SmartGarden
IOT Project Presentation

SAM LEES & MATT LINDLY

Project Overview

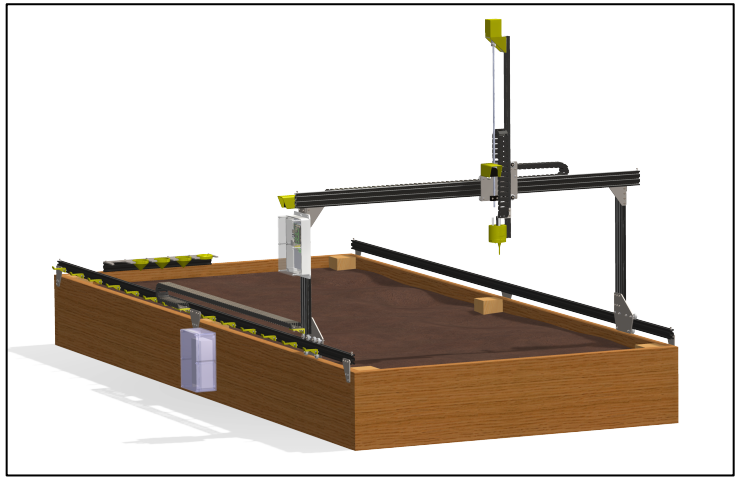
Problem

- Many believe succulents are easy to grow, but watering is significant to plant health
- Too much water causes root rot and drowning
- Too little water causes dehydration and starvation
- Watering need are different than for other house plants
- Easy to neglect succulents
- Easy to forget about

Background

Examples of other automated watering products:

- GreenIQ Smart Garden
- Edyn Smart Garden Sensor
- FarmBot

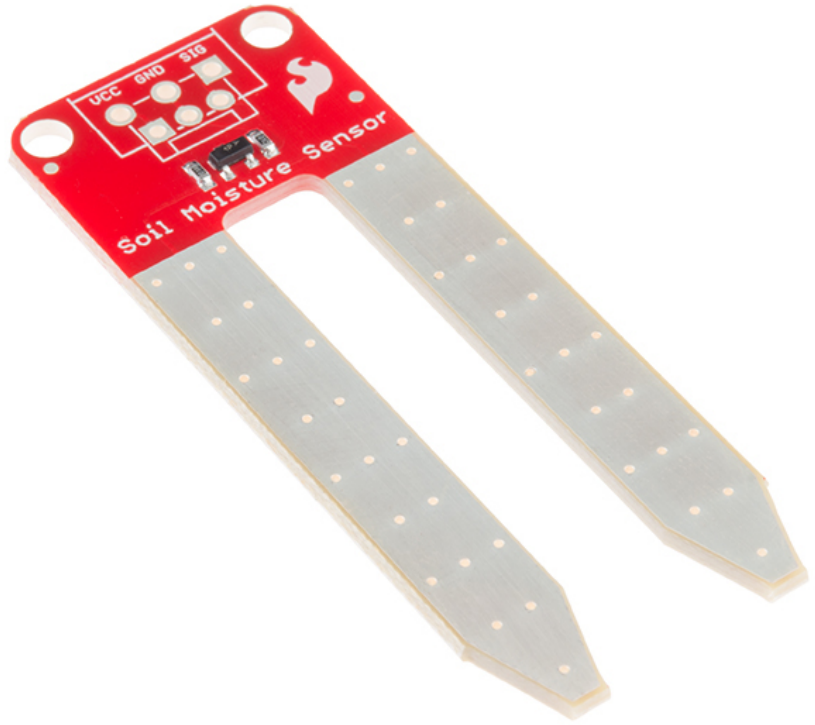


Issues with above products:

- Not suitable for small-scale indoor use
- Prohibitive cost

Solution

- Use Raspberry Pi's at the Edge
- Interface one or more soil moisture sensors per Pi
- Host a collection server in the Cloud
- Provide a web interface for real-time monitoring and historical data
- Leverage Twilio's SMS API for reliable text-message-based reminders



sparkfun.com/products/13322

Project Requirements

Scope

This project…

- is a proof-of-concept only.
- is intended to demonstrate an understanding of IOT systems.
- is being conducted as part of CPE 548-01 at Cal Poly, SLO in Fall 2016.
- is not intended to produce a consumer-ready product.

Feature Requirements

- Raspberry Pi clients must relay moisture measurements back to the server
- The server must host a web interface to display “real-time” moisture levels and a summary of past data
- The server must also send text message reminders of critically low moisture levels

Limitations

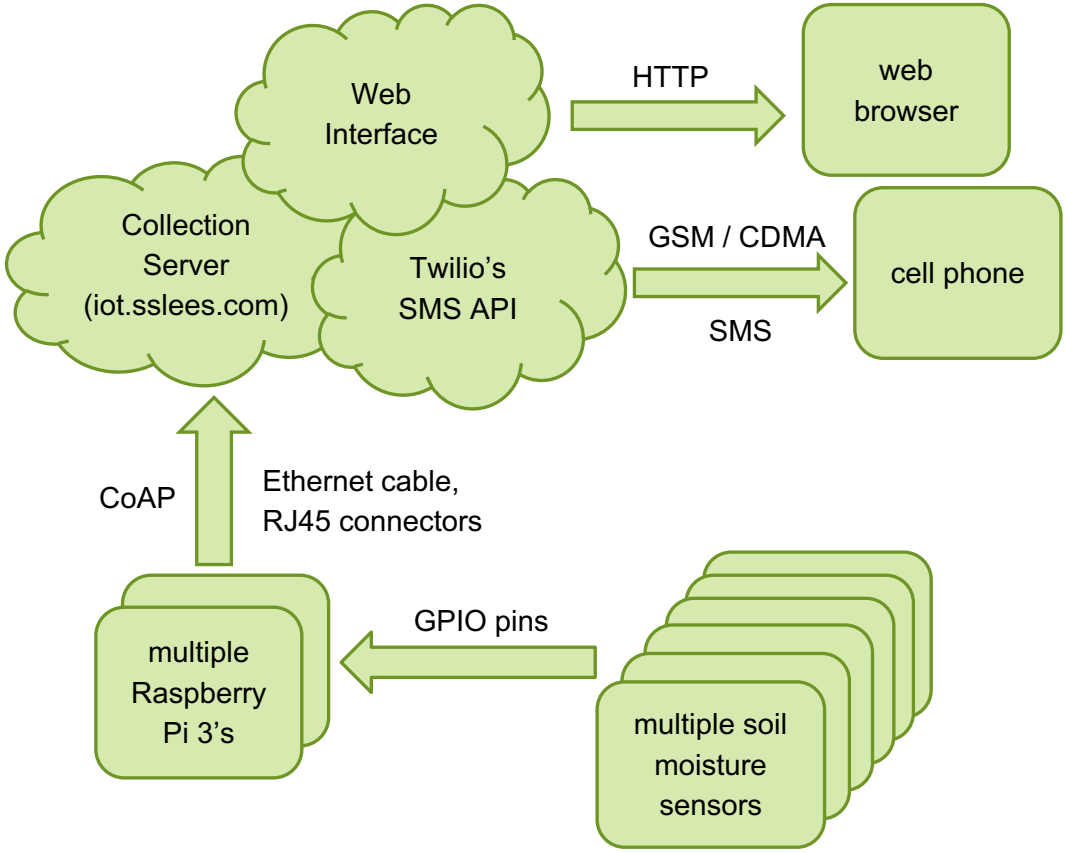
- The UDP packets will not be encrypted
- The server will not authenticate incoming UDP packets
- The server will not validate the data of incoming UDP packets
- The server will not secure the data provided by the web-based interface
- The server will only secure communication between itself and Twilio

Project Specifications

Architecture

- Raspberry Pi's will run a CoAP client
  - This will be code reused from Lab 2.2
- Client will send data as UDP CoAP packet
- Data will be sent every 30 seconds
- Collection server will be hosted at iot.sslees.com
  - This will be code reused from Lab 2.2
- Collection server will run an Apache 2 HTTP server for the web interface
- Collection server will interface with Twilio's SMS API
- Raw data will be stored in SQLite format on the collection server as voltage values

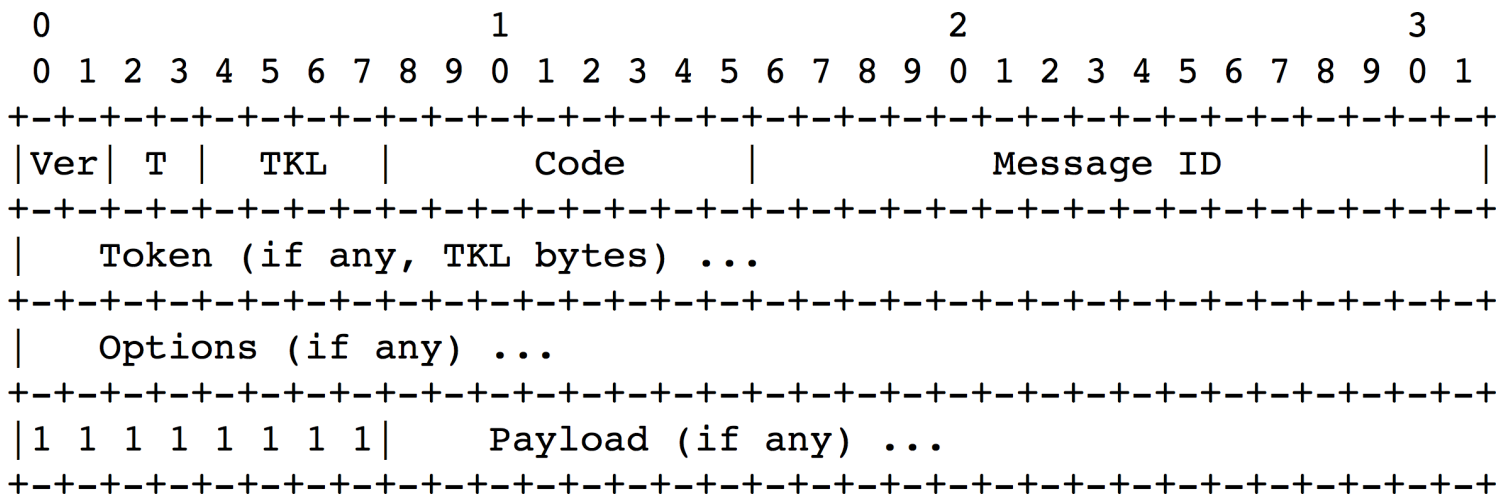
Architecture Diagram



UDP Packet Format

CoAP Payload:

- (1) 64-bit Timestamp
- (1) 64-bit double
  - Voltage value (0 – 5 V)



CoAP Message Format
tools.ietf.org/html/rfc7252#section-3