**Spike 06 outcomes**

**Name: Send a message when a door is left open with the esp8226**

**Goals:**

Control the io pins with Lua script, and send network messages based on the response.

**Personnel:**

primary – Edward

**Technologies, Tools, and Resources used:**

* ESP12 Full evaluation board, flashed with NodeMCU firmware - http://tronixlabs.com/wireless/esp8266/esp8266-esp-12-full-evaluation-board/
* A reed switch, purchased from Jaycar - <http://www.jaycar.com.au/Alarm-%26-Asset-Protection/Home-%26-Commercial-Alarms/Pro-Sensor%2C-Sirens-%26-Strobes/Miniature-Reed-Switch-%26-Magnet---N-C/p/LA5074>
* Example NodeMCU code from a workshop/tutorial - <https://github.com/geekscape/nodemcu_esp8266>
* ESPlorer cross-platform IDE - <http://esp8266.ru/esplorer/>

**Tasks undertaken:**

Adapted an existing tutorial on sending a UDP message to a server when a button is pressed. Added a timer that would run when the switch is left in the open position, and send the message periodically.

Code - init.lua, setup.lua, config.lua, open\_door\_notifier.lua

1. Attach the ESP8266 board to a Wi-Fi enabled PC running ESPlorer, using a USB-Serial cable.
2. Open the Lua scripts in ESPlorer from local copies on the PC.
3. Enter the Wi-Fi SSID and password, and the IP and port of the PC in config.lua
4. Save the scripts to the ESP8266 ('Save to ESP') and restart it ('Restart ESP') using the ESPlorer interface.
5. Detach the device from the PC, add batteries for power and attach to the door.
6. Start a UDP server on the PC (eg. with Netcat: netcat -u -l 1337), open the door and observe the output.

A video demonstration is available at - https://www.youtube.com/watch?v=sDYNA6HAKGU

**What we found out:**

Able to program the device as a wireless client, and send messages to a server based on sensor output.