Final Project ECE36200
Smart-Self Sustaining Garden

By: Anthony Orrell (aeorrell@iu.edu)

Objective:
Write an assembly program to simulate a smart irrigation and environment control system for a self-sustaining home garden. The system should (1) generate a random layout for the user's garden, (2) allow users to adjust moisture levels and temperature, (3) control grow lights for plant growth, (4) manage an automated watering system, and (5) display the current time and date. A safety feature is crucial for the home garden, incorporating a fire detection system that can cut off power and secure the area. The goal is to develop a user-friendly and reliable control interface for the garden.

Startup/Home Screen: A screen for the user to initialize the farming system and implement password. It can also be used as a home screen or menu for the user to decide to update the settings (requirements shown below) or get a new plot layout.
Uses: LCD, Keypad.

- Allow the user to set up the password.
- Welcome the user to the program.
- Generate the "plot layout" and communicate that to the user. (There are 8 available plots, which the user will turn on and off with the switches).
    - HINT: For random generation think about the modulo function in standard programming and how it works.

Watering System: Use a servo motor to simulate the watering mechanism. When activated, it simulates watering the garden.
Uses: Push button, servo motor, buzzer, LCD, LEDs

- Activating the push button starts the watering process.
- A buzzer plays a melody to indicate that watering has started.
- The LCD displays "Watering in Progress" during the operation.
- An LED indicator lights up during watering.
- The servo motor should mimic the motion of watering.
- The motor will automatically shut off after a set amount of time.

Air Circulation: Simulate air circulation using a fan controlled by a motor.
Uses: Motor, potentiometer, LCD

- The fan speed is adjusted using a potentiometer, with the speed proportionate to the encoder's rotation.
- The motor should not operate when the potentiometer is at its minimum position and should increase speed progressively as turned.
- "Temperature" should be displayed on the LCD screen; we will not be picky about the temperature ranges you choose to use but try to be as realistic as possible

Digital Clock and System Settings: Include a digital clock displaying the current date and time.
Uses: Keypad, LCD

- Use the keypad for input; pressing a specific key brings up the settings menu.
- Implement navigation through the settings using designated keys for scrolling and selection.
- Allow password-protected access to system settings, including date, time, and password changes, with validation checks to prevent invalid entries.
- Password will also be used on IRQ lockdown (see safety feature).

Safety Feature: Implement an emergency shutdown mechanism, triggered by the IRQ interrupt, simulating a pest alarm.
Uses: IRQ Interrupt, LEDs, Buzzer, LCD, Keypad

- In an emergency, the irrigation system should power down, stopping all active processes, and the LEDs should power on to max to help detect the pest.
- The LCD prompts for a pin to reactivate the system, with incorrect attempts to maintain the shutdown.

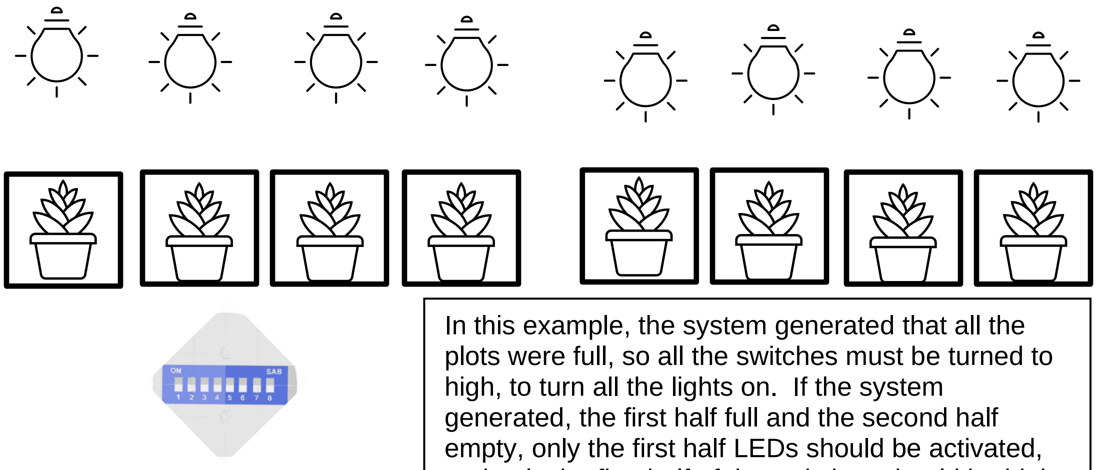
Grow Light Control: Toggle grow lights using switches, a light for each plant, allowing the user to conserve energy on empty plots.
Uses: Switches, LEDs

- Every switch and LED should be used, each switch will correspond to an LED, and based on the layout of the plots generated on startup of a new garden the user should turn on/off the correct LEDs to progress. (i.e Switch0 and LED0 correspond to Plot0, Switch1 and LED1 correspond to Plot1, and so on)

User Experience: The overall experience for the user should be almost like a game, once the grow lights are turned on correctly and the irrigation system has run, the system should generate a new plot layout. Feel free to play with this (a lot of opportunities for play with your additional features), by including things like lose conditions or total yield tracking.

Extra Features: Include additional functionality to enhance the project.

- Implement creative features that align with the system's functionality, with complexity contributing to grading.



In this example, the system generated that all the plots were full, so all the switches must be turned to high, to turn all the lights on. If the system generated, the first half full and the second half empty, only the first half LEDs should be activated, and only the first half of the switches should be high.

Universal Requirements

1. The user must be able to turn on/off any row of lights at any time, not in an emergency (i.e. even when in a menu or animation state). Any changes to the switches must be indicated by the appropriate LED.
2. Each song should be unique, repeating in a loop for the duration of the given event. “Songs” are truly just patterns of a handful of different notes. The speaker should be able to play two distinct songs. One should play when the sprinkler is sprinkling, and one during the alarm.
3. The user must be able to adjust the fan speed at any time not in an emergency, and the DC motor should automatically adjust accordingly.
4. The alarm should have the ability to be activated at any time (i.e. even when in a menu).
5. The overall layout of your system should be easy to understand and make sense. The user should be able to operate the system with little to no training or explanation. If you are unsure if the layout of your system makes sense, ask one of your TAs or fellow students to try to move through your system.
6. No delay loops are allowed; you must utilize the Real Time Interrupt (RTI). DELAY LOOPS ARE ONLY ALLOWED FOR HEX KEYPAD DE-BOUNCE. HINT: If you are having trouble with the switches bouncing, maybe try only scanning them every X many RTIs.

Note:
You are encouraged to be creative and make this project your own. You can make reasonable assumptions in the development of this project, but keep in mind that the assumptions must make sense to the user (and to the Lab TAs) so be ready to explain your choices. This is not an excuse to cut corners.

For your project, 50% of the grade will be based on extras that you will create on your own.
Extra Feature Suggestions:

Gold Tier:

- Implement LED dimming with PWM based on the time of day.
- Track yield over multiple different farming seasons, this should include ways to lose crops potentially by not turning the lights on in time, or by having the fans set too high or too low, a lot of different ways you could run with this.

Silver Tier:

- Add temperature display toggling between Celsius and Fahrenheit, or offer a time format choice (12-hour or 24-hour) on the digital clock.
- Set a “goal temp”, requiring the user to set the potentiometer to a certain value before it progresses to the next plot layout.

Bronze Tier:

- Mask password entry so strangers cannot see what was input on the lcd screen.
- Implement the timer for the “water pump” to be based on how many plots are being watered (i.e. If you had this configuration 1100 0010, you would do 3x the standard length of time used in your system). Display either time left or plots left on the LCD screen.