Plan de Clase: Visualizador de Estado de la Planta con LEDs

Visualizador de Estado de la Planta con LEDs - Área: Arte

Definir el Problema

El Desafío: Las plantas no pueden hablar, lo que dificulta saber cuándo necesitan agua, luz o si están saludables. ¿Cómo podemos usar la tecnología para 'dar voz' a nuestras plantas y entender mejor sus necesidades? Diseñaremos un dispositivo que visualice el 'humor' o el estado de una planta usando LEDs, basándose en datos de sensores. Esto nos permitirá cuidar mejor nuestras plantas y aprender sobre la interacción entre la electrónica, la programación y la naturaleza.

Espacio para Brainstorming

Reúnanse en grupos y piensen en estas preguntas para generar ideas para su visualizador:
¿Qué necesidades de la planta podemos monitorear? (Ejemplo: humedad del suelo, luz, temperatura).
¿Cómo podemos representar visualmente los diferentes estados de la planta con los LEDs del micro:bit o LEDs externos? (Ejemplo: cara feliz/triste, colores, patrones de luz, animaciones).
¿Cuántos estados diferentes queremos representar? (Ejemplo: muy feliz, feliz, neutral, sedienta, muy sedienta).
¿Qué tipo de sensor utilizaremos para recopilar datos? (Ejemplo: sensor de humedad del suelo, sensor de luz).
¿Qué umbrales estableceremos para cada estado? (Ejemplo: si la humedad es > 700, cara feliz; si es < 300, cara triste).
¿Cómo será la interacción con el usuario o la planta? ¿Necesita un botón para calibrar o resetear?

Dibujen bocetos de sus ideas para la interfaz visual en los LEDs.

Criterio de Diseño

Su proyecto 'Visualizador de Estado de la Planta con LEDs' debe cumplir con los siguientes requisitos y criterios de éxito:

Tecnología: Utilizar un micro:bit (o plataforma similar) y sus LEDs integrados, o conectar LEDs externos.

Entrada de Datos: Recopilar datos de al menos un sensor (preferiblemente un sensor de humedad del suelo).

Representación Visual: Mostrar claramente el estado de la planta mediante un patrón de LEDs distintivo para cada condición (ejemplo: 'feliz' = cara sonriente, 'sedienta' = cara triste o gota de agua).

Cantidad de Estados: El visualizador debe representar al menos tres estados diferentes de la planta (ejemplo: óptimo, necesita atención, crítico).

Programación: El código debe ser funcional, estar bien estructurado y documentado, permitiendo una fácil comprensión y modificación.

Robustez: El sistema debe funcionar de manera confiable y ser capaz de responder a los cambios en los datos del sensor en tiempo real.

Creatividad: Se valorará la originalidad y la estética del diseño visual.

Llevar a la Práctica el Proyecto

Sigan estos pasos para construir su visualizador de estado de la planta:

Paso 1: Configuración del Hardware.

- Conecten el sensor de humedad del suelo al micro:bit (ejemplo: pin 0). Si usan LEDs externos, conéctenlos a los pines apropiados con resistencias.
- Asegúrense de que todas las conexiones sean seguras.

Paso 2: Exploración de Datos del Sensor.

- Escriban un programa simple para leer el valor del sensor y mostrarlo en la consola serie del micro:bit o en los LEDs como un número.
- Experimenten con el sensor en diferentes condiciones (suelo seco, húmedo) para entender el rango de valores.

Paso 3: Definición de Umbrales.

- Basándose en la exploración del Paso 2, definan los umbrales numéricos para cada estado de la planta (ejemplo: seco < 300, normal entre 300 y 700, húmedo > 700).

Paso 4: Programación de la Lógica.

- Utilicen bloques 'si/entonces/sino' (if/else if/else) en MakeCode o estructuras condicionales en Python para verificar el valor del sensor.
- Para cada rango de valores, programen la visualización correspondiente en los LEDs (ejemplo: 'mostrar icono feliz', 'mostrar icono triste', 'mostrar patrón de gota').

Paso 5: Pruebas Iniciales.

- Carguen el código en el micro:bit y prueben el visualizador con el sensor en diferentes condiciones (sumérjanlo en agua, sáquenlo al aire, pónganlo en tierra seca/húmeda).
- Observen si los LEDs cambian correctamente según los estados definidos.

Paso 6: Refinamiento y Mejoras.

- Ajusten los umbrales si es necesario para una mejor precisión.
- Agreguen elementos creativos como animaciones de transición, diferentes colores (si usan LEDs RGB externos) o vibraciones para alertar.

Revisar y Evaluar la Solución

Una vez que tengan su prototipo, realicen las siguientes pruebas y reflexiones:

Funcionalidad:

- ¿El visualizador muestra el estado correcto para cada condición de humedad (o el sensor elegido)?
- ¿Los LEDs son claros y fácilmente interpretables?
- ¿Hay alguna situación en la que el visualizador no funcione como se espera?

Prueba de Casos Límite:

- ¿Qué ocurre cuando el sensor está en el extremo más seco? ¿Y en el más húmedo?
- ¿El sistema maneja bien los cambios rápidos en el sensor?

Experiencia del Usuario:

- ¿Es intuitivo el diseño visual? ¿Cualquiera podría entender el mensaje de la planta?
- ¿Hay algo que confunda o dificulte la lectura?

Robustez del Código:

- ¿El código está libre de errores?
- ¿Es fácil de leer y entender para otra persona?

Identificación de Mejoras:

- ¿Qué harían diferente si tuvieran más tiempo o recursos?
- ¿Qué nuevas características podrían agregar para hacer el visualizador aún mejor?

Reflexión Individual:

- ¿Qué desafíos encontraron durante el desarrollo del proyecto y cómo los superaron?
- ¿Qué aprendieron sobre el funcionamiento de los sensores, la programación condicional y el diseño de interfaces?

Comunicar la Solución

Preparen una presentación para compartir su proyecto, proceso y aprendizajes con la clase o con un público más amplio. Su presentación debe incluir:
Introducción: Presenten el problema que intentaron resolver y cómo su visualizador aborda ese desafío.
Diseño: Muestren sus bocetos iniciales y expliquen las decisiones de diseño clave (por qué eligieron ciertas representaciones visuales, qué sensor usaron).
Demostración: Realicen una demostración en vivo de su visualizador. Muestren cómo reacciona a diferentes condiciones de humedad (o el sensor elegido).
Código: Expliquen brevemente las partes más importantes de su código, especialmente cómo configuraron los umbrales y la lógica condicional.
Hallazgos y Desafíos: Compartan lo que aprendieron, qué funcionó bien y qué desafíos encontraron durante el proceso de diseño y construcción.
Preguntas y Futuro: Inviten a preguntas y compartan ideas para futuras mejoras o extensiones de su proyecto.

Anotaciones

Evaluación