SIMPLE IoT Application

Project 4

By: Ahmed Afify

Outline

- Project Overview
  - Set of Requirements
- Hardware
- System Architecture
- System Description
- Implementation Approach
- Key Code segments
- Progress Report
- Demo

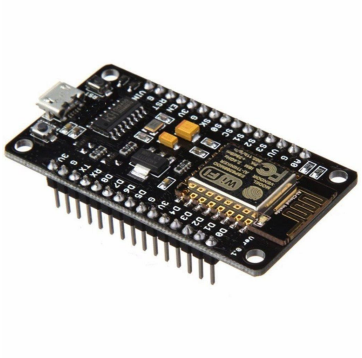
Project Overview - Requirements

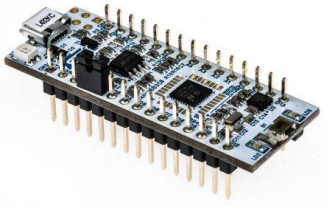
- Develop a simple IoT application that allows users to communicate with a microcontroller to retrieve date/time and control a led status through a web interface

Hardware

Hardware Required

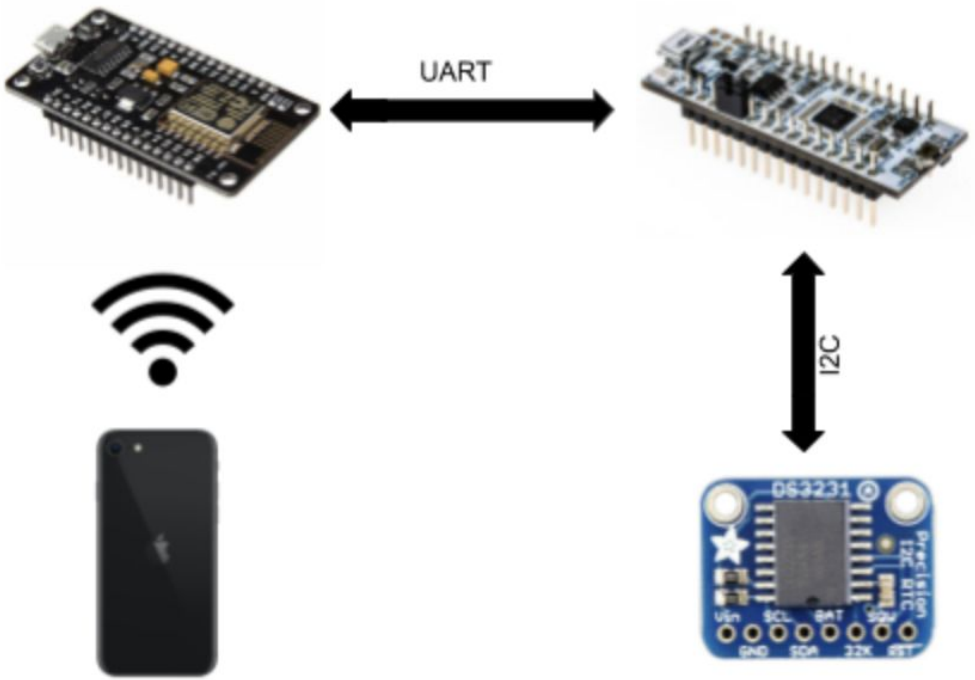
- Nucleo-32 STM32L432KC
- RTC I2C
- NodeMcu
- Laptop/Smartphone/Tablet..etc





System Architecture

System Architecture



System Description

1. The Smartphone/Laptop connects to the Nodemcu that acts as an access point over wifi.
2. The Smartphone/Laptop issues Http requests that the Nodemcu receives.
3. The Nodemcu forwards the request to the Nucleo over UART communication
4. The Nucleo either turns on/off the led or gets the time for the RTC using I2C communication

Implementation

Implementation Approach

Modular Design

- Develop a server on the Nodemcu
  - Develop the API commands to
    - Control LED status on Nodemcu
    - Retrieve dummy date/time data
- Utilize UART on Nodemcu
  - Integrate API commands to transmit and receive data to/from TeraTerm
- Nucleo-32 & RTC integration
  - Retrieve date/time from RTC and output to TeraTerm
  - Receive commands from TeraTerm to retrieve date time
- Integrate Subsystems (Laptop <--> NodeMcu <--> Nucleo <--> RTC)

Key Code Segments

API/Handlers- NodeMcu

```
server.on("/on",turnon);
 server.on("/off",turnoff);
 server.on("/time",getTime);
 server.on("/",turnoff);
```

```
void turnon(){
  s.write("o");
  state=onn;
  pinMode(2,HIGH);
 server.send(200,
"text/html", SendHTML(true));
 Serial.print("Led On\n");
 }
```

```
void turnoff(){
  s.write("f");
  state=off;
  pinMode(2,LOW);
  server.send(200,
"text/html",
SendHTML(false));
  Serial.print("Led
Off\n");
}
```

```
void getTime(){
 Serial.print("Getting time\n");
 timee="";
 bool received=false;
 prnt();
   while (!received)
 {if(s.available()>0){
   data=s.read();
   timee=timee+data;
   if(data=='\n')
   received=true;
   Serial.println(data);
   }
  }
     pinMode(2,LOW);
  server.send(200,
"text/html",SendHTML(state==onn));
```

UART Rx Callback

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
      HAL_UART_Receive_IT(&huart2, &x, 1);
      if(x=='t' || x=='T') // get time
        getTime();
      else if( x=='o' || x=='o') // turn on led
      HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3,GPIO_PIN_SET);
      else if(x=='f' || x=='F') // turn off led
      HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3,GPIO_PIN_RESET);
}
```

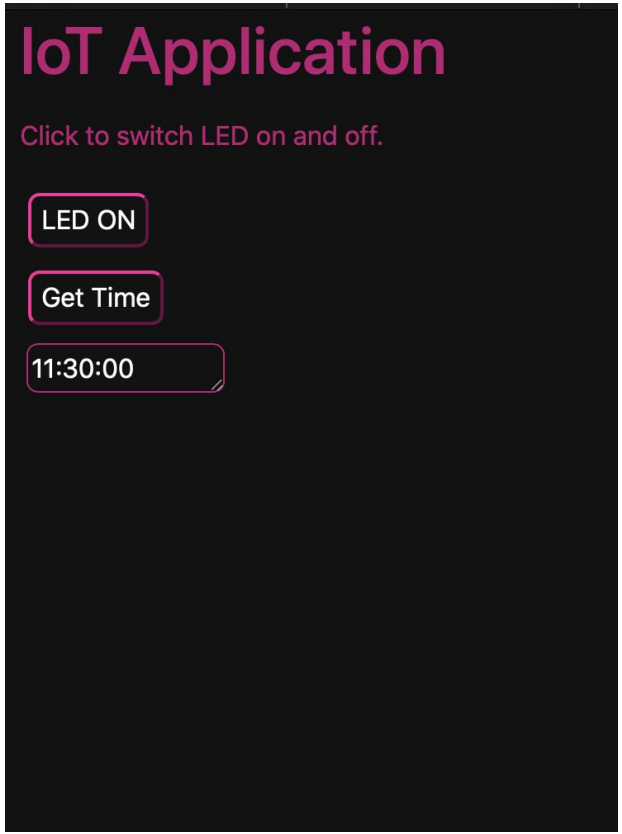
Progress & Results

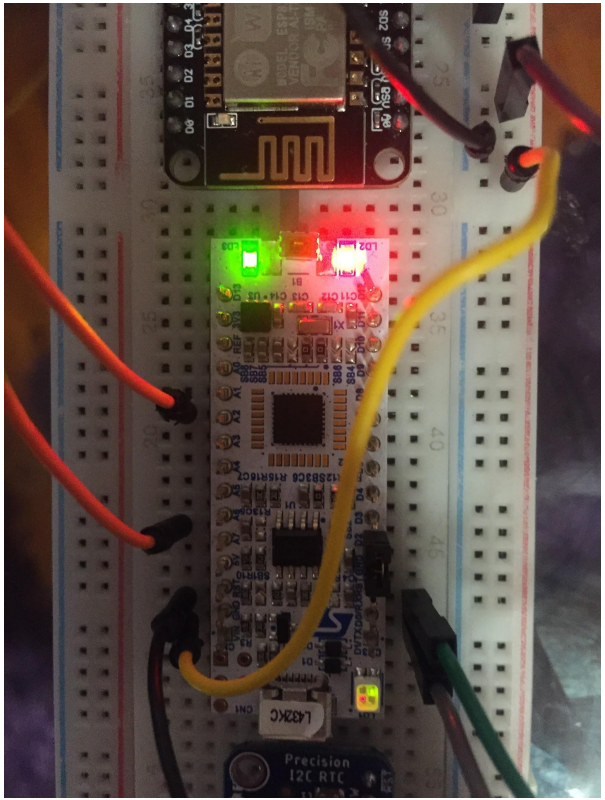
Overall Progress

Completed

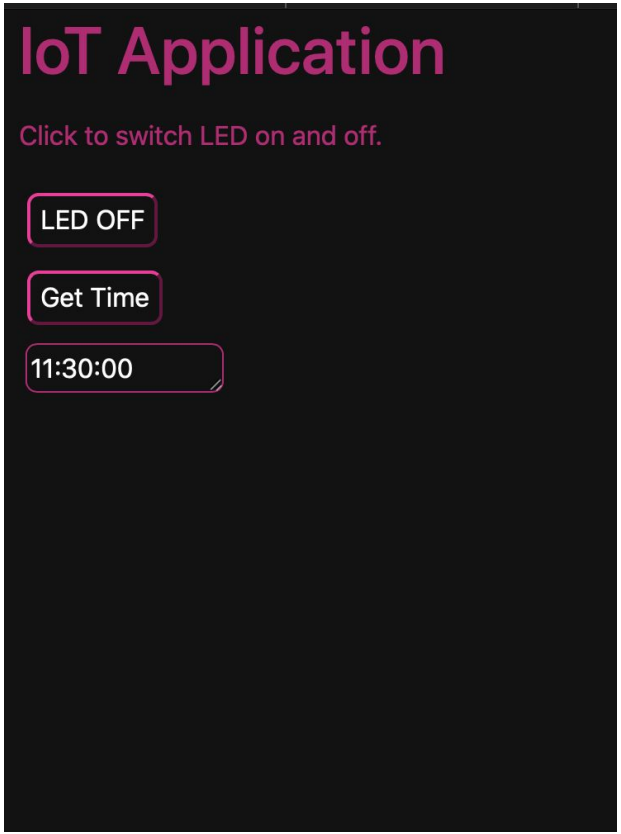
- ✓ Develop a server on the Nodemcu
- ✓ Utilize UART on Nodemcu
- ✓ Nucleo-32 & RTC integration
- ✓ Retrieve date/time from RTC and output to TeraTerm
- ✓ Receive commands from TeraTerm to retrieve date time
- ✓ Integrate Subsystems
- ✓ (Laptop <--> NodeMcu<--> Nucleo <--> RTC)
- ✓ Use Interrupts Instead of pulling
- ✓ Implementation of simple GUI

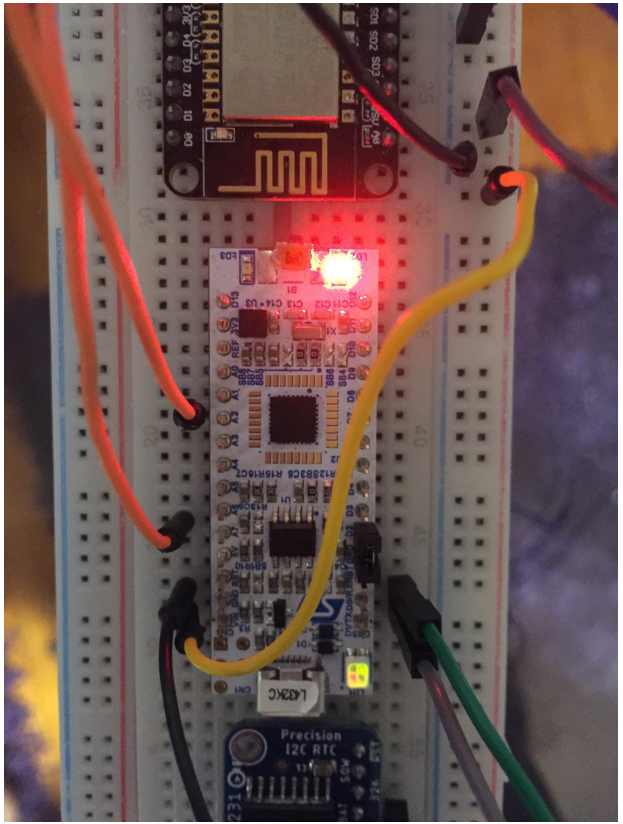
Results - Connecting to Server & Turning on Led



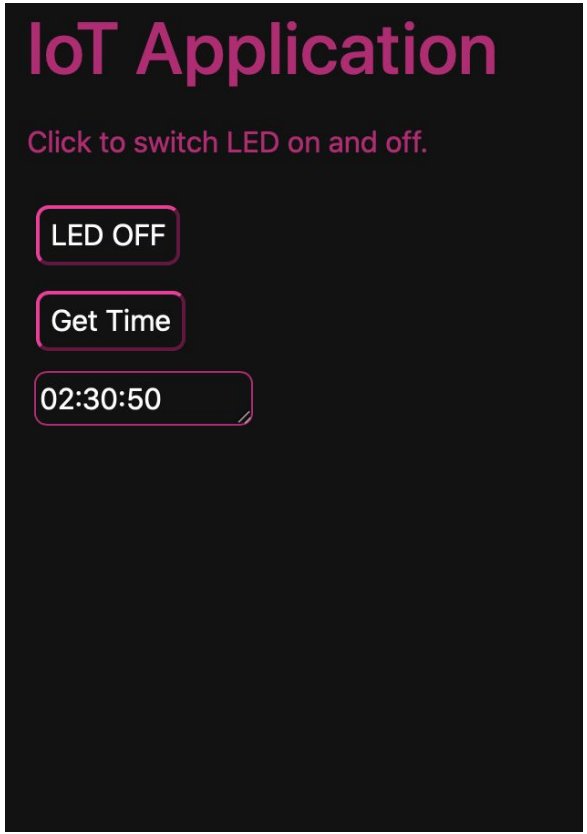


Results - Turning off Led





Results - Retrieving Time



DEMO

Thank you!