CS 462 Winter Progress Report

March 20, 2018

Project LOOM

Prepared for

Kevin McGrath
Kirsten Winters

Prepared by

Group 36

William Selbie

Abstract

A progress report for Senior Capstone Group 36 regarding the current state of the project and William Selbie's actions during the Winter term and intentions for the Spring. It also evaluates the contributions to the project from each member, as well as the group dynamic as a whole.

Contents

List of Listings

1 Project Purpose and Goals

With Project LOOM, we aim to create an open-source, plug-and-play, suite of modular building blocks, the extensible and easy programmability of which expands the demographic of people capable of implementing Internet of Things solutions. For users with limited technical expertise to create complex systems, we aim to build a system that abstracts out the more technical details, allowing them to focus on their system more than the implementation of the modules. The system should also be usable by higher level students and experts by allowing them to modify or write their own firmware, and create new modules. Project LOOM will be developed for university faculty demos of functionality.

2 Current Project Status

Project LOOM has improved greatly over the course of Winter term. The bulk of the progress being made has been in fleshing out the support for different sensors, actuators, and wireless communication platforms. The servo, relay, and NeoPixel now all have robust Max patches that give the user great control over the setup of each actuator. The devices start as access points, but if sent the credentials for a wireless network will then connect to that network, allowing any device on the network capable of sending OSCBundles to communicate with them. The Max patches also limit the commands that can be sent to the actuators to ones that are possible, for instance a servo can only rotate up to 170 before performance declines, so the Max patch does not allow the user to specify more than 170 degrees of rotation. While WiFi is what is currently supported by Max patches, both LoRa and nRF have been successfully used to transmit data that can be parsed into OSCBundles. Beyond this, LoRa has been tested in the context of multiple nodes with a single Ethernet-enabled hub that updates a Google Spreadsheet with the data sent to it by the nodes. The Decagon soil moisture sensor is also supported now, so it can gather data and format it into OSCBundles before transmitting it to a hub that makes use of the data.

The functionality of the project in its current state was demonstrated at two open-houses, one for the OPEnS Lab in which all OPEnS Lab projects were being demoed, and the other specifically for faculty interested in IoT wherein only Project LOOM was demoed.

3 Retrospective

3.1 Weekly Summary

Week One

Summary - Scheduled CS team meetings from 3-4pm on Fridays. Scheduled LOOM team meetings from 4-5pm on Fridays.

Week Two

Summary - Decagon code was added to GitHub. WiFi was integrated into Decagon code. nRF was shown to work on 32u4s.

Problems - nRF sketches have not historically worked on M0s, but it would be really nice if they did now as an M0 could have WiFi and nRF capabilities and therefore act as a middleman between networks.
Solution - A fork of the previously used library that did not support M0s now supports M0s.

Week Three

Summary - Met with both teams. nRF was tested successfully on M0s. Arduino Libraries were updated. Decagon code was refactored and made to be leaner.

Week Four

Summary - Continued to clean up the Decagon code. Kenny and I worked together to make sure that nRF enabled 32u4s and M0s could communicate with each other, as could LoRa enabled 32u4s and M0s.

Week Five

Summary - We merged the CS team meeting and the ECE team meeting because Chet broke his leg in a snowboarding accident and will not be able to make it to meetings for a while, so Tom DeBell is running them now. OSCBundle receiving capability was added to the Decagon sensor, as well as the ability for the user to set the transmission frequency. Kenny began working on the OSCBundle translator and I began writing basic node and hub test sketches.

Week Six

Summary - Kenny completed the translator, this allows nRF and LoRa (which only send byte streams) to transmit strings that can then be parsed into OSCBundles on the receiving end. I incorporated the translator into my basic node and hub sketches.

Week Seven

Summary - Kenny and I used the translator and sketches we had been working on to have three 32u4 nodes and an M0 node transmit information simultaneously to an M0 hub. I started looking into how to use the Ethernet Featherwing.

Week Eight

Summary - Altered the LoRa Node and Hub sketches to function more generically and work better as templates. Successfully tested using the Ethernet Featherwing to send GET requests.

Week Nine

Summary - Went to the OPEnS Lab demo as well as the IoT faculty demo to show off the current capabilities of Project LOOM. Altered the translator to be more readable and dynamic.

Week Ten

Summary - Altered the architecture of the node and hub sketches as well as the PushingBox scenario. Succesfully tested the LoRa network with multiple nodes all transmitting data to the hub, which in turn added this data to a Google Spreadsheet by sending a GET request to PushingBox. The Google Apps Script that formats the PushingBox data is shown below.

Listing 1: Google Sheets Script

```
function doGet(e) {
  var result = 'Ok'; // assume success

  if (e.parameter == undefined) {
    result = 'No Parameters';
  }
  else {
    var id = '1kkD8Tql4uJ7usnIM5TLM35XSYinU2_JmILoUw8eEjg4';
    var data_dict = getDataDict(e.parameter);

    var column_set = Object.keys(data_dict);
    var sheet_id = 'Device' + stripQuotes(data_dict['IDtag']);
    var spreadsheet = SpreadsheetApp.openById(id);
    var sheet_list = spreadsheet.getSheets();
    var sheet = spreadsheet.getSheetByName(sheet_id);

    if(sheet == undefined) {
      sheet = spreadsheet.insertSheet(sheet_id);
      sheet_list = spreadsheet.getSheets();
      sheet.appendRow(column_set);
      setSheetColumnList(sheet.getName(), column_set);
    }

    var new_range;
    var new_row = sheet.getLastRow() + 1;
    var row_data = [];

    if(!arraysEqual(column_set, getSheetColumnList(sheet.getName()))) {
      new_row += 1;
      new_range = sheet.getRange(new_row, 1, 1, column_set.length);
      new_range.setValues([column_set]);
      new_row += 1;
      setSheetColumnList(sheet.getName(), column_set);
    }

    for(i = 0; i < column_set.length; i++)
      row_data.push(data_dict[column_set[i]]);

    new_range = sheet.getRange(new_row, 1, 1, row_data.length);
    new_range.setValues([row_data]);
  }

  // Return result of operation
  return ContentService.createTextOutput(result);
}
```

3.2 Retrospective

Positives	Deltas	Actions
Project LOOM now has working sketches with each of three wireless communication platforms: WiFi, LoRa, and nRF.	Both LoRa and nRF need to become more fully developed to the point that template sketches exist, allowing for easy creation of new LoRa and nRF capable sketches.	The focus of early Spring term will be on developing sketches with LoRa and nRF capability. Chet is also planning on deploying a network involving several Decagon sensors that transmit to a hub connected to the internet early in Spring term so this goal will be a serious focus during Spring term.
The MaxMSP sketches that Luke has written are now much more robust and clean, allowing for easy testing and demonstration of currently supported LOOM modules.	We will need to keep expanding the modules supported by Project LOOM as well as create tutorials or guides of some kind to make it easier to learn how to use the available modules.	Chet wants us to make demo videos showing how to use MaxMSP to control the modules.
A large portion of the desired capabilities of Project LOOM as well as the requirements for the end product for capstone have been met.	Now that a lot of the functionality is in place, it is important to refactor code to make it leaner and more readable as well as thoroughly document all the code we have.	We will systematically go through the code base Spring term to standardize and document all of the sketches and remove erroneous content.
	We need to start developing the dynamic build system that uses preprocessor directives to flash the device with the minimum functionality needed to achieve the user's goals for the device.	Trevor and Luke have started testing out ways to achieve this using Arduino Builder, which is a tool used for compiling Arduino sketches via the command line. This will continue to be worked upon during Spring term.

4 Conclusion

Overall, the project is progressing nicely and Chet has mentioned on multiple occasions how glad he is that all of his goals are being met and that we are actively consulting him to make sure his input continues to help form the end product. We have achieved much of the desired functionality of the end product, so our goal for Spring term will be making it leaner and more modular. This means we will be turning a lot of generic functionality, such as WiFi or LoRa

communication, into dynamically included blocks decided by configuration files when the sketch is compiled. Outside of this, a heavy focus of the project will be thoroughly documenting existing code and creating tutorials to teach users how to make use of Project LOOM.