

# COP 5615: Distributed Operating Systems Principles

## Internet of Things Support in Xinu

Fall 2016

### Term Project Report

# Group 11

[manmdevineni@ufl.edu](mailto:manmdevineni@ufl.edu), [kchokshi@ufl.edu](mailto:kchokshi@ufl.edu), [lalanimaulik@gmail.com](mailto:lalanimaulik@gmail.com), [rathi@ufl.edu](mailto:rathi@ufl.edu), [bgautam@ufl.edu](mailto:bgautam@ufl.edu)

---

## 1. Describe your project using this table

Part	Indicate Completeness (give a no. from 1-10), followed by Description
Xinu I/O Interface design	<u>Completeness</u> – 10 <u>Description</u> – We have used three i/o devices i.e LED, Seven segment display(SSD) and temperature sensor. LED is used as an emergency light. SSD is used to display message sent from the remote location. And temperature sensor is used to indicate ambient temperature on the terminal.
IoT-specific concerns your design addressed, including but not limited to Energy	<u>Completeness</u> – 10 <u>Description</u> - Our design is energy efficient because it is not polling based. Task is performed by the I/O devices only when user gives input. Using the Web UI a user can access any device connected to EDGE. This helps user from around the world access the devices.
Xinu I/O Interface implementation and testing	<u>Completeness</u> – 10 <u>Description</u> - First, we tested and implemented individual i/o devices such as LED, SSD and

	<p>temperature sensor and then integrated them.</p> <p>In our final design output pin of temperature sensor is connected to ain0 pin configured in input mode and LED at GPIO 14 pin configured in Input/Output mode. Seven Segment Display has been connected in output mode to pins 31,59,15,51,60,50,48.</p>
Design of IoT Description Language, Language processing and code generation	<p><b>Indicate:</b> XML</p> <p><b>Source:</b> We used <code>xml.etree.ElementTree</code> package to access xml file in Python.</p> <p><b>Design:</b></p> <p>Using XML as the Device Description Language (DDL) we wrote the description of the device which is then processed using python to generate code from parser.</p>
Implementation and testing of IoT Description Language, Language processing and code generation	<p><u>Completeness</u> – 10</p> <p><u>Description</u> – First we wrote DDL for individual devices. We have pin modes and pin number hardcoded in our DDL.</p> <p>Using python we generated lower level code from the parser of individual devices which we tested. After testing all the i/o devices individually, we integrated the entire code.</p>
Implementation and testing of overall on-board driver code (upper- and lower-level drivers, including generated code)	<p><u>Completeness</u> – 10</p> <p><u>Description</u> – We merged the code for device drivers and include code of the parser to finally test the entire project. Using router, we could send UDP packets from a Web Based IoT App. to Xinu.</p>
Did you use the same existing device driver structure and mechanisms in Xinu?	Yes

Approximate % driver code generated with respect to overall on-board driver code	70 %
Which device externalization abstraction have you chosen (which existing technology or any new ideas)? You may, or may not explain the reason for your choice.	<p><u>Completeness</u> – 10</p> <p><u>Description</u> -We are using two laptops and a BBB. BBB is connected to the network and one of the laptop(L1). L1 acts as an edge device. Laptop (L2) is for the user to interact with the IoT things through the internet.</p>
How, where, and when do you specify the edge and cloud addresses of the device? Explain how device configuration and initialization are done including device externalization.	<p><u>Completeness</u> – 10</p> <p><u>Description</u> -Edge and cloud addresses of the devices are specified in hexadecimal, hardcoded using socket programming in server and jsp files. Addresses assigned during runtime.</p>
Give the details of the externalization abstractions design.	<p><u>Completeness</u> – 10</p> <p><u>Description</u> -Using web based IoT device demo app on a device i.e laptop we can access i/o devices located at the remote location.</p>
Describe the implementation of the abstractions (how they connect to the actual device), and discuss any IoT-specific concern (including energy) that may have been addressed by your implementation.	<p><u>Completeness</u> – 10</p> <p><u>Description</u> -User input is taken using UI on the laptop. The input is sent using UDP to another laptop to which the BBB is connected. Input giving by the user is processed by Xinu and then the i/o device perform the task accordingly.</p>
Describe your on-board IoT devices Demo App.	<p><b>Devices:</b> Laptop1 and BBB</p> <p><b>App:</b> Devices connected to Xinu behave in different ways based on the message sent to Xinu.</p> <p>EX –</p> <p>For</p> <ul style="list-style-type: none"> <li>Msg 0 Action LED status</li> <li>Msg 1 Action SSD status</li> <li>Msg 2 Action TMP status</li> <li>Msg 3 Action LED read</li> <li>Msg 4 Action LED ON</li> <li>Msg 5 Action LED OFF</li> <li>Msg 6 Action SSD write</li> <li>Msg 7 Action TMP read</li> </ul>

Describe your web-based IoT devices Demo App.	<u>Completeness</u> – 10 <u>Description</u> – Our web based IoT devices Demo App. consists of a Java application which has several buttons to get input from the user. UDP packet is sent from one laptop to another laptop to which Xinu is connected whenever user gives input.
---	--

## 2. Challenges

Challenges your group faced. What was the most time consuming parts of the project? what piece(s) would you have really liked to have us provide to you so the total effort is more manageable (again, if any)?

- We faced challenges at every stage of the project from conceptualizing to implementation.
- Most time-consuming part was the communication using UDP. We found it difficult to send packets to another laptop and that took a lot of effort to make it work.
- Another time-consuming part was clock setting for ADC and figuring out how to use base register to access GPIO.
- Details regarding whether to use xinu with applying patch or without patch would have simplified our job.

## 3. Overall Experience

Overall experience. Describe your overall experience good or bad.

- Overall experience was good which provided us with good amount of learning about sensors and communication using UDP packets.
- This project also provided insight on how the operating system works and how to design and implement drivers for IoT.

## 4. Effort Distribution

Report only if effort was considered by any member of the group to not be even. In this case a table showing the names, ID's, and percentage of effort should be provided.