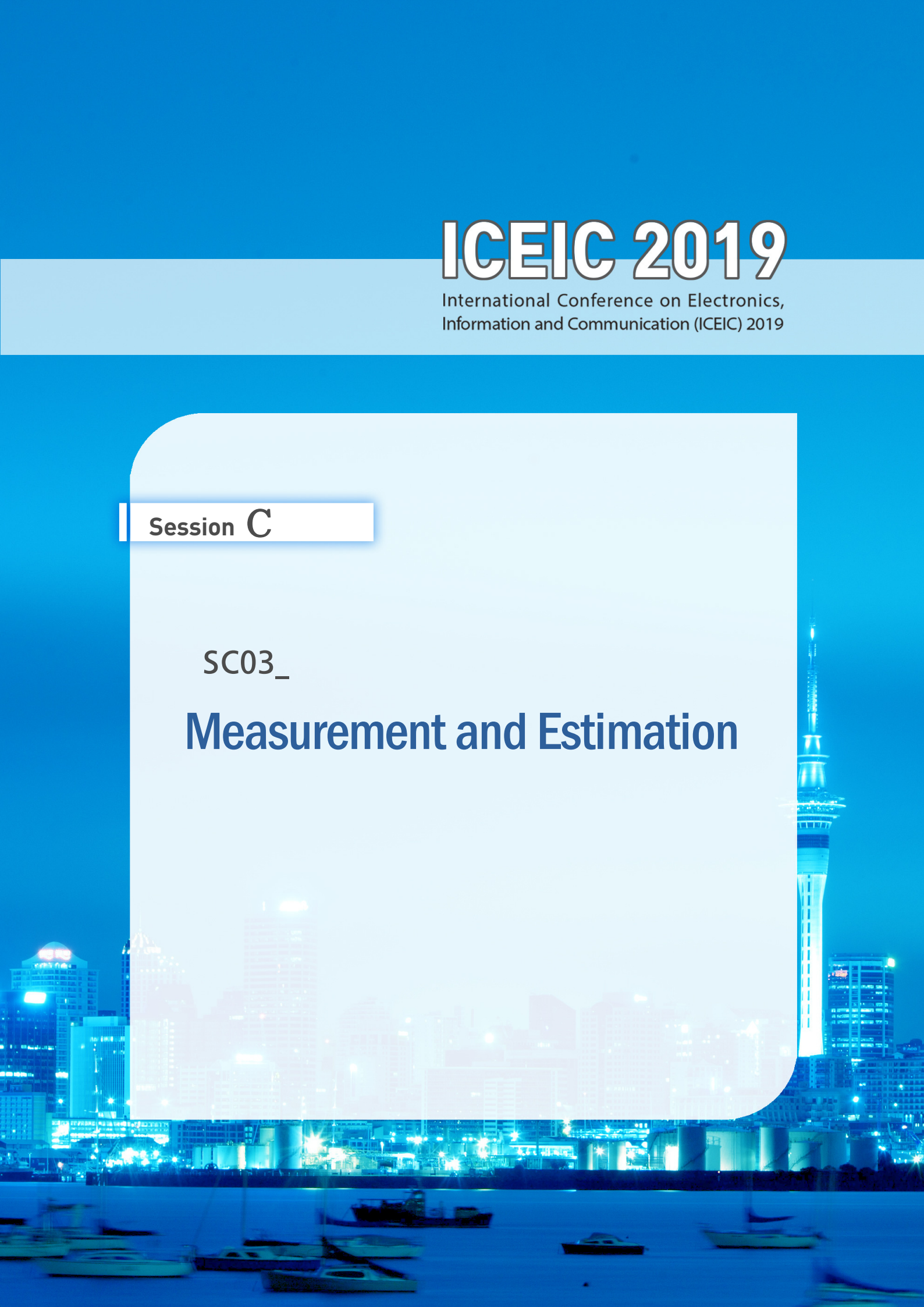
ICEIC 2019

International Conference on Electronics, Information and Communication (ICEIC) 2019

Session C

SC03_

Measurement and Estimation



Reliability of wireless sensors using low cost WiFi chipsets for Structural Monitoring

Morgan Look*, Wayne S Holmes
Department of Engineering
Unitec Institute of Technology
Auckland, New Zealand

Roger Birchmore
Department of Construction
Unitec Institute of Technology
Auckland, New Zealand

***Abstract*—This paper focused on the gathering of comparative temperature data from the ventilation systems installed in two buildings constructed to differing building standards. The Logging of temperature at 6 locations in the ventilation system ducting, plus current for the AC load was undertaken. An interval of 15 minutes for temperature was requested, however more frequent current measurements were required as short term variation had to be accounted for. At one of the sites the system proved to be reasonably reliable, with typical uptime of 7-14 days, the other site suffered from frequent drop-outs/crashes often within 24 hours of being manually reset.**

Keywords—Wireless sensing, ESP8266, Internet of things, Structural Monitoring, temperature, power measurement.

I. Introduction

This project forms part of a large long-term research project undertaken at Unitec by Birchmore [1], comparing traditional New Zealand house construction (control house) to new, airtight house specifications (test house). Two 3 bedroom houses have been constructed and environmental factors have been monitored over the last eight years. The objective for this project is to monitor the performance of identical ventilation systems (Moisture Master, NZ) which have been placed in each of the houses. This system utilizes a cross flow heat exchanger which uses the warm internal air to preheat the incoming fresh air during the cooler months, supplemented by a low power electrical heater, or bypassed completely depending on outside air temperature. International research by White, Gillott, Wood, Loveday & Vadoria, [4] indicated that the airtightness of the house impacts significantly on the energy consumed by these systems. The installed ventilation system reports the incoming and outgoing duct temperatures as shown in the plan layout and records monthly electrical consumption but does not store the results. Detailed analysis of system operating efficiencies require this data to be recorded in at least 15 minute intervals. In order to achieve this temperature measurement at multiple locations in the house ducting along with power consumption of the ventilation unit was required to be captured and logged.

Previous work in the buildings had attempted to employ Arduino Due microcontroller development boards with resistive temperature sensors on the analogue inputs logging data to SD cards, however no complete satisfactory solution had been implemented to date. As the buildings are only available for a short time, a solution needed to be implemented relatively quickly.

As there are wireless network access points installed in each of the two buildings direct logging of data to some internet location seemed like a possibility, with the distinct advantage of allowing near real-time access to data and monitoring of system status. A library and example code for uploading data from the ESP8266 to google sheets by way of a web app created by Sujay Phadke [2] was deemed to be suitable for this application.

II. Measurement Site

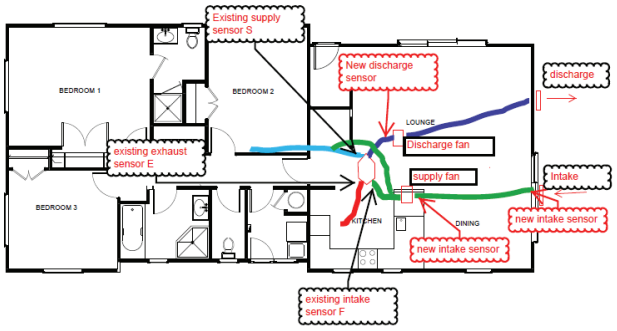


Figure 1. Test and control house temperature sensor locations.

Each of the two buildings were configured in an identical manner to capture data for both temperature and current consumption.
Temperature sensors are installed at three locations along the intake ducting, near the intake vent, and also before and after the supply fan. There are also sensors in the kitchen exhaust duct, internal supply and before the discharge fan.
Each of these sensors requires a penetration into the insulated ducting. Care was taken to seal these penetration points with tape, difficulty in taping over the inner insulating layer resulted in a small gap which may allow some leakage between layers.
A current transformer (CT) is sensing current on the supply. This is representative of the load from two fans, and the integrated heater inside the unit.

III. Wireless Measurement System

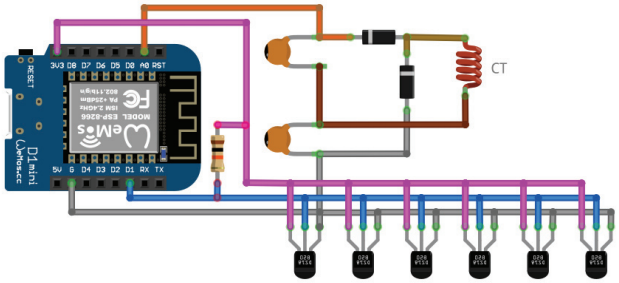


Figure 2. wireless sensor system diagram.

Due to availability and cost, the ESP8266 WiFi enabled microcontroller was selected for the first prototype. A version of the WeMos D1mini development board was chosen, as it provided a single analogue input channel, a feature not available on many other lower cost boards of this chipset. It also uses a compact PCB form factor which is emerging as a de facto standard for this class of devices.

Digital DS81B20 temperature sensors were selected over analogue equivalents, as they do not require calibration against analogue resistances and voltages. They also provide much simpler installation, with all sensors utilising only a single shared signal wire and digital input/output (IO) pin, along with power and ground wires.

Current measurements would be taken by the ADC input, via a current transformer (CT) rectified to DC by means of a voltage doubler circuit employing Schottky diodes.

Power was provided via micro-USB cable from a standard power supply typical for charging mobile phones and similar devices.

A. Embedded software

The microcontroller is programmed using the Arduino IDE, with libraries installed for ESP8266 board support, Dallas Semiconductor "OneWire" temperature sensors, and https redirection for interacting with google web apps.

At boot, the code configures the sensor array, WiFi and connection to the google script server. It then starts a timer which acquires temperature and current data every second.

The main loop waits until sufficient data samples have been gathered before initiating an HTTPS GET to the google script server and uploading averaged values for the sample period.

The HTTPS Redirect [2] library is required when interacting with google web apps, as the server responds to the initial GET request with a new hostname and URL which must then be used for the final transaction.

B. Google web app

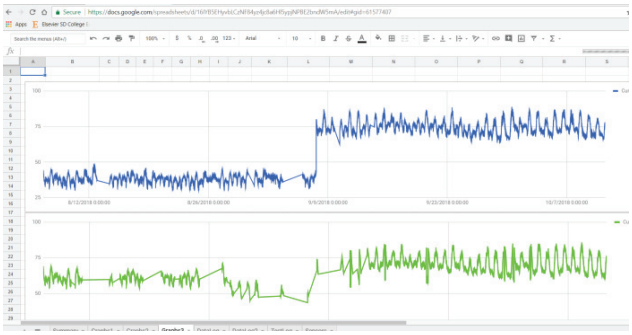


Figure 3. near real-time google sheets graphing of house data.

Data is logged to a google sheet, a web app is used to insert the data into the spreadsheet. This is created in a similar way to a macro in Excel but written in gscript and published as web accessible URL.

The web app parses the HTTPS GET string and passes the individual parameter values as arguments to a function which sets the values of the sheet cells.

Additional sheets are used to graph data and monitor system status.

IV. Deployment

Initial tests with the microcontroller proved promising, however it became apparent that the device had a tendency to trigger an internal watchdog when large strings were being manipulated. This would often occur while preparing data for posting to web addresses with long URL strings, as is the case with google web apps.

Initially it appeared the device was locking up at these times, however this only occurs the first time code is run following upload. If the device is power cycled or reset prior to running the code, subsequent watchdog events reset the device and allow the program to restart.

Following a test period of more than one week of stable operation, the device became unstable failing to establish connection to WiFi networks, often requiring many attempts before a connection was reestablished. This coincided with a significant increase in the number of people using the network, however there is no specific evidence to suggest this as a cause.

The setup code was modified to reset the device should a connection not be able to be established within a reasonable period of time. Code was also added to periodically check the state of the connection and reset the device should it not be able to be reestablished within a reasonable time.

An instance of the system was installed in each of the two houses, and data was logged to a shared google sheet for comparison. The devices were able to successfully connect to the network, capture temperature and electric current data and upload this to the spreadsheet. Unfortunately, reliability issues quickly became apparent, particularly with one of the two sites.

The device installed into the control house would frequently stop uploading data and require physical reset in order to continue.
As the two installations were identical in terms of code and hardware, and given the earlier issues with wireless reliability it appeared that the source of was related to the wireless network.
A more aggressive software monitoring and reset regime was implemented, along with making use of the on board LED to provide visual feedback on the state of the system.

This proved successful in that short term loss of connectivity was often recovered within the 15-minute sampling period. Unfortunately, lockups requiring manual device reset were still occurring reasonably frequently, resulting in loss of data.

The system installed in the test house has shown comparatively high reliability, but not entirely without failure, requiring two manual resets over a one-month period.

The control house system has unfortunately required manual resets on an almost daily basis, despite having a new microprocessor board. During periods when it doesn't require resets it logs significantly less data than the test house installation due to recoverable watchdog resets.

To reduce the number of physical interventions required at both sites, simple digital timers have been installed to reset the power at regular intervals. The test house is being reset every 3 hours and the control house every 1.5 hours. This could possibly be replaced or supplemented with hardware watchdog circuits, however due to time constraints it will not be implemented within the test period.

V. Results and Discussion

By considering time stamps of subsequent data log entries of less than 240 seconds apart as full availability, and those longer than 240 as a loss of availability with an assumed 125s available period for the eventual successful connection, we can evaluate the availability of each installation.
Availability of Internet of Things (IOT) systems can be determined by comparing the mean time to failure (MTTF) with the mean time to recovery(MTTR). Sarkar [3] defines availability (A) for internet of things service as

$$A = \frac{MTTF}{MTTF + MTTR}$$

Figure 4. calculation of device availability

As the full duration of testing is relatively short, the total available and recovery times may be used instead of mean values to establish the availability as discussed below.

By this method, prior to installation of the digital timers on the power supplies, the test house showed an availability of approx. 85%, whereas the control house was approx. 38%.
These results include several failures of duration greater than 24 hours due to occurring during the weekend. Availability should improve if a simple method of limiting these outages to less than 24 hours were implemented (e.g. a simple hardware timer to reset the power one or more times a day).
Manipulating the data to limit recovery to 24 hours, suggests availability of greater than 87% and 50% respectively should be achievable, further reducing this to 6 hours brings the availability to better than 91% and 65% respectively.
Aggressively reducing this period further still down to 3 hours for the test house and 1.5 hours for the control house we might expect availability of 92% and 75% respectively.

It was possible to test this hypothesis with the physical installation of the digital timers on the power supplies.
The test house was setup with a 3 hour reset period, and the control house 1.5 hours. Following this intervention both sites were able to operate for over 3 weeks without any physical attention.
Under these new conditions the test house provided approx. 93% availability which is very close to that predicted. The control house achieved nearly 90% a significant improvement over the predicted result.
One possible conclusion that may be drawn from this is that the microcontrollers are more stable when reset frequently.

The duration of the trial was not sufficient to introduce and test a watchdog timer circuit, this would theoretically limit recovery time to less than 120 seconds.
Neither of these interventions account for gross network or power outages which may impact availability beyond what can be managed at a device level.

VI. Further Development

Although desirable to have a robust and low cost remote sensing solution, it appears that this installation in its initial form has not resulted in a system of sufficiently high reliability. The introduction of power reset timers did remove the requirement for human intervention and bring the availability up to a level which allowed the system to deliver useful and useable data, however not all data points were within the 15-minute sampling period specification.

Addition of a hardware watchdog circuit may not significantly improve the availability beyond the approx. 90% with the current setup, but may improve the average sample period to a more satisfactory level.

Additional redundant sensor nodes could also be deployed in order to increase system availability, and also potentially allow for mutual monitoring and recovery of neighbouring nodes.

Another approach might be to substitute the very low cost microcontroller chipset with a higher quality but also higher cost alternative.

Although like the power reset and watchdog methods, all of these developments would still remain vulnerable to gross network and power outages.

VII. Conclusion

The low cost and high availability of cheap wireless networking capable microcontroller systems such as the ESP8266, make them an appealing option when looking to implement remote sensing solutions.

In the case of the installations documented here, sufficient reliability and therefore availability have proven to be more difficult to achieve than initially anticipated.

By introducing self-monitoring and recovery mechanisms, it should be possible reduce the impact of some of the observed reliability issues to significantly improve availability.

In cases where this still fails to meet reliability and availability targets higher quality components may be more suitable.

VIII. References

[1] R. Birchmore, A. Pivac, and R. Tait (2015). Impacts of an Innovative Residential Construction Method on Internal Conditions. *Buildings, 5*(1): 179-195.

[2] S. Phadke, HTTPS Redirect (Version 2.0), A library for seamless data logging, communication and control for Internet of Things., 2017. [Online]. Available: https://github.com/electronicsguy/ESP8266/tree/master/HTTPSRedirect. [Accessed: 7-Sep-18].

[3] S. Sarkar, "Internet of Things – Robustness and Reliability" in Internet of Things: Principles and Paradigms, R. Buyya Ed. Goa India: Elsevier Science & Technology, 2016, pp. 202-202.

[4] White, J., Gillott, M.C., Wood, C.J., Loveday, D.L., Vadoria, K., (2016). Performance evaluation of a mechanically ventilated heat recovery (MVHR) system as part of a series of UK residential energy retrofit measures, Energy and Buildings, 110, 1 January 2016, 220-228 https://doi.org/10.1016/j.enbuild.2015.09.059