

PECL1. COMPUTACIÓN UBICUA

Pablo Acereda García, David Emanuel Craciunescu, Pablo Martínez Gracia,
and Laura Pérez Medeiro

1. Introducción

El proyecto consistirá en un sistema inteligente para controlar los factores abióticos que afectan al bienestar de las plantas, con el objetivo de mejorar la salud de las mismas.

Para ello se monitorizarán la humedad y temperatura tanto exterior como de la tierra así como la cantidad de luz que reciben. Con estos valores y una serie de parámetros se generarán las correspondientes alarmas que indiquen situaciones de riesgo para la salud de la planta.

2. Contexto

2.1. Situación actual del problema a abordar

Actualmente existen bastantes sistemas que se encargan de controlar el riego de las plantas y los cuales ofrecen la posibilidad de ser controlados mediante dispositivos móviles. Algunos de ellos son Blossom, PlantLink o Edyn. Incluso encontramos maceteros inteligentes, como el que nos ofrece Xiaomi, capaz de regar las plantas automáticamente. También encontramos otros proyectos como GRO, capaz de sugerirnos especies de plantas que podríamos cultivar en función del terreno que disponemos.

Sin embargo, ninguno nos ofrece integración con ningún asistente virtual como pueda ser Alexa, ese será la principal diferencia de nuestro proyecto con los anteriores.

2.2. Situación prevista al final del proyecto

Al finalizar la asignatura se pretende tener un dispositivo capaz de mantener con vida un cultivo de manera automática y el cual pueda ser controlado mediante comandos de voz gracias a Alexa. Para ello, nuestro dispositivo contará con la predicción del tiempo, aviso de posibles plagas, estado del terreno y condiciones óptimas necesarias para el tipo de planta que se posea.

2.3. Beneficiarios del proyecto

Con este proyecto se pretende ayudar a los aficionados de la jardinería que no disponen de una gran cantidad de tiempo para el óptimo cuidado de las plantas, así como a agricultores que necesiten una ayuda extra para minimizar el impacto de situaciones climáticas extremas. Por situaciones climáticas extremas se entienden, por ejemplo, heladas producidas en fechas inesperadas.

3. Alcance del proyecto

4. Descripción de ideas descartadas

5. Tecnología a utilizar

Las decisiones acerca de la tecnología a utilizar en el proyecto se han tomado teniendo en cuenta la experiencia de los distintos miembros que conforman el área de desarrollo, así como el costo que tienen los distintos componentes. Las distintas opciones barajadas para la realización del proyecto han sido:

Hardware

Controladoras

- Arduino: Ofrece una amplia gama de placas, con distintas prestaciones, cuenta con un entorno de programación para la creación de aplicaciones y una gran comunidad que apoya el Desarrollo. El principal inconveniente que se encontraba era su precio, el cual ronda los 20 euros.
- Raspberry: Fue una opción que se estuvo barajando utilizar, ya que ofrece una amplia variedad de lenguajes de programación a utilizar. Sin embargo, fue descartada debido a su gran tamaño.
- ESP32: Ha sido la opción finalmente elegida, ya que se trata de un SoC de muy bajo coste (su precio ronda los 5 euros), cuenta con Bluetooth y es utilizado en multitud de proyectos de IoT. Además, ofrece compatibilidad con Arduino, pudiendo utilizar los amplios recursos desarrollados por esa comunidad.

Sensores

- DHT22:
- DSB18B20:
- LDR:
- YL-69:

Lenguaje de programación

Dado que la microcontroladora elegida ha sido la ESP32, las opciones a la hora de programar disponibles son:

- MicroPython: consiste en un pequeño intérprete de Python, el cual contiene un subconjunto mínimo y optimizado de librerías para que pueda correr en microcontroladores. En un primer momento, fue la idea elegida para utilizar en nuestro proyecto.

Ventajas:

- RELP Interactiva, es decir, un programa que permite leer e interpretar los comandos para evaluarlos e imprimir el resultado sin la necesidad de compilar ni cargar el programa en el microcontrolador.
- Gran cantidad de librerías disponibles
- Extensibilidad. Ofrece la posibilidad de mezclas código que requieran una ejecución más rápida a bajo nivel mediante la extensión de sus funciones.

Desventajas:

- Código más lento y con necesidad de mayor cantidad de memoria, en comparación con C o C++
- Proceso de inicialización del microcontrolador más complejo
- Funcionalidad de ciertas librerías para componentes más limitadas
- RTOS: es un sistema de operación en tiempo real utilizado en sistemas embebidos. Proporciona métodos para múltiples subprocesos o hilos, mutexes, semáforos, temporizadores... Además de soportar las prioridades de los hilos. Por último, cuenta con un modo que reduce el consumo energético (tickless). En FreeRTOS las aplicaciones pueden asignarse de manera estática, mientras que los objetos pueden asignarse dinámicamente con distintos esquemas de asignación de memoria. Pese a contar con una enorme cantidad de documentación, la complejidad y la curva de aprendizaje que supondría el uso de este lenguaje hizo que fuera descartado rápidamente.
- Mongoose OS: framework de desarrollo disponible bajo la licencia de Apache (con una versión community y otra enterprise) y ampliamente utilizado en aplicaciones relacionadas con el IoT. Cuenta con compatibilidad para microcontroladores de bajo consumo, como es el caso del ESP32. Cuenta con un servidor web integrado, soporta programación tanto en C como en JavaScript y cuenta con integración de nubes privadas y públicas (como AWS IoT o Mosquitto).
- IDE Arduino:

6. Metodología de desarrollo
7. Arquitectura de la aplicación
8. Modelo de negocio
9. Planificación temporal y de desarrollo
10. Riesgos del proyecto
11. Plan de contingencias
12. Resumen y conclusiones