# Solaire

Team EE - GSM Controlled LED

CSM117 Project Report

Team Members:

Rex Taymany — 404841213

Shayan Darian — 904679011

Nick Marks — 904844704

**Introduction:**

After conducting the Special Wireless Experiments, we thought it would be useful and practical to incorporate wireless technology to relay input information to our circuits rather than having to manually push a button on the circuit itself. By creating a mobile app and incorporating a GSM module, we hoped to learn more about server and network architecture and the processes that encompass each step.

**Functionality:**

Our app (Solaire) is an iOS based mobile app that allows users to wirelessly control an circuit elements, in this case an LED, through the use of a GSM module connected to an Arduino micro-controller. The use of a GSM over WiFi or Bluetooth allows for more flexibility in distance and location when controlling the circuit.

**How to run the project:**

The Xcode project and its code should be exported either as a simulation of a device in Xcode or it can be exported to an iOS based device. The code related to the Arduino and GSM device should be installed to the Arduino UNO through the Arduino IDE after all the necessary connections are established. When all components are properly loaded, the app should enable the user to control the LED.

**Wireless Technologies:**

- GSM
- Ting (ISP)
- dweet.io

**Software and Hardware:**

- Xcode, Swift, Arduino IDE, Adafruit FONA 800, Antennae, Ting 2G SIM card, Arduino UNO, LED, Lithium Batter, Resistor, Breadboard

**GSM:**

Global System for Mobile Communication, or GSM for short, is a digital mobile telephony system that is widely used in Europe and other parts of the world. GSM uses a variation of time division multiple access (TDMA) and is the most widely used of the three digital wireless telephony technologies, operating at either the 900 MHz or 1800 MHz frequency band. We chose to use GSM as our choice wireless technology as opposed to WiFi or Bluetooth since we wanted to be able to operate and monitor our circuit from anywhere, whether it be from home or a moving car, over long distances.

**Ting (ISP)**

In order for our GSM module to work reliably as intended, we used Ting as our ISP to provide the coverage we needed to access our circuit from any remote location. The provided SIM card in installed onboard the GSM module and operates any way a typical phone would operate with standard message rates and fees.

**dweet.io**

We incorporated the web based RESTful API that dweet.io employs in order to communicate with and monitor our circuit. This allows for more flexibility as the JSON requests can be easily coded on the front end through built in function calls and libraries provided in the Xcode and Swift environment while Adafruit GSM libraries included in the Arduino IDE would be able to handle the reception of these requests.

**Implementation:**

The app's front end is coded in Swift and utilizes Xcode's mobile development platform to create an interface for user input in the form of a switch and button. The app's code would relay the user input to a web based RESTful API (dweet.io) in the form of JSON requests. These requests would be picked up and communicated by the GSM module to the serial monitor in the Arduino. By reading the incoming stream provided by the GSM module, the Arduino would be able to match strings to the correct functionality intended for the circuit from the user inputs.

**Member Contribution:**

Rex Taymany — Worked on the the front end UI and implemented the middleware necessary to connect and send information to the web based RESTful API. Configured the backend connection for the GSM to the cellular network.

Shayan Darian — Worked on the GSM to Arduino communication. Configured the necessary functions to send and receive HTTP requests from the module and record the information in the serial monitor.

Nick Marks — Worked on the Arduino to circuits connection. Configured the GSM to parse the incoming JSON response from the GSM module and set up a detection method to carry out the circuits intended functionality.

**Conclusions and Future Improvements:**

We were able to successfully implement and demonstrate our intended functionality of switching on and off an LED through our design. There is a lag time of about 2 seconds in between user input and LED output; the tradeoff of this lag time is the benefit of being able to operate the app from any location within an ISP's coverage range for greater operating distance and flexibility of location. This project served as a great proof of concept and exposure to wireless technology and mobile app development where future improvements can be easily implemented. One such improvement would be for the circuit to accommodate more circuit elements that would be able to relay information such as LED brightness, sensor information, and connected Arduino port to the app. The app itself could be improved upon to automatically update the UI with controls in the event that elements are added/removed from the circuit.