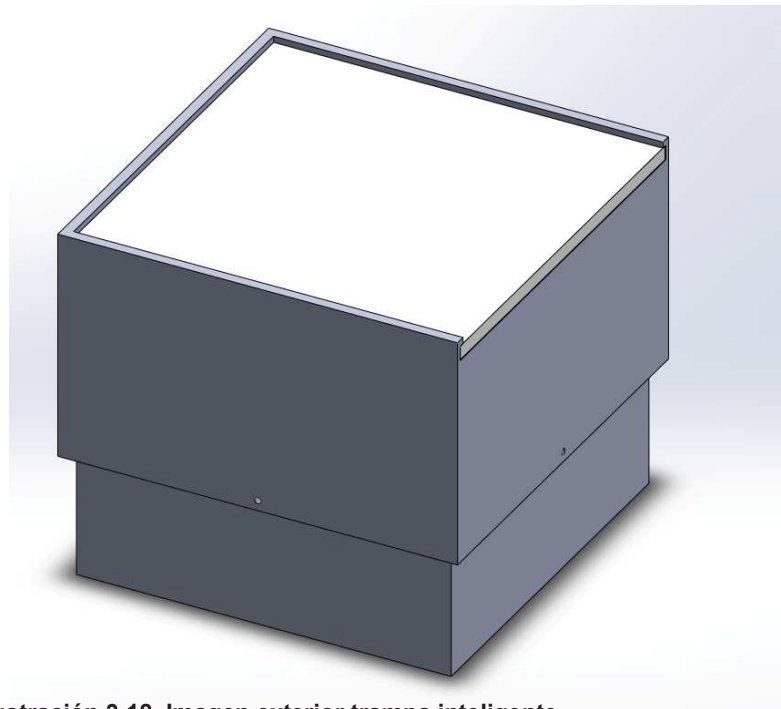


Ilustración 3.18. Imagen exterior trampa inteligente

Se muestra como sería el prototipo 2 exteriormente, en el que solo se puede observar los dos compartimentos, los agujeros que unen a estos y la tapa que cierra la parte superior de la trampa.

En la siguiente imagen se muestra la trampa, pero con un corte central para mostrar el interior de esta.

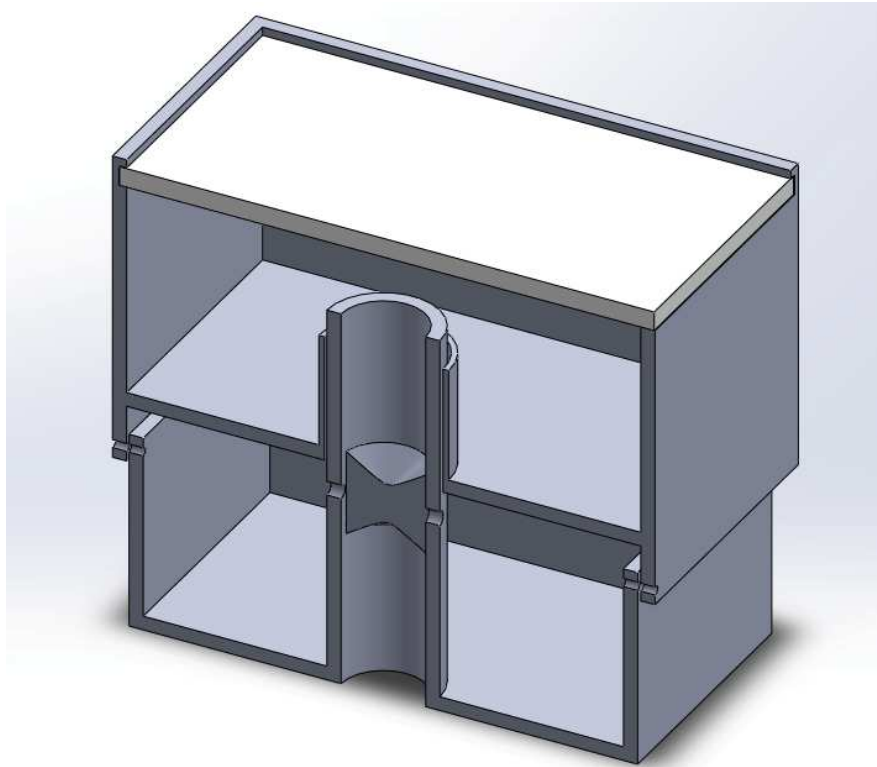


Ilustración 3.19. Corte transversal trampa inteligente

Se puede observar ahora como quedarían situadas las piezas que estrechan el tubo principal, también la separación de los dos compartimentos como ya se explicaron anteriormente y los agujeros laterales que unen las dos piezas.

3.2.2.1. Diseño en 3D de la placa base

Como ya se ha explicado en el punto 2.1, ha sido necesario fabricar un sensor más preciso, ya que el sensor FC-51 no detecta objetos pequeños correctamente. Como el nuevo circuito tiene varias resistencias, un transistor, diodos de emisión y recepción de infrarrojos, todos estos componentes tienen que estar conectados entre ellos. Por este motivo se ha diseñado una placa que contiene pistas donde se pueden colocar los componentes. Este diseño se hace para que los circuitos del sensor queden compactos y ocupen menos espacio.

Para el diseño de la placa, se ha utilizado el programa *Proteus 8 Professional* [6], que es un programa de diseño en 3D de placas electrónicas, para posteriormente realizar esta placa. Para este diseño se llevan a cabo tres partes en el programa:

- La primera parte se trata de encontrar todos los elementos que se van a poner en la placa, para obtener las huellas que dejarían los componentes en la placa.

Después se debe de diseñar el circuito que se coloca en la placa con la parte de proteus llamada *Schematic Capture*. En este caso se ha colocado el circuito que se mostró en el punto 2.1 y el Arduino micro que es el que utilizaremos para realizar el control de esta trampa inteligente. Una vez colocados los componentes, se conectan entre sí y con el Arduino.

Quedaría el circuito de la siguiente manera:

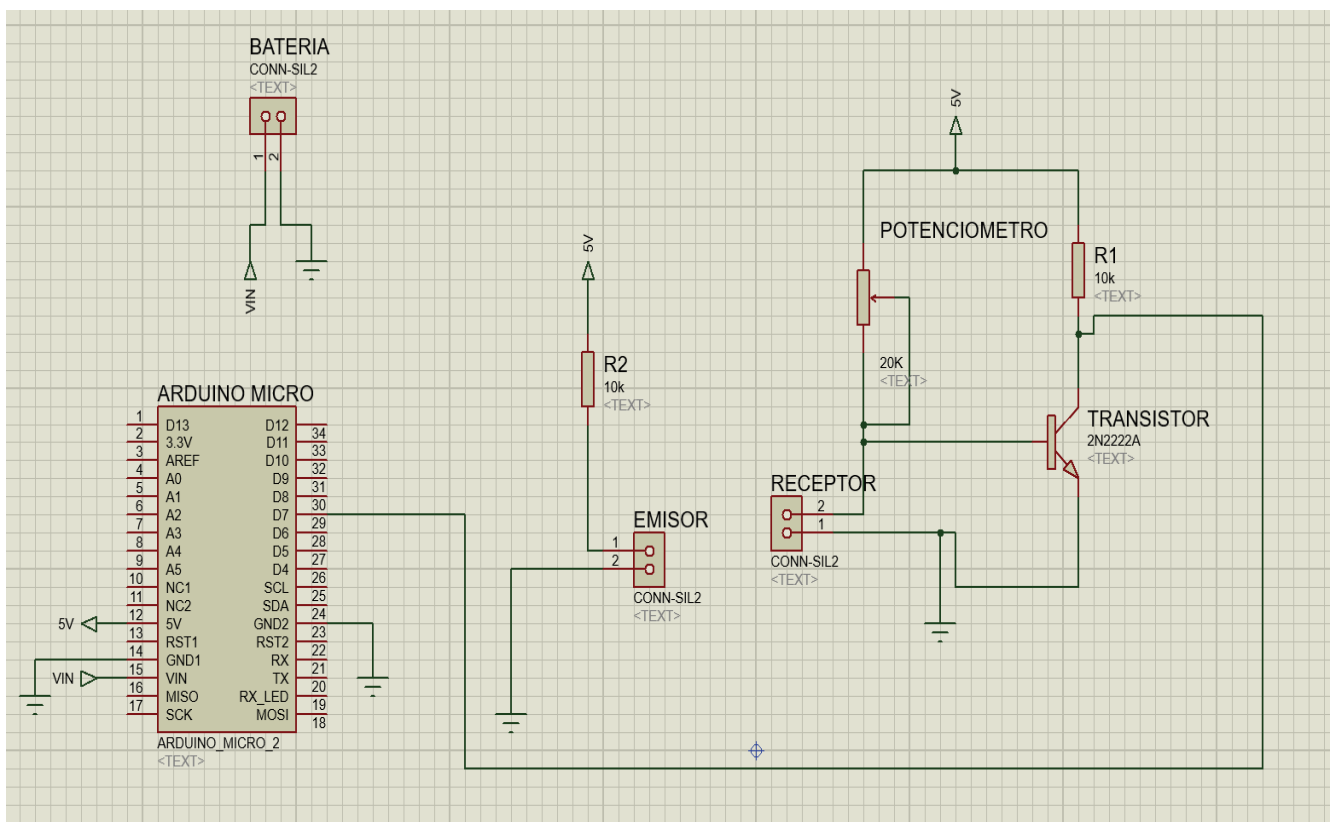


Ilustración 3.20. Circuito diseñado con proteus

- La segunda parte de este programa, se utiliza para colocar el circuito que se ha diseñado anteriormente en la placa. Una vez que se colocan estos componentes en la placa, se deben conectar a través de pistas, que son las pistas que quedarán en la placa final.

Para realizar esto se utiliza el *PCB Layout* de este programa. En este caso quedaría de la siguiente manera:

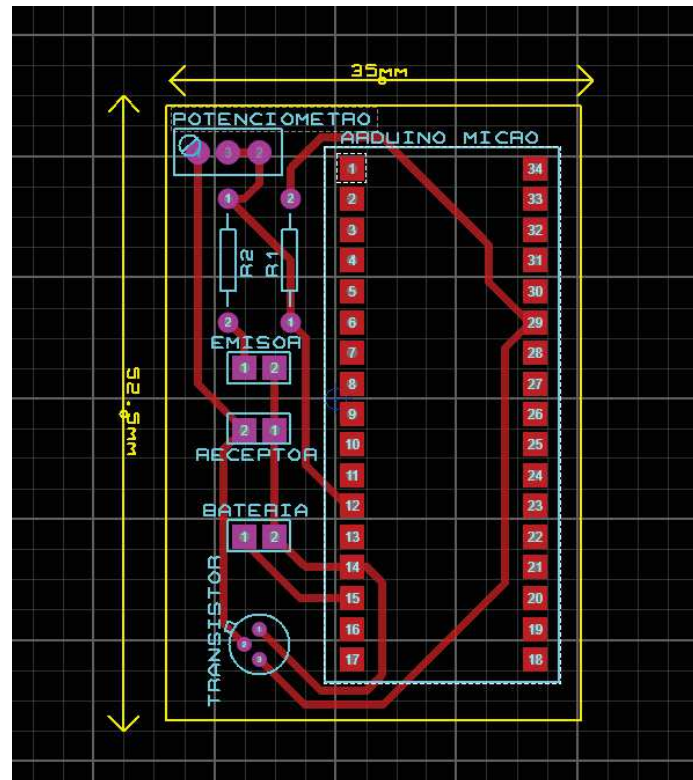


Ilustración 3.21. Diseño de placa electrónica

Se han colocado todos los componentes de la forma más compacta posible y se han conectado entre sí utilizando solo una cara de la placa, para que a la hora de imprimir la placa, no se tenga que estar cambiando de posición para hacer pistas en la parte de arriba y en la parte de abajo. También el *PCB Layout* aparte de como quedarían los componentes en la placa, te muestra el tamaño que ocupará esta. En este caso tiene unas medidas de 52,5x35 mm. Son tamaños correctos para que entren en la pieza 1 del mosquero (la parte donde se encuentran los componentes electrónicos).

- La tercera y última parte de este diseño da la posibilidad de visualizar en tres dimensiones la placa que hemos diseñado con la función de proteus *3D visualizer*, que nos muestra cómo quedaría nuestra placa con todos los componentes que se han diseñado.

En este caso quedarían de la siguiente forma:

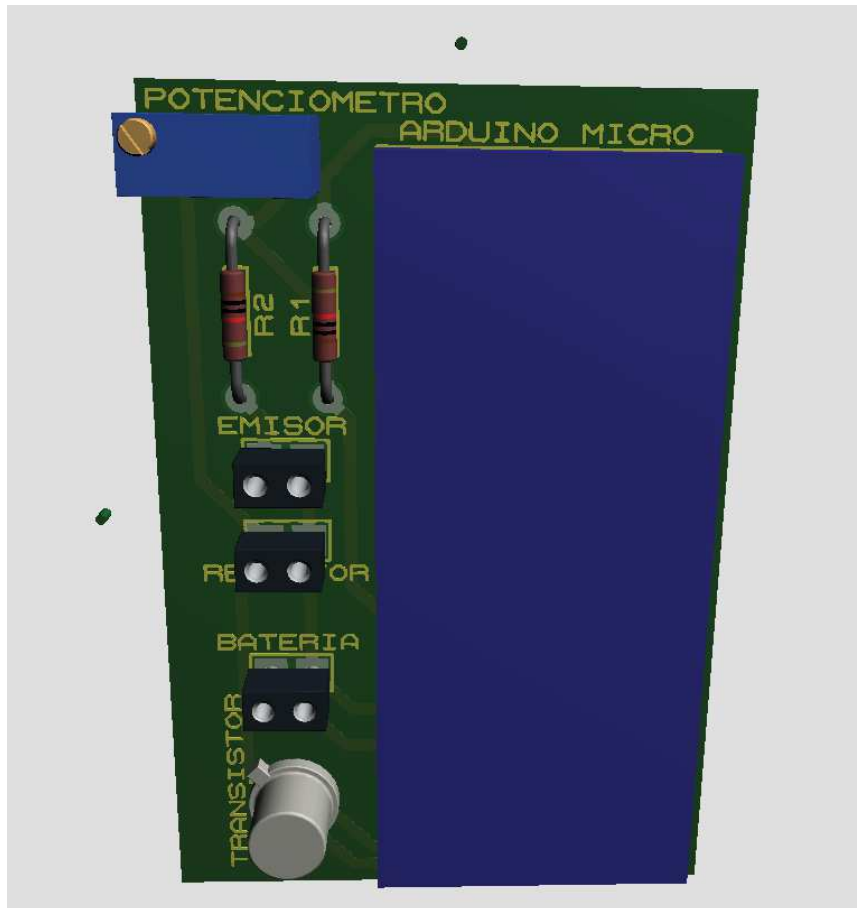


Ilustración 3.22. Placa electrónica en 3D

Se pueden observar los diferentes componentes conectados al Arduino y las pistas que los unen entre sí. Esta parte del programa es meramente visual, para mostrar si el diseño creado quedará como se había pensado en un principio.

3.3. Diseño e implementación del software para el microcontrolador

3.3.1. Arduino

Arduino es una plataforma de hardware libre para desarrollar y experimentar programación de bajo nivel. Está basado en una placa con un microprocesador, en el que se programa en C, que es el lenguaje de programación más básico.

La placa de Arduino posee un microcontrolador Atmel AVR y numerosos puertos que actúan como entradas y salidas. Los microcontroladores más usados en la producción de placas de Arduino son el Atmega168, Atmega328, Atmega1280 y Atmega8, por su sencillez y su bajo coste de producción y fabricación, que hacen posible el desarrollo de múltiples diseños.

En este proyecto se utilizará una clase de Arduino llamado Arduino micro, que tiene las mismas funciones que el Arduino básico (Arduino uno), pero con menos puertos digitales, menos funciones a interrupción y un menor consumo energético y dado que este proyecto estará situado en el campo, esto beneficia para tener una mayor autonomía. Para este proyecto, sus pines son suficientes para realizar la programación de la trampa inteligente.

El uso del Arduino micro en este proyecto se debe a que tiene un tamaño muy reducido y los compartimentos de la trampa inteligente son reducidos también. El Arduino micro es el más parecido al Arduino Leonardo en la gama de Arduino.

El Arduino micro es una placa electrónica basada en el ATmega32U4, desarrollado conjuntamente con Adafruit. Tiene 20 pines digitales de entrada/salida (de los cuales 7 se pueden utilizar con salidas PWM y 12 entradas como analógicas), un oscilador de 16 MHz de cristal, una conexión micro USB, una cabecera ICSP, y un botón de reinicio que detiene y reinicia la última acción cargada desde el ordenador al procesador. Contiene todo lo necesario para apoyar el microcontrolador. Para alimentarlo basta con conectarlo a un ordenador con un cable de micro USB para empezar o una batería conectada al pin Vin de este microcontrolador. Tiene un factor de forma que le permite ser fácilmente colocado en una placa. Permite alimentar a 5 V o 3.3V desde dos de sus pines, y tiene 2 pines asignados a tierra.

El Arduino micro tiene dos medios de alimentación, el primero es mediante micro USB de transferencia donde aparte de recibir alimentación conecta con la interfaz de programación donde se introducen y cargan los códigos que el Arduino leerá. El segundo medio es utilizar una alimentación por batería recargable para suministrar 5V-12V de entrada. La regulación de tensión la lleva a cabo el propio Arduino de la placa. En este proyecto se utilizará la alimentación de batería de iones de litio de 8,4V.

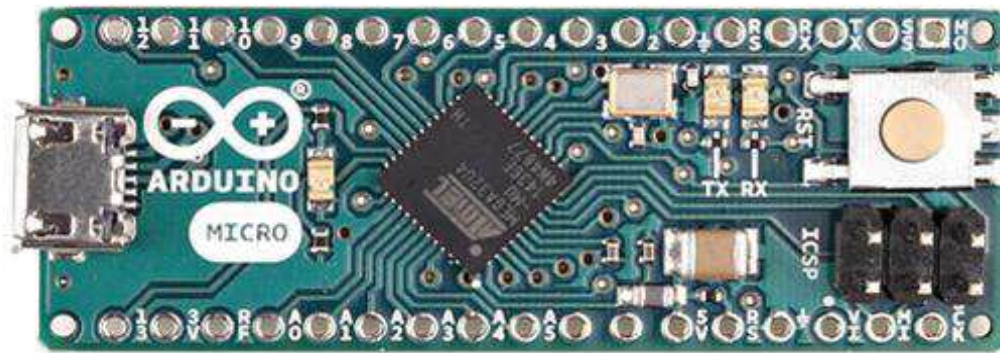


Ilustración 3.23. Arduino micro

Una vez que cualquier Arduino se conecta a un ordenador, se necesita descargar de forma gratuita la interfaz oficial para trabajar con él. Esta interfaz te permite escribir directamente sobre hojas en blanco (*sketches*), que se van guardando en el directorio predeterminado o preasignado. Cada *sketch* tiene una propia carpeta con el mismo nombre que el archivo, y permite guardar y cargar librerías definidas por el usuario o descargadas de la red.

Las *sketches* de Arduino funcionan principalmente en código C, con tres grandes bloques principales. Por un lado y al inicio, está el bloque en el que se cargan las librerías y se definen e inicializan las variables. En segundo lugar, se encuentra el bloque principal llamado *setup*, donde se definen las configuraciones, como por ejemplo la configuración de algún pin de entrada o salida, también es el lugar donde se inicia la comunicación serial, se resetean las variables y se llevan a

cabo las acciones que ocurren solamente la primera vez que se ejecute el código. Por último lugar, se encuentra el cuerpo del programa llamado *loop*. Este cuerpo es una función que en su interior contiene el código del programa con todas las acciones que se deben ejecutar. El *loop* es una función especial que permanecerá ejecutándose de forma cíclica sin interrupciones a menos que se detenga expresamente con alguna declaración condicionada en su interior.

Otra parte necesaria para el correcto funcionamiento del Arduino es la comunicación serial. Esta comunicación consiste en la transmisión e intercambio de paquetes de datos multiplexados en tiempo, bit a bit, entre dos o más destinatarios. Para el Arduino es especialmente útil esta comunicación, ya que es su medio para conectarse con toda clase de periféricos y programas externos, como el utilizado en este proyecto para capturar en un archivo de texto los datos impresos de los sensores que detectan la presencia de la mosca del olivo. Así mismo, aun sin una comunicación establecida entre Arduino y otro hardware o software, la interfaz de programación permite una visualización del puerto serial con un PC en tiempo real, lo que facilita el trabajo con ella.

El puerto serial de Arduino permite la transmisión de datos desde 300 baudios a 230,400 baudios. El baudio (band en inglés) es una unidad de medida utilizada en teoría de comunicaciones para representar el número de símbolos por segundo en un medio de transmisión digital. Dependiendo del esquema de modulación, cada símbolo podrá codificar 1 o más bits [7].

3.4. Desarrollo de plataforma web.

Para el desarrollo de la plataforma web se ha utilizado una plataforma llamada ThingSpeak que es un framework gratuito que permite comunicarte de una forma sencilla con equipos que dispongan de acceso a internet. Permite a los desarrolladores interactuar con los dispositivos utilizando tecnologías Web estándar. ThingSpeak puede funcionar bien a través de un servicio gratuito de host, o bien servidores personales. Algunos aspectos de la plataforma ThingSpeak incluyen registro, procesado y distribución de la información, servicios basados en la localización, actualizaciones de estado, integración con redes sociales, aplicaciones y plugins. Dentro de estos plugins podemos encontrar ejemplos de conexión con

Arduino, interacción de ambiente con Matlab e incluso plugins para enviar mensajes vía Twitter. Un framework (ilustración 3.24) realmente muy interesante.



Ilustración 3.24. Página oficial ThingSpeak

La página de ThingSpeak, nos permite crear un canal privado o público, donde los datos que reciba serán guardados y mostrados a través de gráficas o plugins. También nos permite descargar la base de datos de todo el canal con la extensión cvs. El canal creado para este proyecto muestra una gráfica con el número de moscas capturadas por la trampa inteligente respecto al tiempo. Esta gráfica se puede editar y se puede añadir la localización de donde se encuentra el dispositivo que manda esta información [8].

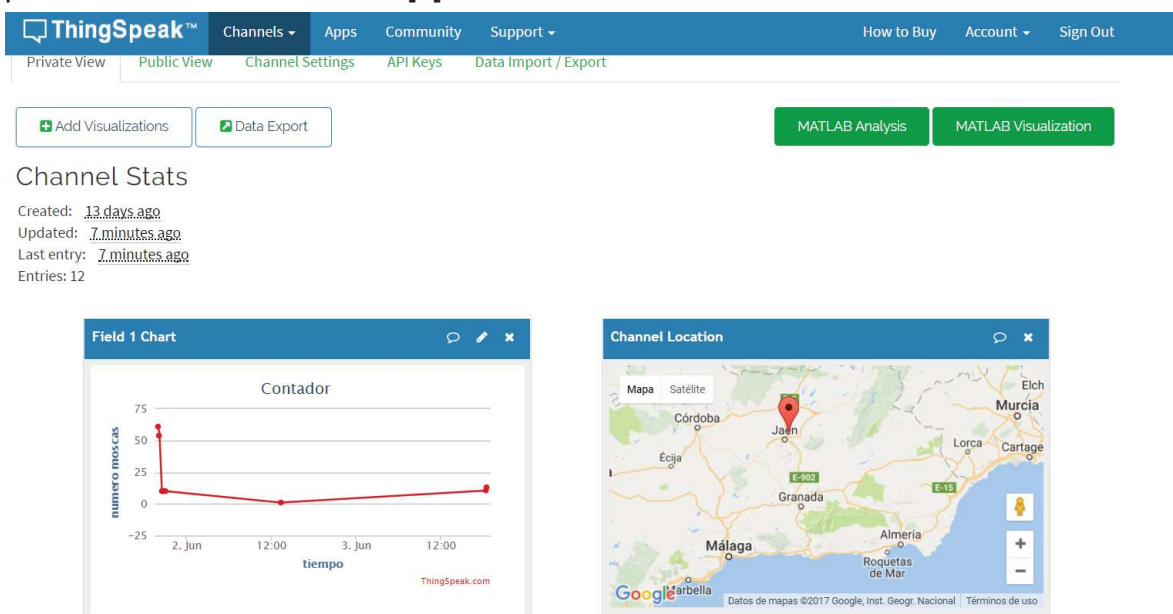


Ilustración 3.25. Configuración canal creado

3.5. Implementación del software de comunicación con la plataforma web

Para subir los datos a la plataforma, hay que utilizar una shield para Arduino y tener conocimientos previos de algunos conceptos antes de utilizar esta shield.

3.5.1. Internet de las cosas

La expresión “internet de las cosas” hace referencia a la interconexión entre objetos de consumo o de uso cotidiano (aparatos electrónicos, electrodomésticos, etc.) a través de ciertos dispositivos capaces de conectar a estos a la red. En la siguiente ilustración se resume este concepto.

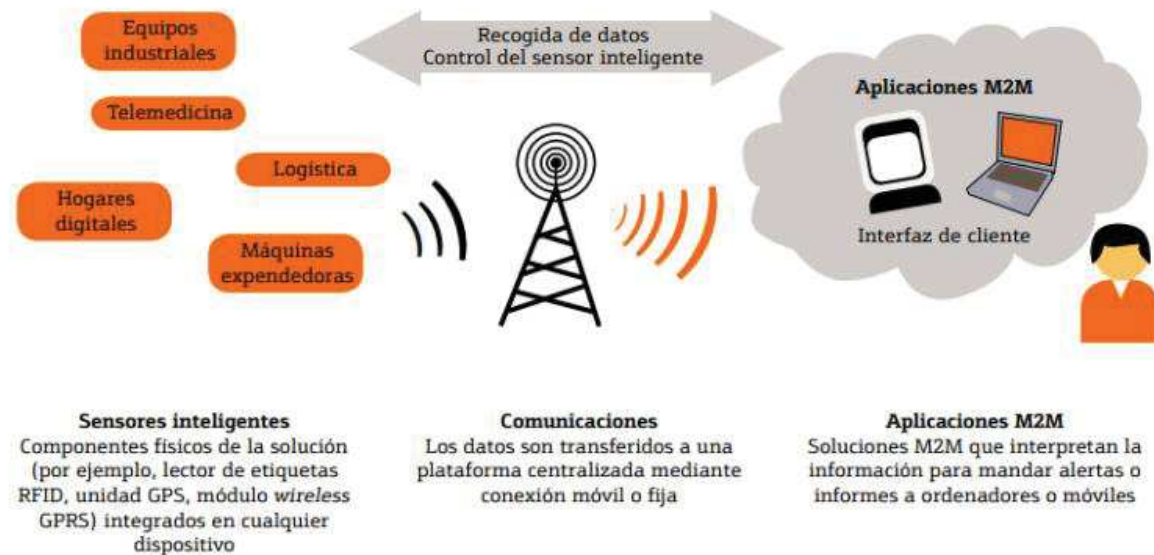


Ilustración 3.26. Internet de las cosas

Las utilidades del internet de las cosas pueden considerarse prácticamente infinitas. Cada vez son más los medios que tenemos a nuestra disposición para implementar, de manera no muy compleja, un posible dispositivo capaz de dotar a cierto objeto una función de conexión a la red. Gracias a la llegada de un protocolo de comunicación llamado IPv6 se hace posible la gestión de direccionamiento de innumerables dispositivos.

Por otro lado, ya son muchas las aplicaciones móviles y servicios en la nube que nos permiten la conexión a todos estos dispositivos y proporcionan una vía para el tratamiento de una inmensa cantidad de datos en el tiempo real, facilitando la integración de infinidad de sensores aplicables prácticamente a cualquier tipo de necesidad. De hecho, ya han surgido incluso redes sociales de sensores, como la plataforma “Xively” o “ThingSpeak” que ya se ha mencionado antes, donde los usuarios comparten datos a tiempo real procedentes de distintos sensores. En los próximos años se espera un gran aumento en el número de equipos de uso cotidiano interconectado, entre otras cosas, gracias a la inminente llegada de todos estos sensores si se le puede llamar inteligentes, conectados a nuestros hogares.

Dentro de las diversas plataformas, se puede resaltar que Arduino ha marcado un punto de inflexión en el sentido de lograr obtener la conexión a esta comunicación de manera realmente sencilla, convirtiéndose en la herramienta ideal para llevar a cabo multitud de prototipos y obtener nuevos casos de uso. Gracias a su bajo costo, sencillez, versatilidad y variedad de modelos, resulta una herramienta de gran ayuda a la hora de implementar ideas y soluciones. Además es posible conectar casi cualquier tipo de sensor que se requiere y para ser precisos podemos conectar otros dispositivos de diferentes tecnologías de comunicación (GSM/GPRS, 3G, Bluetooth, RFID, Wi-fi, etc.) aprovechando una variedad de shields en su expansión [9].

3.5.2. GSM

Es la abreviatura de “Sistema Global para las comunicaciones Móviles” (en inglés, Global System for Mobile communications). Conocido como estándar de segunda generación (2G), su principal diferencia respecto a la primera generación de teléfonos móviles es que se estandarizó totalmente digital. En Europa las bandas de frecuencia utilizadas son 850, 900 y 1800 MHz, mientras que en E.U.A. se usa la banda de frecuencia de 1900MHz. Por lo que los dispositivos pueden operar en todo el mundo y conoce como cuatribanda (Quadband). El estándar permite transmisiones digitales de voz y datos, como mensajes de texto (SMS) o mensajes multimedia (MMS).

Respecto a su arquitectura de red, en GSM todo terminal móvil debe estar constituido de una tarjeta SIM (Módulo de identificación de abonado) y el propio dispositivo, normalmente un teléfono móvil. La tarjeta SIM es la encargada de identificar en la red al usuario y al terminal móvil. Estos dispositivos se identifican gracias a un número exclusivo de identificación denominado IMEI (Identificador internacional de equipos móviles), compuesto por 15 dígitos. Por otro lado, cada tarjeta SIM también posee un número de identificación denominado IMSI (Identificador internacional de abonados móviles). La red GSM está compuesta por múltiples estaciones base (BTS), y a su vez, se conecta a un controlador de estación base (BSC), encargado de administrar la red. A esto se le conoce como subsistema de estaciones base. En un nivel superior los Centros de conmutación móvil (MSC), al que se conectan físicamente los controladores de estaciones base. Este es encargado de establecer la conexión con la red de telefonía pública y con internet. Su operación depende del operador de red telefónica. En la siguiente figura se muestra una representación de esta comunicación a grandes rasgos [10].

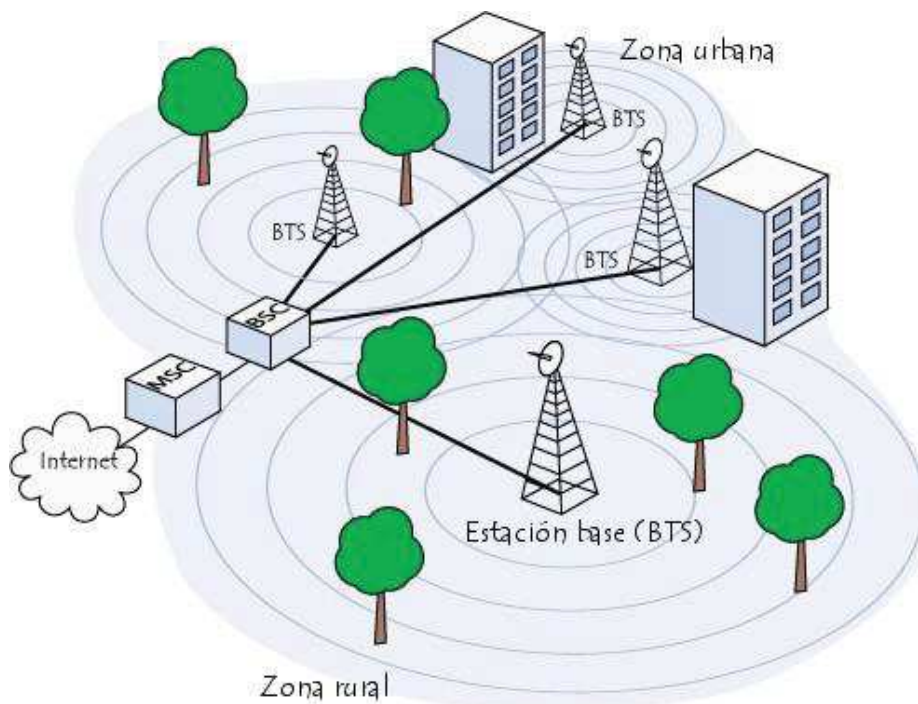


Ilustración 3.27. GSM

3.5.3. GPRS

El conocido como Servicio General de Paquetes vía Radio “GPRS” (en inglés, General Packet Radio Service) es una evolución del sistema GSM. Es también conocido como GSM++, pero dado que se trata de un estándar de telefonía móvil intermedio entre la segunda generación (2G) y la tercera (3G), normalmente recibe la nomenclatura de 2.5G. GPRS extiende la arquitectura del estándar GSM para permitir la transferencia de datos mediante la conmutación de paquetes con velocidad de transferencia que rondan los 114 kbps. La transmisión de datos sólo utiliza la red cuando es necesario, permitiendo una transmisión por volumen de información en lugar de tiempo de conexión, por lo que sólo se utilizará la red cuando se envíe o se reciba un paquete de datos. Para el acceso a la red de datos, se utilizara el protocolo IP, mientras que para el transporte de voz la arquitectura de la red GSM [11].

3.5.4. GSM Arduino shield 2

El GSM Arduino shield 2 permite a la tarjeta Arduino conectarse a la red, hacer y recibir llamadas de voz, enviar y recibir mensajes SMS a través de un tarjeta SIM de algún operador de telefonía móvil insertada en la shield. La shield tiene un modem de radio M10 por Quectel. La comunicación es posible gracias a los comandos de comunicación AT. En el IDE de Arduino la librería GSM tiene un gran número de métodos para la comunicación con la shield. Para la comunicación con el software se utilizan los pines digitales 2 y 3. El pin 2 está conectado al TX y el pin 3 al RX del chip M10. El PWRKey del módem está conectado al pin 7 de Arduino. Por lo que es importante tener en cuenta que no se debe utilizar ninguno de los pin anteriormente mencionados. El M10 es un módem cuatribanda GSM/GPRS que funciona con las frecuencias GSM 850, 900, 1800 y 1900 MHz. También es compatible con los protocolos TCP/UDP y HTTP a través de una conexión GPRS. La velocidad de conexión es de 85.6 kbps [12].

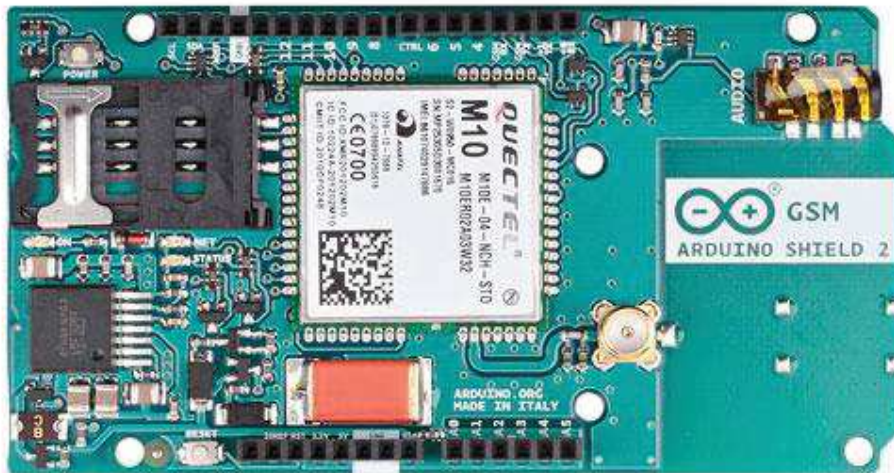


Ilustración 3.28. Shield GSM 2

3.5.4.1. Interconexion entre Arduino micro y shield gsm2

Para mandar información a la plataforma web ThingSpeak, en concreto al canal que se ha configurado, se tiene que utilizar el Arduino micro junto la shield que estandariza sus pines para que puedan encajar sobre esta shield gsm 2. Una vez que encajan las 3 piezas, hay que modificar algunos pines como son doblar la patilla del pin 7 del Arduino micro para que esta patilla no encaje en el mismo pin 7 de la shield gsm 2 y no mande información ya que el pin 7 de este es utilizado para la interrupción del código necesario para la trampa inteligente. Como el pin 7 lo necesita la shield gsm 2 para controlar la misma placa, la solución es cortocircuitar la patilla del pin 6 con la patilla del pin 7 de la shield gsm 2 para que los datos de control pasen del pin 6 del Arduino micro a los dos pines 6 y 7 de la shield gsm 2.

Otro problema es que por defecto, la comunicación de la shield gsm 2 se realiza por los pines digitales 2 y 3. En el Arduino uno esto funciona sin modificaciones, pero para utilizar Arduino micro, se necesitan modificaciones. En el Arduino micro, la biblioteca de GSM utiliza el pin digital 8 para comunicarse con el micro. Por lo tanto, es necesario enviar la señal desde el pin 2 de la shield gsm 2 al pin 8 del Arduino micro, y al mismo tiempo evitar que el pin 2 del Arduino micro se

conecte a la shield, por lo que habrá que juntar las patillas 2 y 8 a través de un cable y al mismo tiempo doblar la patilla del pin 2 del Arduino micro.

El resultado final para poder establecer una conexión con el servidor se puede ver en la ilustración 2.29.

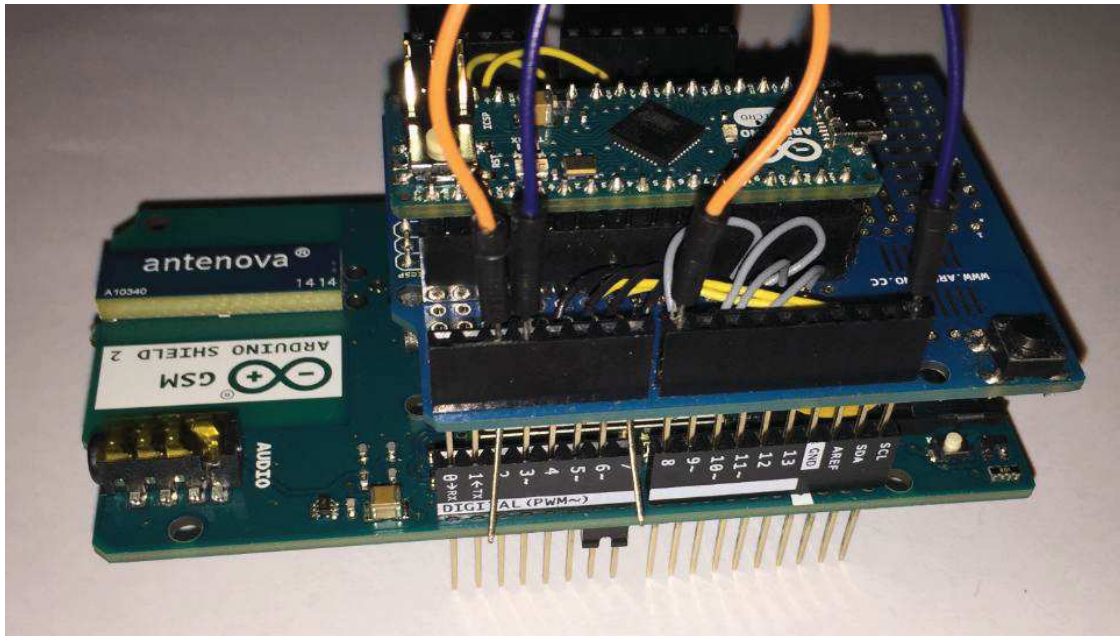


Ilustración 3.29. Conexión entre Arduino micro y shield gsm 2

3.6. Flujograma del código utilizado

El código utilizado para la detección de moscas de la trampa inteligente en el cual se detecta los insectos que entran en esta trampa para después enviar los datos a la plataforma web ThingSpeak, se deben separar en dos códigos diferentes. Un código principal en el que se basa todo el programa de la trampa y otro para flashear la memoria EEPROM, que es la única memoria no volátil que tiene Arduino.

3.6.1. Programa principal.

Este código se basa en controlar los insectos que entran en la trampa y guarda esta información en la memoria interna del Arduino, para después subirla a la plataforma ThingSpeak. Para la realización de este software, se utilizará la interrupción externa del Arduino micro, que está en el digital número 7 y es la única que se puede utilizar para una interrupción externa. También son necesarios los pines 2 y 3 de la shield gsm para enviar y recibir la información guardada en la memoria interna.

En este programa, en primer lugar se incluirán las librerías necesarias para realizar el código, en este caso se usaran la Liberia para utilizar la memoria interna del Arduino “EEPROM”, otra librería para poner el Arduino en modo sueño “SLEEP” y otra librería para enviar y recibir información a la plataforma ThingSpeak “GSM” . Una vez cargadas las librerías y creadas las variables, se pondrá en modo sueño (modo sleep) el microcontrolador, para que tenga un menor consumo mientras no esté detectando la entrada de insectos, ya que en principio este sistema estará alimentado con una batería externa. En el futuro, se alimentara además con una placa fotovoltaica, para no estar constantemente cargando esta batería.

Una vez que se ha activado la interrupción externa del Arduino micro (pin digital 7), el Arduino sale del modo sueño y entra en el cuerpo del código principal, incrementando el contador del número de moscas y guardando esta información en la memoria EEPROM, que es la única memoria no volátil que tiene Arduino. Esta información se guardará en esta memoria hasta que se resetee esta.

Cuando se hayan detectado 10 moscas del olivo, se mandará la información que se había guardado anteriormente en la memoria EEPROM a la plataforma ThingSpeak [13], además de enviar un sms al usuario de la cantidad de insectos que han entrado en la trampa inteligente. Después de mandar la información, el Arduino volverá a ponerse en modo sueño (modo sleep) hasta que vuelva a detectar la entrada de otro insecto y así consuma menos energía.

En la ilustración 2.30 se puede observar el flujograma del código principal.

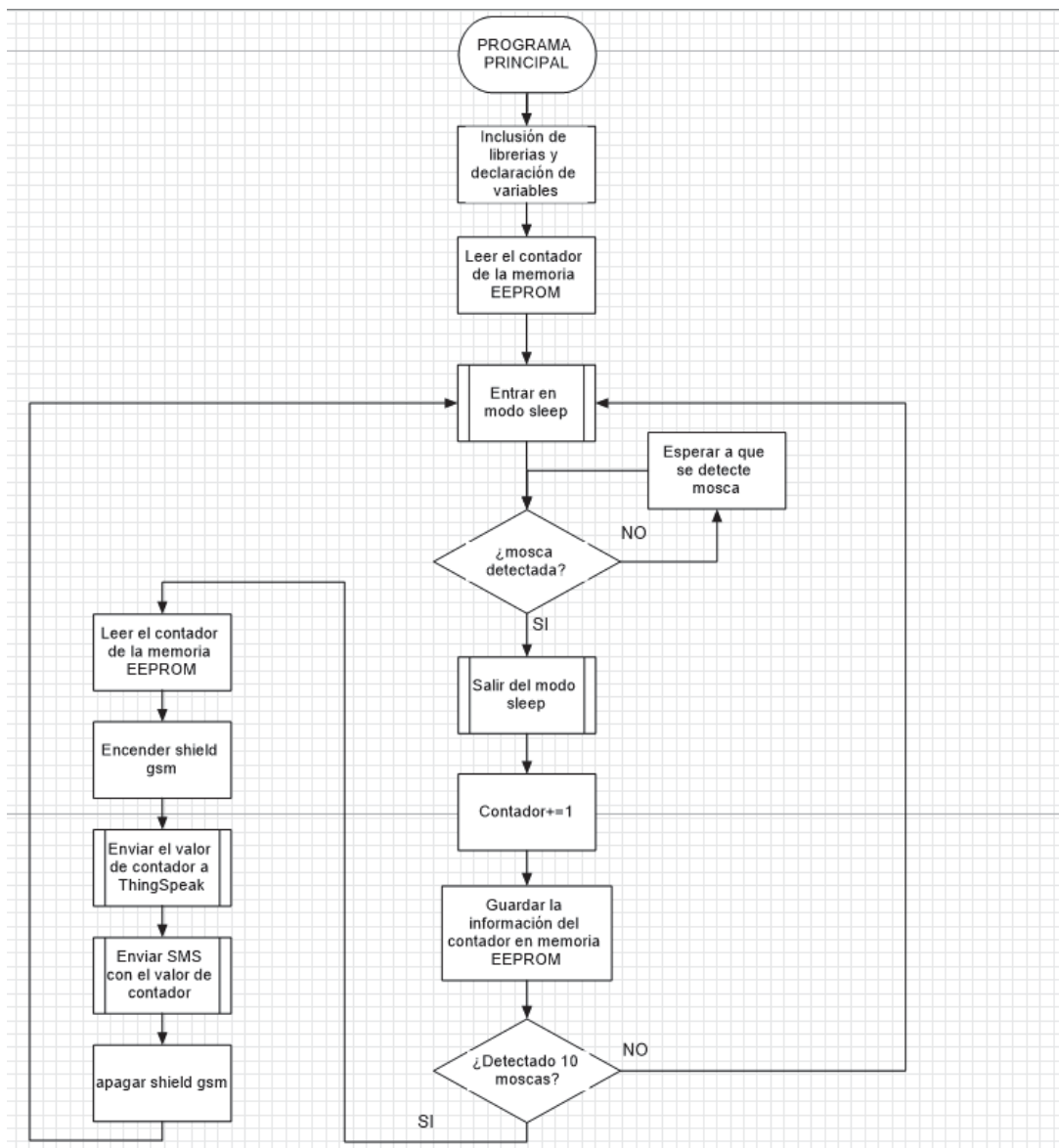


Ilustración 3.30. Flujograma código principal

3.6.2. Código para resetear la memoria interna del microcontrolador.

Para completar el software se necesitará un segundo programa que haga un reseteo o flasheo a la memoria no volátil EEPROM.

En este simple programa, se vuelve a cargar la librería para utilizar la memoria EEPROM, se crea una variable que tenga un valor de 0 y se introduce en la memoria no volátil EEPROM, para que el contador de las moscas del olivo se vuelva a reiniciar una vez que ya no se quiera contabilizar más insectos. Como este código de programa es tan simple, no se mostrará flujograma del código de flasheo de la memoria EEPROM.

3.7. Autonomía de la trampa inteligente

Para la alimentación de la trampa inteligente se ha utilizado una batería de iones de litio con una tensión nominal de 7,4V y una capacidad de 5200 mAh. La gráfica de descarga de esta batería se ha obtenido utilizando el mismo circuito implementado en la trampa inteligente. Los datos obtenidos se han subido a la plataforma ThingSpeak, para aprovechar que este canal ya estaba creado y configurado.

Dado que un Arduino solo es capaz de medir tensión de valores, entre 0 y 5 voltios en sus pines analógicos, se ha implementado un divisor de tensión en el que la medida que se realiza en el pin analógico es como máximo de 5V y una vez leída la tensión, el Arduino devuelve un valor de 10 bits (entre 0 y 1023 donde 0 es 0V y 1023 es 5V). Este valor no es el valor real, por lo que hay que convertirlo a través de un factor de conversión en el valor original de la batería.

El divisor de tensión utilizado para para medir el valor de la batería se ha calculado para que cuando la batería tenga la carga completa (8,4V) de una tensión en este divisor de 5V como máximo, por lo que se han calculado las resistencias adecuadas para esto.

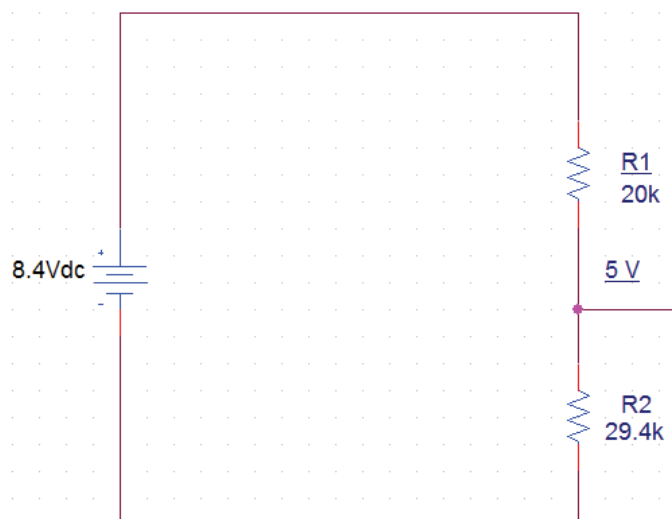


Ilustración 3.31. Divisor de tensión

Para conseguir la resistencia de 29,4 K Ω se ha utilizado una resistencia variable de 10 K Ω en serie con una resistencia fija de 20 K Ω para llegar al valor exacto. El valor en este divisor de tensión es medido por un pin analógico de Arduino micro que realiza la siguiente función:

$$\text{Tensión en el divisor} = \frac{5}{1024} \cdot \text{medida analógica del pin}(0 - 1024)$$

Ecuación 3.1. Medida del divisor de tensión

El valor obtenido, será un valor entre 0V y 5V, que se tiene que multiplicar por un factor que viene de:

$$\text{Factor de conversión} = \frac{8,4 \text{ V}}{5} = 1,68$$

Ecuación 3.2. Factor de conversión

Finalmente la tensión leída en el pin analógico, se multiplica por este factor y se obtiene la medida real de la batería, la cual se ha obtenido utilizando también la Shield gsm 2 para realizar la gráfica de descarga de la batería que se ha usado. En esta gráfica se puede observar como el voltaje de la batería va disminuyendo poco a poco manteniéndose prácticamente constante desde los 8,4V hasta 6,7V. Después se produce una bajada de tensión drástica desde 6,7V aproximadamente, hasta 5,5 V en la que se descarga la batería en poco tiempo y finalmente pasa de estos 5,5 V a 0V.

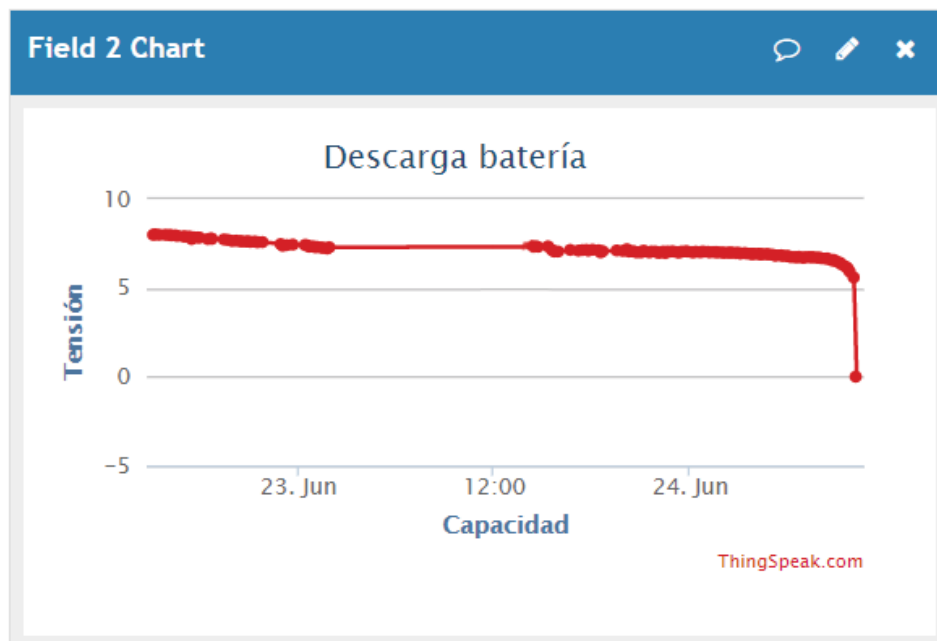


Ilustración 3.32. Gráfica descarga de batería ThingSpeak

El software utilizado para calcular estos valores y subir la información a ThingSpeak se muestra en el siguiente flujograma:

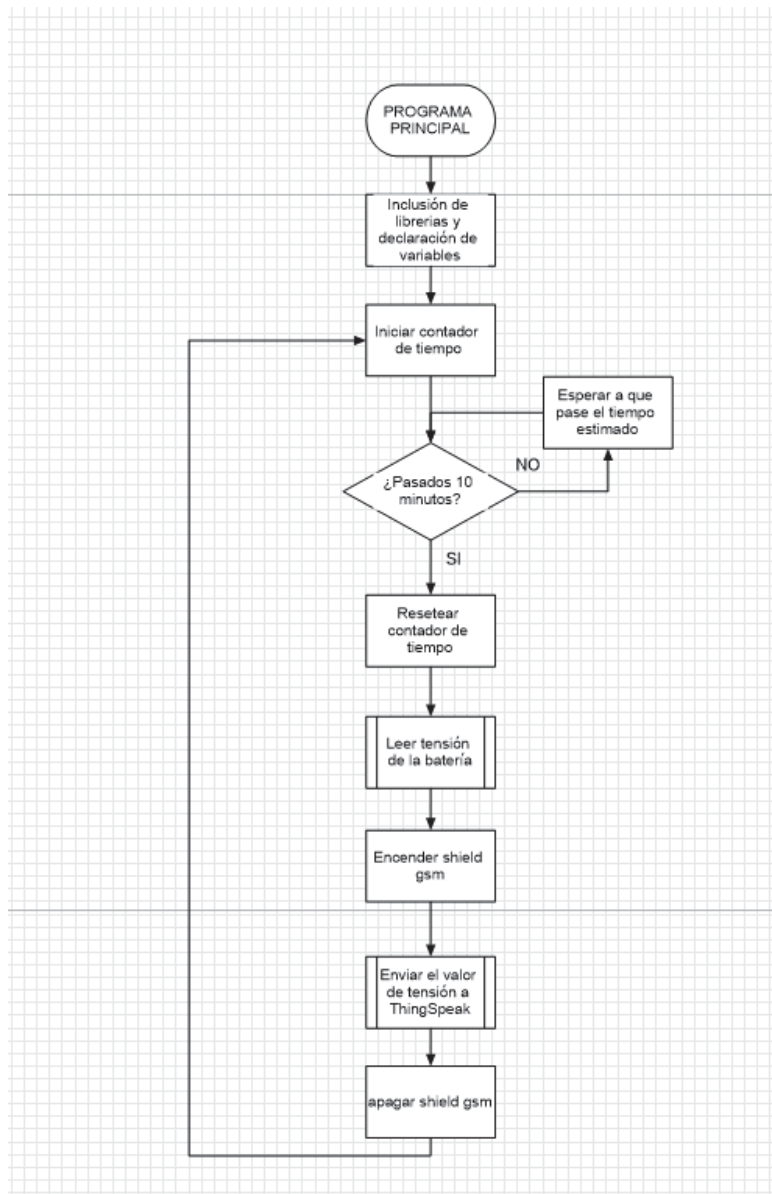


Ilustración 3.33. Flujograma para calcular descarga de la batería

El programa principal inicializa las variables e incluye la librería “GSM” como se explicó anteriormente. Inicia un temporizador de 10 minutos a través de una función de Arduino llamada millis [14]. Cuando pasen los 10 minutos, se leerá el valor de tensión de la batería y se subirá al canal de ThingSpeak configurado previamente y se reseteará este temporizador.

4. RESULTADOS

4.1. Porcentaje de acierto del sensor receptor y emisor de infrarrojos

Gracias al potenciómetro del sensor construido, se puede ajustar para detectar diferentes tamaños, en principio para las pruebas reales de la captura de la mosca del olivo se ha ajustado para un tamaño de este insecto de 5 mm porque este sensor ha sido diseñado para controlar este insecto, pero para comprobar que este sensor tiene una buena precisión, se ha realizado una prueba de hacer pasar bolas con un diámetro de 2 mm por la trampa inteligente, para hacer un porcentaje de acierto de la trampa. Para la detección de estas bolas se ha ajustado el sensor para que sea capaz de detectarlas.

- Primera prueba:

50 bolas lanzadas y 28 detectadas.

- Segunda prueba:

50 bolas lanzadas y 30 detectadas.

- Tercera prueba:

50 bolas lanzadas y 33 detectadas.

Con estos datos, se puede obtener un porcentaje de acierto:

$$\%acierto = \frac{\text{Total de bolas detectadas}}{\text{Total de bolas lanzadas}} \cdot 100 = \frac{91}{150} \cdot 100 = 60,67\%$$

Ecuación 4.1. Ecuación porcentaje de acierto

Se puede comprobar que tiene un alto porcentaje de acierto de estas bolas que tienen un diámetro tan pequeño, a esto se le añade el hándicap de la velocidad a la que pasan estas bolas por los sensores emisor y detector de infrarrojos que hace que aumente aún más la dificultad de detectar estas bolas.

Este sistema se ha diseñado de tal forma que no detecte falso positivos que puedan influir a la hora de contabilizar estos insectos.

4.2. Pruebas de consumo

Una vez que todo el producto se ha montado y está en pleno funcionamiento con todos sus componentes, se ha realizado una prueba de consumo de la batería que alimenta la trampa inteligente. Esta prueba se trata de activar la trampa durante 10 horas y comprobar después con la gráfica de descargar de la batería, cuanto se ha consumido y así poder hacer una estimación de la duración de esta una vez que se haya implementado en el campo.

El consumo de este sistema se debe de dividir en varias partes, la primera es cuando el Arduino micro se encuentra en “modo sleep” en el que su consumo es alrededor de 26,8 mAh. La segunda parte es cuando este sistema detecta la entrada de un insecto y sale del modo sleep para contabilizarlo y guardar la información en la memoria EEPROM, por lo cual se tendrá un consumo en torno a 54,9 mAh. Por último, se encuentra la parte de consumo en la que se ha llegado a un número de moscas contadas, en la que hay que activar la shield gsm 2 para enviar los datos a ThingSpeak y se tendría un consumo que variará según la potencia incorporada en la shield gsm 2 para enviar esta información, oscilará entre 65 mAh y 111mAh aproximadamente.

La tensión medida en las 10 horas en la que se tuvo el sistema conectado a la batería decreció desde 7,90V hasta 7,73 V, la diferencia es de 0,17 V. Estimando que la parte lineal de la batería tiene una diferencia de 1,7 V aproximadamente, se puede calcular que la duración de la batería será en torno a 100 h.

4.3. Montaje final

Una vez realizadas todas las pruebas convenientes, la trampa inteligente ha sido implementada en el campo para verificar que contabiliza los insectos que son capturados por esta y después mandar un sms con la contabilización de los insectos contados además de subirlos a la plataforma ThingSpeak.

En las siguientes ilustraciones se mostrará los componentes electrónicos que se han incorporados en el interior de la trampa inteligente para la detección de

moscas del olivo. Para estas pruebas no se incorporó la shield gsm 2 ya que sabiendo que tiene un correcto funcionamiento, solo era necesario probar que el Arduino micro detectaba los insectos que entraban en la trampa. También se mostrará la trampa colocada en las zonas para detectar en primer lugar moscas de la fruta que tiene un tamaño similar a la de la mosca del olivo para hacer las primeras pruebas y por último, esta trampa colgada en un olivo para detectar la mosca *Bactrocera* (*Dacus oleae*).

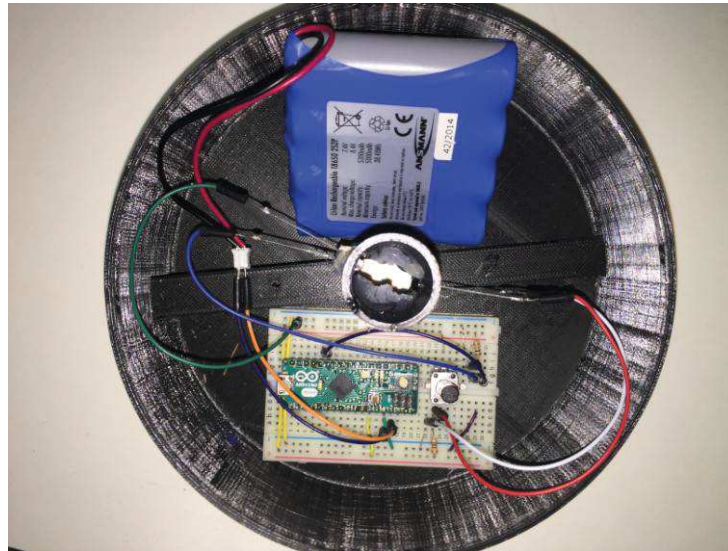


Ilustración 4.1. Componentes de trampa inteligente

La ilustración 4.2 muestra la trampa colgada en un patio común en el que contiene una fruta en su interior para detectar moscas de la fruta y realizar las pruebas.



Ilustración 4.2. Prueba con mosca de la fruta

La ilustración 4.3 muestra el lugar final donde se ha implementado la trampa inteligente con feromonas en su interior a la espera de que las moscas del olivo entren en ella.



Ilustración 4.3. Implementación final en olivo

5. PRESUPUESTO DEL PROTOTIPO

El presupuesto de este proyecto se tiene que dividir en dos partes, una parte se basa en la estimación de los costes indirectos de este proyecto y una segunda parte del precio del producto que se va a desarrollar.

5.1. Estimación de costes

En este proyecto, es conveniente hacer una estimación de los costes que supondrá hacerlo. En esta parte se incluirá el tiempo estimado para la realización del proyecto.

Para elaborar este coste, se debe asignar un precio por hora del ingeniero que desarrolla el mismo y la herramienta principal que use que en este caso será un ordenador. A continuación se calculan los diferentes precios por hora.

En este proyecto, se ha necesitado un ingeniero que haya diseñado la trampa inteligente para atrapar el insecto del olivo. El precio por hora de un ingeniero es de 30 €/h y para la realización de este proyecto se han necesitado 165 horas.

También se ha utilizado como herramienta principal un ordenador portátil Asus de 15,6 pulgadas con sus características básicas. Estas características son las siguientes:

- Procesador Intel Core i5 a 2,8 GHz.
- Memoria RAM de 12 GB.
- Tarjeta gráfica Nvidia Geforce 930M.
- Almacenamiento de 1000 GB.

El precio total del ordenador portátil es de 780€. Se estima que la vida útil de este portátil es de 5 años de vida y como el proyecto se ha realizado en unos 4 meses. Realizando los cálculos necesarios, se puede saber que el coste de esta herramienta serán aproximadamente de 56€.

En la siguiente tabla se puede observar el coste total indirecto.

Código	Denominación	hora	Precio/hora	Total
UD 001	Ingeniero(Diseño electrónico)	60	30 €	1800 €
UD 002	Ingeniero(Diseño gráfico)	15	30 €	450 €
UD 003	Ingeniero(Diseño software)	80	30 €	2400 €
UD 004	Ingeniero(Diseño plataforma web)	10	30 €	300€
UD 005	Ingeniero(Implementación)	5	30 €	150 €
UD 006	Ordenador personal	-	56 €	56 €
TOTAL				5.156,00 €

Tabla 5.1. Costes indirectos

5.2. Precios unitarios del producto

Todos los precios unitarios vienen dados con el I.V.A. incluido por lo que no hará falta el cálculo de este al final.

Código	Denominación	Cantidad	Precio unitario	Total
01	Arduino micro.	1	24,08 €	24,08 €
02	Batería.	1	46,00 €	46,00 €
03	Placa electrónica.	1	1,30 €	1,30 €
04	Piezas plástico trampa.	1	48 €	48 €
05	Cable con conector.	15	0,07 €	1,05 €
06	Resistencia 1K.	1	0,06 €	0,06 €
07	Resistencia 10K.	1	0,06 €	0,06 €
08	Transistor 2222A.	1	0,13 €	0,13 €
09	Emisor infrarrojos.	1	0,48 €	0,48 €
10	Receptor infrarrojos.	1	0,45 €	0,45 €
11	Potenciómetro	1	1,69 €	1,69 €

Diseño y desarrollo de una trampa Inteligente para el control de plagas		CONCLUSIONES Y TRABAJOS FUTUROS		
12	Shield gsm 2	1	82,89 €	82,89 €
13	Tarjeta Sim	1	10,00 €	10,00 €
14	Relé	1	5,90 €	5,90 €
TOTAL				222,09 €

Tabla 5.2. Presupuesto del producto

5.3. Precio total del proyecto

Una vez realizada la estimación de los costes y el precio de implementación de la trampa inteligente, se puede concluir que el precio total estimado del proyecto asciende a la expresada cantidad de "CINCO MIL TRESCIENTOS SETENTA Y OCHO CON NUEVE CÉNTIMOS (5378,09€).

6. CONCLUSIONES Y TRABAJOS FUTUROS

Se puede concluir con la validación de este sistema en el campo, obteniendo buenos resultados ya que la trampa inteligente tiene buena precisión para detectar la mosca del olivo, porque este proyecto se ha ajustado a su tamaño.

Como tiene un correcto funcionamiento, se puede desarrollar ahora el prototipo completo con la placa electrónica para que los componentes ocupen menos espacios y no haya cables que puedan soltarse. También se pueden incluir nuevas mejoras como las que se dictan a continuación.

6.1. Posible diseño futuro

La trampa inteligente se podría mejorar añadiéndole otro sistema de emisión y recepción de infrarrojos para detectar los insectos que pueden salir y que este sistema no contabilice 2 o más veces el mismo insecto. Para implementar el nuevo sistema habría que modificar dos cosas:

- Modificación de la carcasa exterior de la trampa:

A la trampa diseñada habría que añadirle en el tubo central dos orificios para colocar los nuevos sensores de emisión y recepción de infrarrojos, además de añadirle dos nuevos semicilindros en revolución de estrechamiento como ya tienen los sensores de entrada.

La siguiente imagen se muestra el corte transversal de la trampa inteligente puede observar la implementación de los orificios de salida, ahora la trampa contabilizará los insectos que entran y también los que se puedan volver a escapar.

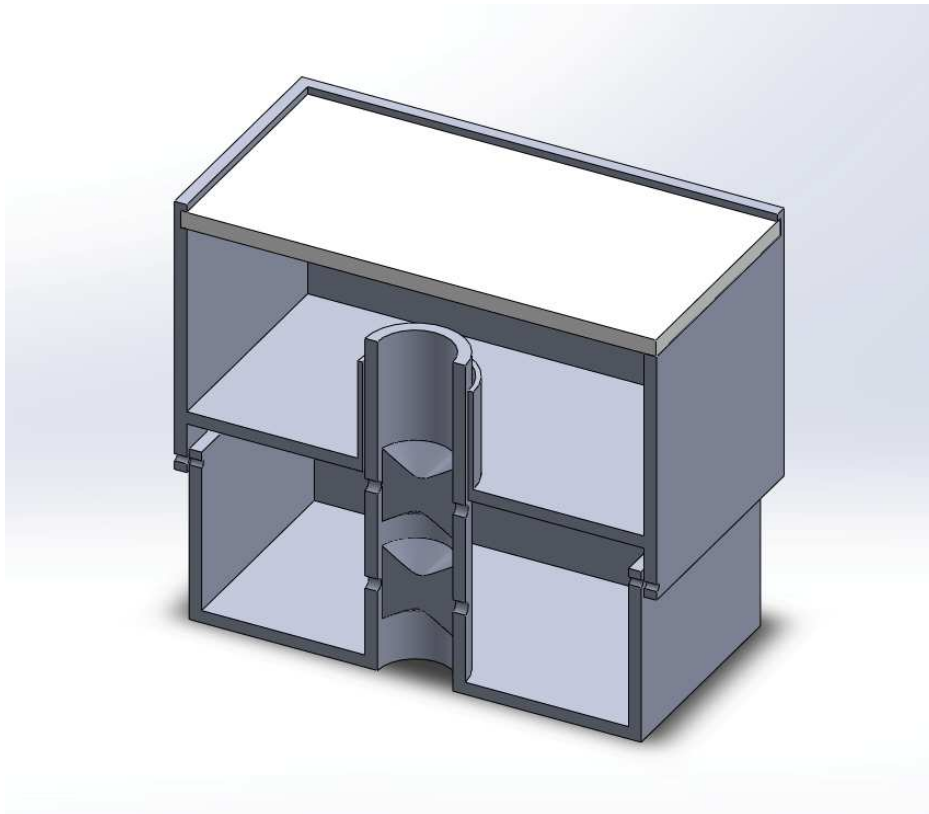


Ilustración 6.1. Diseño futuro mosquero entrada y salida

- Nuevo diseño de la placa electrónica

Para detectar la salida de los insectos que puedan escapar de la trampa, hay que añadir otro circuito de emisor y recepción de infrarrojos, por lo que hay que rediseñar la placa electrónica de la trampa inteligente. Se deben incorporar en una misma placa dos circuitos de detección y se ha diseñado de la misma forma que se ha diseñado el circuito ya implementado. En las siguientes imágenes se muestra como se explicó anteriormente 3 imágenes, el circuito, su Layout y la visión 3D de cómo quedaría la placa final.

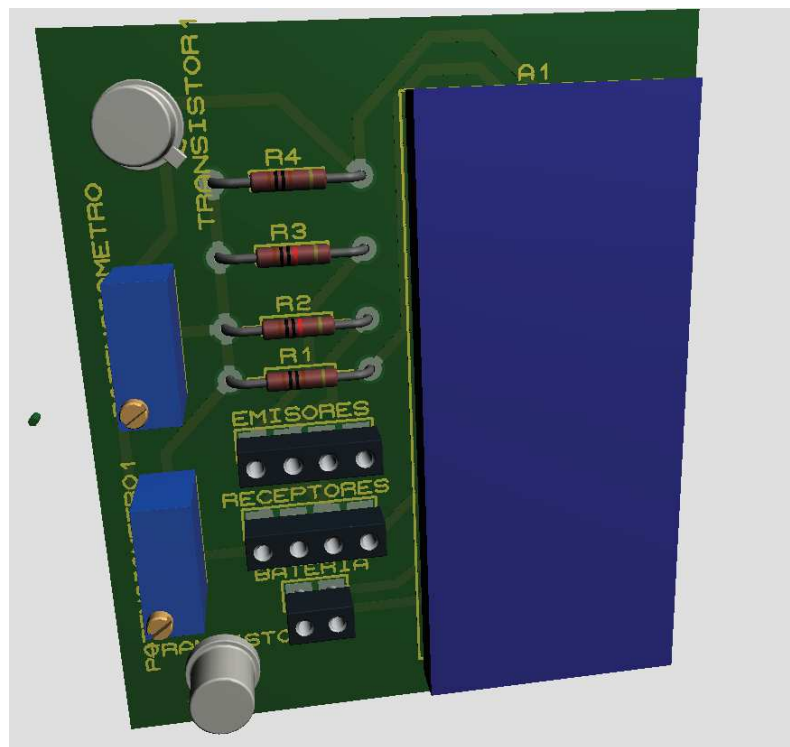


Ilustración 6.2. Diseño futuro placa electrónica entrada y salida

Para poder añadir un nuevo circuito que detecte los insectos que se puedan escapar de la trampa se debe de programar en el microcontrolador ATmega del Arduino micro, para definir otro pin de este como interrupción, ya que el Arduino micro tiene solo un pin de interrupción que es el pin digital 7. Para configurar otro pin como interrupción se deberá tener conocimientos avanzados de programación de esta placa.

La nueva placa programada, tendría la función de detectar los insectos que entran como la tenía anteriormente, con la nueva función de detectar también los insectos que salen de la trampa inteligente, así no contabilizaría algún insecto que haya entrado pero se quede en mitad de camino y vuelva a salir.

- Implementación de una placa fotovoltaica:

Además de realizar estos cambios, también se puede implementar una placa fotovoltaica alrededor de la carcasa de la trampa inteligente para que esta tenga un consumo autónomo y no dependa siempre de que se tenga que cargar la batería cuando esta se consuma.

Para calcular la potencia que se necesita para alimentar el sistema se hará dimensionándola con esta simple ecuación:

$$P = V \cdot I = 8,4V \cdot 120 \text{ mA} = 1 \text{ W}$$

Ecuación 6.1. Cálculo de potencia de placa fotovoltaica

La tensión será la de la batería y la intensidad es el consumo que tiene el sistema. Como la potencia de la batería se debe sobredimensionar se escogerá una batería de 5 W con unas dimensiones de 200x140x12mm.



Ilustración 6.3. Placa solar para el diseño futuro

6.2. Incorporación de sensores

Todavía se puede mejorar más el funcionamiento de esta trampa añadiendo dos nuevos sensores como son:

- **Acelerómetro de 3 ejes ADXL 345:**

Es un sensor micromecanizado (MEMS) capacitivo que detecta la aceleración en los ejes X, Y y Z. También es posible determinar la orientación del sensor, gracias a la acción de la fuerza de la gravedad. Tiene un rango de medición que se puede ajustar entre los valores de 2G, 4G, 8G y 16G. El menor valor da más resolución en los movimientos lentos, mientras que los valores altos tienen más precisión para los movimientos rápidos.

Este sensor detectaría si hay un cambio brusco en cualquiera de las 3 dimensiones, esto se utilizaría para comprobar que la trampa no se ha caído o ha recibido algún golpe o movimiento brusco. Si se produce un cambio de aceleración, el sensor mandaría esta información a través de la shield gsm a la plataforma ThingSpeak y el usuario puede si esto ocurre [15].

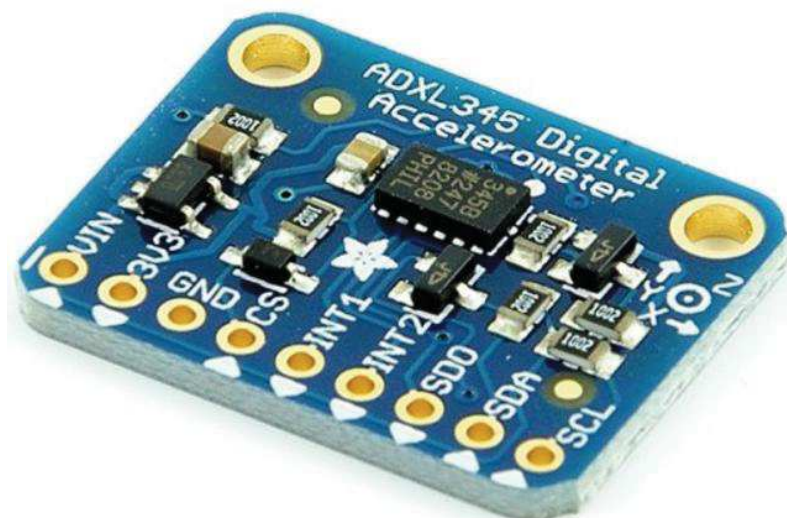


Ilustración 6.4. Acelerómetro de 3 ejes ADXL 345

- **Módulo GPS Ublox NEO-6M:**

El módulo GPS Ublox NEO-6M es un dispositivo de pequeño tamaño, bajo coste y pequeño consumo. Tiene una precisión de posición de 2,5m, 0,1 m/s de velocidad y 0,5° de orientación que son valores aceptables para un sistema de posicionamiento de GPS.

Este módulo se utilizaría para saber dónde se ha instalado la trampa inteligente y llevar a cabo un control de posición de cada trampa, ya que se deben colocar varias trampas por hectárea. La posición GPS también se enviaría a la plataforma web y el usuario sabrá en qué lugar se encuentra la trampa [16].



Ilustración 6.5. Módulo GPS Ublox NEO-6M

8. Bibliografía

- [1] «PLAGAS Y ENFERMEDADES DEL OLIVAR (1ª parte),» [En línea]. Available: http://www.infoagro.com/olivo/plagas_enfermedades_olivo.htm.
- [2] J. d. Andalucía, «Protocolo de campo para el seguimiento del cultivo,» [En línea]. Available: https://www.juntadeandalucia.es/agriculturaypesca/portal/export/sites/default/comun/galerias/galeriaDescargas/minisites/raif/manuales_de_campo/ProtocolosCampos_Olivar.pdf.
- [3] «web-robotica,» [En línea]. Available: <http://www.web-robotica.com/arduino/conceptos-basicos-arduino/como-usar-el-modulo-sensor-de-infrarrojos-ir-fc-51-para-evitar-obstaculos-con-robot-arduinogenuino>.
- [4] Autodesk, «Autodesk,» [En línea]. Available: https://www.autodesk.es/store/products/autocad?AID=11087035&PID=2016554&SID=jkp_CNm2ocvezTQCFdfJGQodyvEOBw&gclsrc=ds&mktvar002=afc_es_nmpi_ppc&gclid=CNm2ocvezTQCFdfJGQodyvEOBw&term=1year&support=advanced.
- [5] Solidwork, «Solidworks,» [En línea]. Available: <http://www.solidworks.es/>.
- [6] Proteus, «Labcenter,» [En línea]. Available: <https://www.labcenter.com/>.
- [7] Arduino, «Arduino.cc,» [En línea]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardMicro>.
- [8] ThingSpeak, «ThingSpeak,» [En línea]. Available: <https://thingspeak.com/>.
- [9] F. d. I. i. Bankinter, de *El internet de las cosas*, 2011.
- [10] Kioskena.net, «Estándar GSM (Sistema global de comunicaciones móviles),» *Telefonía móvil*, 2010.
- [11] Kioskena.net, «Estándar GPRS (Servicio general de paquetes de radio),» *Telefonía móvil*, 2010.
- [12] Arduino, «Arduino.cc,» [En línea]. Available: <https://www.arduino.cc/en/Main/ArduinoGSMShield>.
- [13] Arduino, «Arduino.cc,» [En línea]. Available: <https://www.arduino.cc/en/Tutorial/GSMExamplesXivelyClient>.
- [14] Arduino, «Arduino.cc,» [En línea]. Available:

<https://www.arduino.cc/en/Reference/Millis>.

[15] L. Llamas, «Luis Llamas,» [En línea]. Available: <https://www.luisllamas.es/arduino-acelerometro-adxl345/>.

[16] Cetronic, «Cetronic,» [En línea]. Available: http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/producto/DetalleProducto.jsp?idIdioma=&idTienda=93&codProducto=999334007&cPath=1343&gclid=CjwKEAjwvYPKBRCYr5GLgNCJ_jsSJABqfw7dxqc76Bc_Y0PkLrtV0-v5g_zxUmujA-mFAZK3DHxQhoCfNjw_wcB.