AN INNOVATIVE PORTABLE MONITORING UNIT FOR AIR QUALITY IN ANIMAL HOUSING

BY

BOYU JI

THESIS
Submitted in partial fulfillment of the requirements for the degree of Master of Science in Agricultural & Biological Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Advisor:
Professor Richard S. Gates
ABSTRACT

The Portable Monitor Unit (PMU) is a system developed to measure ammonia (NH$_3$) and carbon-dioxide (CO$_2$) concentrations in CAFOs. However, the NH$_3$ electrochemical (EC) sensor used in the existing PMU design has become obsolete; moreover, the original PMU design required a substantial amount of manual work for system setup and data post-processing. Therefore, the objective of this project was to upgrade the PMU with a new NH$_3$ EC sensor and data acquisition and control system.

In this project, four different models of NH$_3$ EC sensors were evaluated for suitability in this application. One, the HONEYWELL EC FX sensor, was selected as the replacement. It demonstrated sufficiently fast response time to a step change in NH$_3$ (60 s to reach 95% equilibrium) and reasonable listed accuracy ($\pm 5$ ppm at 100 ppm full scale). Other evaluation criteria were nonlinearity (maximum 3.8 ppm with 54 ppm NH$_3$ reference gas), uncertainty (about $\pm 3$ ppm) and drift error (maximum 4.8 ppm within 48 h). The sensor was deemed to be acceptable based on these evaluations, and a multi-point calibration and 48 h laboratory evaluation with 24.3, 54 and 99.3 ppm NH$_3$ reference gas. The new sensor was utilized in the upgraded PMU system with 5.5 min sampling (3 min line purge + 2.5 min measurement) and 54.5 min sensor purge. An Arduino microprocessor (Mega 2560) with extended function modules (Wireless SD shield, Real Time Clock shield, Relay and LCD screen) provided functions including sampling control, system auto-reset, data centralization, real-time data processing and wireless data transfer.

The upgraded PMU (PMU III) was evaluated in two field tests at a commercial laying hen facility, and the system successfully implement the upgraded functions. The system was modified between the first and second field test mainly to improve the virtual timer and real-time data
processing algorithm in its program. With the modified PMU III system, the data acquisition system uses a real time clock, so that during the measurement, real-time processing can provide reasonable results compared to the post-processing with consistency of 94%.

A 12 h laboratory evaluation was performed to the NH₃ sensor after the field tests for comparing the consistency with the prior 48 h laboratory evaluation, and thus demonstrated the reliability (maximum difference 2.6 ppm with 24.3 ppm NH₃ reference gas) of the sensor during the field test.

**Keywords:** microprocessor, data acquisition and control system, ammonia, carbon dioxide, concentration measurement.
To father, mother, and Wenjia
I would like to gratefully thank my academic advisor, Dr. Richard S. Gates, for his guidance, patience and encouragement during my M.S. studies. He helped me to find this engineering project from which I learnt what I really expected to learn about automatic system and microprocessor. His mentorship was paramount in encouraging me to be an independent thinker and creative engineer. I was lucky to receive such an opportunity to develop my own individuality and self-sufficiency in this project. Please accept my sincere thankfulness for everything you’ve done for me, Dr. Gates.

I would also like to thank all of the teammates in this project, especially Drs. Angela Green and Tony E. Grift as my committee and who helped me to obtain the correct answer when solving my puzzles. I would like to show my thankfulness to Drs. Green and Grift for the learning and teaching experience with 3D printing they provided to me. These teaching experiences are valuable for my academic career in future to be a qualified professor and teach students. Additionally, I am very grateful for the assistance from Dr. Kenneth W. Koelkebeck, Jingwei Su, Dr. Weichao Zheng, Yijie Xiong and Xuehui Guo. I would like to thank Jingwei, Xuehui and Yijie for their help on my studying and life on campus. I would like to thank Drs. Koelkebeck and Weichao Zheng for their assistance during the field testing of this project. Without all of you, my M.S. studies could not be such a smooth and successful experience.

Finally, I would like to sincerely thank my parents. Their unending support on finance and spirit guarantee I can overcome every obstacle during the M.S. studies. I would like to thank my girlfriend, Wenjia, for her faith in me and allowing me to be as ambitious as I wanted. It is never enough to express how lucky and grateful I am, but I am -- because of all of you.
# TABLE OF CONTENTS

LIST OF ABBREVIATIONS ........................................................................................................................................... IX

CHAPTER 1. INTRODUCTION AND LITERATURE REVIEW .................................................................................. 1

1.1 AIR POLLUTION IN LIVESTOCK AND POULTRY HOUSING ........................................................................... 1

1.1.1 Gas Pollutants .................................................................................................................................................. 1

1.1.2 Odor ............................................................................................................................................................... 3

1.1.3 Dust, Endotoxin and Microorganisms ............................................................................................................ 4

1.2 METHODS AND INSTRUMENTS FOR MEASURING NH₃ AND CO₂ ................................................................. 4

1.2.1 Measurements of NH₃ .................................................................................................................................. 8

1.2.2 Measurements of CO₂ ................................................................................................................................... 15

1.3 PORTABLE MONITORING UNIT DEVELOPMENT .......................................................................................... 18

1.3.1 The First Generation PMU (PMU I) .................................................................................................................. 18

1.3.2 Modification, Second Generation (PMU II) and Field Application ................................................................. 20

1.3.3 Upgrading Motivation ................................................................................................................................... 23

CHAPTER 2. OBJECTIVES ................................................................................................................................. 24

CHAPTER 3. MATERIALS AND METHODS ..................................................................................................... 25

3.1 SELECTION OF NH₃ EC SENSORS .................................................................................................................. 25

3.1.1 Criteria .......................................................................................................................................................... 25

3.1.2 Hardware and Software ............................................................................................................................... 28

3.1.3 Evaluation Procedures ................................................................................................................................... 31

3.2 EVALUATION OF THE ELIGIBLE EC SENSOR (HONEYWELL EC FX) ....................................................... 34

3.2.1 Sensor Calibration ......................................................................................................................................... 35

3.2.2 Long-term Laboratory Test .......................................................................................................................... 37
3.3 PMU System Upgrading and Field Test ................................................................. 42
  3.3.1 Microprocessor and Different Function Modules ............................................. 42
  3.3.2 System Schematic ......................................................................................... 55
  3.3.3 Software and Programming .......................................................................... 57
  3.3.4 Field Test in Laying Hen House .................................................................... 62
  3.3.5 Reliability Check ......................................................................................... 66

CHAPTER 4. RESULTS AND DISCUSSION .................................................................. 67
  4.1 Selection Results of NH₃ EC Sensors ................................................................. 67
  4.2 Evaluation Results of Eligible EC Sensor (Honeywell EC FX) ......................... 69
    4.2.1 Calibration Results .................................................................................... 69
    4.2.2 Long-term Laboratory Test Results ........................................................... 73
    4.2.3 Reliability Check ....................................................................................... 80
  4.3 Field Test Results of PMU Evaluation ............................................................... 83
    4.3.1 First Field Test with PMU III-1 ................................................................. 83
    4.3.2 Second Field Test with PMU III-2 .............................................................. 85

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS .......................................... 93
  5.1 Summary of the Project ..................................................................................... 93
  5.2 Conclusions ...................................................................................................... 94
  5.3 Recommendation for Future Work ................................................................... 96
    5.3.1 Further Study on NH₃ EC Sensor ............................................................... 96
    5.3.2 Further Optimization on PMU System ....................................................... 96
    5.3.3 Enhance the Wireless Function ................................................................. 98

REFERENCES ............................................................................................................. 99

APPENDIX A ARDUINO CODE FOR EC SENSOR EVALUATION SYSTEM .................. 112

APPENDIX B SETTING PROFILE FOR X-BEE TRANSMITTER ................................. 116
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-BEE TRANSMITTER CONNECTED TO PC</td>
<td>116</td>
</tr>
<tr>
<td>X-BEE TRANSMITTER CONNECTED TO ARDUINO MICROPROCESSOR</td>
<td>117</td>
</tr>
<tr>
<td>APPENDIX C ARDUINO CODE FOR PMU III (IPMU V2)</td>
<td>119</td>
</tr>
<tr>
<td>APPENDIX D SOPS FOR PMU III</td>
<td>139</td>
</tr>
<tr>
<td>HONEYWELL EC FX CALIBRATION</td>
<td>139</td>
</tr>
<tr>
<td>X-BEE TRANSMITTER SETUP</td>
<td>140</td>
</tr>
<tr>
<td>ARDUINO PROGRAM UPLOAD</td>
<td>141</td>
</tr>
<tr>
<td>INSTRUCTION OF PMU III OPERATIONS IN FIELD APPLICATION</td>
<td>142</td>
</tr>
<tr>
<td>APPENDIX E PLOTS AND FIGURES</td>
<td>144</td>
</tr>
</tbody>
</table>
# List of Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMU</td>
<td>Portable Monitoring Unit</td>
</tr>
<tr>
<td>CAFO</td>
<td>Concentrated Animal Feeding Operations</td>
</tr>
<tr>
<td>ER</td>
<td>Emission Rate</td>
</tr>
<tr>
<td>EF</td>
<td>Emission Factor</td>
</tr>
<tr>
<td>PA</td>
<td>Photo-acoustic</td>
</tr>
<tr>
<td>CL</td>
<td>Chemiluminescence</td>
</tr>
<tr>
<td>EC</td>
<td>Electro-chemical</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>EPA</td>
<td>Environmental Protection Agency</td>
</tr>
<tr>
<td>DACS</td>
<td>Data Acquisition and Control System</td>
</tr>
<tr>
<td>GHG</td>
<td>Greenhouse Gas</td>
</tr>
<tr>
<td>AAQ</td>
<td>Agricultural Air Quality</td>
</tr>
<tr>
<td>RTC</td>
<td>Real Time Clock</td>
</tr>
<tr>
<td>SSR</td>
<td>Solid State Relay</td>
</tr>
<tr>
<td>EMR</td>
<td>Electromechanical Relay</td>
</tr>
<tr>
<td>LCD</td>
<td>Liquid Crystal Display</td>
</tr>
<tr>
<td>IPMU</td>
<td>Intelligent PMU control box</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION AND LITERATURE REVIEW

1.1 Air pollution in livestock and poultry housing

Air pollutants are generated by livestock and poultry housing include gases, odor, dust and microorganisms, which raise concerns to both indoor and outdoor air quality. High concentration of air pollutants is harmful to the health of both the animals and employees in the animal building (e.g. Donham et al., 2002). By exchanging air with the outside environment, those pollutants may also damage the local environment near the farm (e.g. Casey et al., 2006). Certain of these air pollutants are Greenhouse Gases (GHG), which speed up the global warming (e.g. Wathes et al., 1994).

1.1.1 Gas Pollutants

Ammonia (NH₃):

Ammonia contains one nitrogen and three hydrogen atoms (NH₃). It is colorless and has a pungent smell, and can exist in gas phase for 14-36 h, on average. When NH₃ is in vapor phase, it can react with other compounds and generate particulates. NH₃ and its related chemical combinations NHₓ can acidify with SOₓ, NOₓ and volatile organic components (Groot Koerkamp, 1994) to create fine particulates. According to EPA, about 86% of the national NH₃ emissions are generated from miscellaneous sources including livestock and fertilizer from agriculture (EPA, 1998). In American broiler housing, litter is utilized continually with about one replacement per year. In the built-up broiler litter, feces are combined with uric acid plus nitrogen (chickens do not have urine, and excretes feces with uric acid together). Urease in the litter will catalyze a five step enzymatic degradation of uric acid, and NH₃ is generated as the byproduct (Singh et al., 2005).
Exposure to high concentration will cause risks to the animal and workers. Particularly, high NH$_3$ concentration affects the respiratory systems of poultry, causes keratoconjunctivitis or other diseases, and thus decreases the growth, production and food conversion efficiency in poultry industry (e.g. Kristensen et al., 2000; Arogo et al., 2002). Moreover, combined with other dust, ammonia can cause serious health problems to people working in poultry industry (Donham et al., 2002).

**Carbon Dioxide (CO$_2$):**

Carbon dioxide is composed of two oxygen atoms, each of which is covalently double bonded to a carbon atom (CO$_2$). It is colorless and odorless. Carbon dioxide is about 1.67 times heavier than the air. It is often selected as a gas parameter to study gas emissions from livestock and poultry houses (Gates et al., 2005), since animal’s respiration is the main source of producing carbon dioxide. Ventilation rates in animal housing can be estimated by deriving the relationship between metabolic heat production and CO$_2$ production (e.g., Groot Koerkamp et al., 1998). Carbon dioxide can seriously affect both indoor and outdoor air quality. At high concentrations CO$_2$ reduces the respiratory functions of the animals in the building, and can suffocate the animals (McKeegan et al., 2011). It will also affect the production of poultry by depressing the body weight of chicken with high concentration (Reece et al., 1980). Carbon dioxide is a greenhouse gas that raises the global atmospheric temperature and cause series of environmental change that significantly affects the health, animal welfare, performance and production of the livestock industry (Kuczynski et al., 2011).

**Other Gases:**

Nitrous Oxide, N$_2$O, is produced through both nitrification and denitrification processes that are common in livestock and poultry industry (e.g. Chadwick et al., 1999). It is a greenhouse
gas and is about 298 times more effective than the CO$_2$ to absorb infrared radiation (Houghton et al. 1992).

Hydrogen sulfide, H$_2$S, is colorless, heavier than air and highly soluble in water. It is very poisonous with foul odor of rotten eggs. Anaerobic conditions are necessary for H$_2$S to be generated from bacterial sulfate reduction and sulfur-containing’s decomposition (Arogo et al., 2000). Although under the same indoor air quality condition, the concentration of H$_2$S is extremely lower than the concentration of CO$_2$ and NH$_3$, because of its high poisonous, the exposure time to H$_2$S is strictly limited by the U.S. OSHA for protecting the employees in livestock and poultry industries (Lide, 1995).

Methane (CH$_4$) is generated by the organic components from microbial reactions (Casey et al., 2006). A greenhouse gas, it is 26 times more effective than CO$_2$ to absorb infrared radiation and contributes 9% to 20% of global warming potential (Sommer et al., 2000). Non-methane and volatile organic compounds (VOC) emissions can be estimated by analyzing the ventilation system of the animal housing (Burns et al., 2008; Li et al., 2008a), and their generation from animal feeding operation were documented by several projects (e.g. Trabue et al., 2013; Kreis, 1978; Hartung et al., 1994; Heber et al., 2008).

1.1.2 Odor

Odor is perceptive to the human olfactory sensation and is considered as an indicator of airborne pollutants. Livestock housing, manure storages, and land application are the main sources of manure odor emissions, and manure irrigation is still frequently applied in United States, which releases a significant amount of NH$_3$ to environment that causes odor issues (Casey et al., 2006). Studies to quantify the odor emission rates from animal housing have focused on its relationship
with the ventilation system (Lim et al., 2001), NH$_3$ emission (Gay et al., 2003), microbiology control (Zhu et al., 2000) and climatic effects (Watts et al., 1994).

1.1.3 Dust, Endotoxin and Microorganisms

Dust is generated from the animals, feed, facilities and other related sectors of livestock and poultry production housing (Casey et al., 2006). It can be classified as inhalable (all size particles) and respirable (particle diameter less than 5 micrometers) (Takai et al., 1998). Dust includes particles from soil, feed, skin, hair, feathers, feces and endotoxins (Koon et al., 1963; Anderson et al., 1966; Curtis et al., 1975; Heber et al., 1988). Studies focused on the impact of dust to the environment were classified by its concentration and characterization in animal housing (Barber et al., 1991, Maghirang et al., 1997, Jones et al., 1984, Carpenter et al., 1986), and the effects on the health status of the occupants (animal and worker) (Donham and Gustafson, 1982; Donham et al., 1986). Another two health risks of airborne particulates are endotoxins and microorganisms which can impair the lung function (Hoff et al., 2002). The generation of endotoxins and microorganisms is affected by the building types and seasons (Yang et al., 2013). Nevertheless, the occupational exposure limits of the endotoxins and microorganisms in the U.S. are not well established (Duchaine et al., 2001), mainly due to the lack of optimization and validation methods in research projects on these two injurant exposure in animal housing systems (Casey et al., 2005).

1.2 Methods and instruments for measuring NH$_3$ and CO$_2$

Studies on air quality in livestock and poultry housing were first conducted to identify the pollutants sources and to specify the components (Cotterill and Winter, 1953; Day et al., 1965; Merkel et al., 1969). The following studies focused on the effects of NH$_3$ and CO$_2$ on the health status of the animals and workers in the animal housing, the influence on animal production and
their impact after release to the atmosphere. To characterize the relationship between the pollutants and their effects on several aspects as mentioned above, concentration and emission rates of these pollutants should be tested, in order to establish upper limits on their concentrations to protect occupants and the global environment. Since this project is focusing on an instrument that is mainly used for measuring NH$_3$ and CO$_2$ concentration in animal housing, the review will focus on these two gases’ measurement.

Ni et al., (2009) presented a systematical and historical review of the development in studies on agricultural air quality (AAQ). Table 1.1 is adapted from the table in Ni et al., (2009) listing the selected studies in AAQ focused on NH$_3$ and CO$_2$ from 1963-2009. Being accelerated by the evolution in advanced instruments and computer technologies, the development of AAQ studies includes several aspects as follows:

1) More types of air pollutants can be studied in each project with wider detection range and higher resolution.
2) More precise and accurate results can be gained from monitors and sensors with fast response time that increases the density of data.
3) More durable measurement systems with less accuracy drift and interference caused by the environmental disturbance that allow continuous long-term detection at high sampling frequency.
4) More automatic operation or robotic system that reduces manual labor and human error.
5) Multiple points sampling with a central data acquisition (signal communication and data storage) and control (periodically auto-sampling) system is widely applied in animal housing.
Table 1.1: Comparison of selected publications demonstrate the development of AAQ research (1963-2003) adapted from Ni et al. (2009)

<table>
<thead>
<tr>
<th>Year</th>
<th>Scale and Facility of Study</th>
<th>Pollutant Studied</th>
<th>Measurement Duration</th>
<th>Measurement Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1963*</td>
<td>One 104 - pig finishing barn in the U.S., 2 samples (E)</td>
<td>NH₃, H₂S, CO₂, CH₄ (C)</td>
<td>NA</td>
<td>Cold trap gas collector, glass fiber paper, IR, paper chromatography, pyrolysis</td>
<td>Day et al. (1965)</td>
</tr>
<tr>
<td>1969</td>
<td>1 swine facility in the U.S., 3 samples (E)</td>
<td>CO₂, CH₄, NH₃, H₂S, alcohols, carbonyls, odor (C)</td>
<td>NA</td>
<td>Wet chemistry, gas chromatographs, sniffing</td>
<td>Merkel et al. (1969)</td>
</tr>
<tr>
<td>1982 - 1983*</td>
<td>3 layer barns in Canada, 6 trials (C)</td>
<td>NH₃, H₂S, CO₂, dust (C/E)</td>
<td>Six 24 h tests in each barn</td>
<td>MPS, IR, sulfur analyzer, particle counter</td>
<td>McQuitty et al. (1985)</td>
</tr>
<tr>
<td>1983 - 1984*</td>
<td>6 dairy barns in Canada (C)</td>
<td>NH₃, H₂S, CO₂, dust (C/E)</td>
<td>48 h test in each barn</td>
<td>Same as McQuitty et al. (1985)</td>
<td>Clark and McQuitty (1987)</td>
</tr>
<tr>
<td>1985*</td>
<td>2 turkey barns in Canada (C)</td>
<td>NH₃, H₂S, CO₂, dust (C/E)</td>
<td>1 week test in each barn</td>
<td>Same as McQuitty et al. (1985)</td>
<td>Feddes and Licsko (1993)</td>
</tr>
<tr>
<td>1988</td>
<td>5 pig farrowing rooms in Canada (C)</td>
<td>NH₃, H₂S, CO₂ (C/E)</td>
<td>1 farrowing to wean cycle</td>
<td>Same as McQuitty et al. (1985)</td>
<td>Clark and McQuitty (1988)</td>
</tr>
<tr>
<td>1992 - 1996*</td>
<td>329 livestock and poultry buildings in the U.K., Germany, The Netherlands, and Denmark (C)</td>
<td>NH₃, CO₂, microorganism, endotoxin, PM</td>
<td>24 h test each in winter and summer, 4 replicates for most buildings</td>
<td>MPS, CL, IR, mass oscillator, impaction</td>
<td>Wathes et al. (1998); Phillips et al. (1998)</td>
</tr>
<tr>
<td>1994 - 1995*</td>
<td>4 finishing swine rooms in Belgium, 1 M data points with 12 min interval (C)</td>
<td>NH₃ and CO₂ (C/E)</td>
<td>6.5 months continuous</td>
<td>MPS, CL, IR</td>
<td>Berckmans et al. (1998)</td>
</tr>
<tr>
<td>1997 - 1998*</td>
<td>8 finishing swine barns in 2 U.S. states, 155M data points with 5 s interval (C)</td>
<td>NH₃, H₂S, CO₂, PM, odor (C/E)</td>
<td>6 months continuous</td>
<td>MPS, CL, IR, FL, gravimetric, olfactometer</td>
<td>Heber et al. (2001)</td>
</tr>
<tr>
<td>2002 - 2003*</td>
<td>2 pig finishing houses in the U.S., 67M data points with 1 min interval(C)</td>
<td>NH₃, CO₂, H₂S, CH₄, NMHC, PM, odor (C/E)</td>
<td>1 yr continuous</td>
<td>MPS, CL, IR, FL, TEOM, olfactometer</td>
<td>Heber et al. (2004); Ni et al. (2008)</td>
</tr>
<tr>
<td>2003-2004*</td>
<td>10 layer houses in 2 U.S. states, 26,400 data points with 30 min interval (C)</td>
<td>NH₃, CO₂ (C/E)</td>
<td>550 house - days</td>
<td>EC, IR</td>
<td>Liang et al. (2005)</td>
</tr>
<tr>
<td>2003-2004*</td>
<td>12 broiler houses in 2 U.S. same methodology to Liang et al. (2005)</td>
<td>NH₃, CO₂ (C/E)</td>
<td>&gt; 1 yr</td>
<td>EC, IR</td>
<td>Wheeler et al., (2006); Topper et al. (2008)</td>
</tr>
</tbody>
</table>
Table 1.1(cont’d) Comparison of selected publications demonstrate the development of AAQ research (2003-2009) adapted from Ni et al. (2009)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2003 - 2004*</td>
<td>12 barns in 6 U.S. states, 200M data points with 1 min interval(C)</td>
<td>NH3, CO2, H2S, PM, odor (C/E)</td>
<td>1 year continuous</td>
<td>MPS, CL, IR, FL, TEOM, olfactometer</td>
<td>Heber et al. (2006a); Jacobson et al. (2008)</td>
</tr>
<tr>
<td>2003 - 2004*</td>
<td>1 pig house in Austria (C)</td>
<td>NH3, CH4, N2O, VOC (C/E)</td>
<td>10 months</td>
<td>FTIR spectrometer, VOC analyzer</td>
<td>Amon et al. (2007)</td>
</tr>
<tr>
<td>2004 - 2008*</td>
<td>3 layer houses in the U.S., 205M data points with 1 min interval(C)</td>
<td>NH3, CO2, H2S, PM, odor (C/E)</td>
<td>6 months in 1 house and three 6 month periods in 2 houses, all continuous</td>
<td>MPS, CL, IR, FL, TEOM, olfactometer</td>
<td>Zhao et al. (2006); Lim et al. (2007)</td>
</tr>
<tr>
<td>2006 - 2007*</td>
<td>2 TV broiler houses in the U.S., 86M data points with 30 s interval (C)</td>
<td>NH3, CO2, CH4, N2O, H2S, NMHC, PM (C/E)</td>
<td>13 months continuous</td>
<td>MPS, IR, FL, hydrocarbon, analyzer, TEOM</td>
<td>Moody et al. (2008); Burns et al. (2008)</td>
</tr>
<tr>
<td>2007</td>
<td>1 pig finishing room in Belgium (E)</td>
<td>NH3, N2O, CH4, CO2 (C/E)</td>
<td>6 days per month for 20 months</td>
<td>IR</td>
<td>Philippe et al. (2007)</td>
</tr>
<tr>
<td>2007 - 2008*</td>
<td>4 layer barns and 1 manure compost in 2 U.S. states, 107M data points with 1 min interval(C)</td>
<td>NH3, CO2, H2S, PM, odor (C/E)</td>
<td>1 year continuous</td>
<td>MPS, CL, IR, FL, TEOM, olfactometer</td>
<td>Heber et al. (2006)</td>
</tr>
<tr>
<td>2007 - 2008*</td>
<td>1 TMV turkey barn in the U.S., 83M data points with 30 s interval(C)</td>
<td>NH3, CO2, H2S, PM, odor (C/E)</td>
<td>1 year continuous</td>
<td>1 year continuous</td>
<td>Li et al. (2008)</td>
</tr>
<tr>
<td>2007 - 2009*</td>
<td>35 TMV and NV barns and one NV manure shed on 14 farms in 8 U.S. states, 2.4B data points with 1 min interval(C)</td>
<td>NH3, CO2, H2S, CH4, PM, VOC (C/E)</td>
<td>2 years continuous</td>
<td>MPS, IR, TEOM, GC - MS</td>
<td>Heber et al. (2008)</td>
</tr>
</tbody>
</table>

[a] Asterisk (*) indicates year when measurement was conducted.
[b] C = commercial; E = experimental; MV = mechanically ventilated; NV = naturally ventilated; TMV = tunnel mechanically ventilated.
[c] C = concentration; C/E = concentration and emission; PM = particulate matter.
[d] NA = not available.
[e] CL = chemiluminescence gas analyzer for NH3 measurement; EC = electrochemical sensor for NH3 measurement; FL = ultraviolet fluorescence gas analyzer for H2S measurement; IR = infrared gas analyzer for CO2 or multi-gas measurement including NH3, N2O, CH4, etc.; MPS = multi - point sampling using MPS equipment; NMHC = non - methane hydrocarbons; TEOM = tapered element oscillating microbalance.
1.2.1 Measurements of NH₃

Table 1.2 reviews the instruments utilized in studies from 2001-2009. There are mainly three different ways of measuring ammonia concentration in animal houses: Chemiluminescence (CL), Electro-chemical (EC), Infrared gas analyzing (IR) and Photo-acoustic (PA).

Chemiluminescence analyzer:

Chemiluminescence analyzers (e.g. Thermo-Scientific Model 17C, Heber et al., 2001; 2006a) works with the following theory and equation (2-1). The reaction occurs when nitric oxide (NO) mixes with ozone (O₃), and generates nitric dioxide (NO₂) and oxygen (O₂) and infrared light (hv). The intensity of the infrared light is proportional to the concentration of NO. The light emission is measured by a photomultiplier tube that generates an electronic signal, which is processed by the microcomputer to compute the NO concentration.

\[ \text{NO} + \text{O}_3 \rightarrow \text{NO}_2 + \text{O}_2 + \text{hv} \]  

(2-1)

Measuring NO₂ concentration is necessary to detect NH₃ concentration. Prior to the measurement of NO by the CL process, NO₂ and NH₃ must be transformed to NO before reacting with O₃. The Thermo-Scientific Model 17C (Heber et al., 2001 and 2006) measures the NOx (NO+NO₂) and Nt (NO+NO₂+NH₃) concentrationa to calculate the concentration of NH₃. The transformation of NO₂ occurs in a molybdenum convector that is heated to 325 °C and the transformation of both NO₂ and NH₃ happens in a stainless steel converter heated to 825 °C. The NH₃ concentration is then derived by equation (2-2), where C(Nt) is the sum of NO, NO₂ and NH₃ concentration, C(NOx) is the sum of NO and NO₂ concentration, and C(NH₃) is NH₃ concentration.

In 2006, the cost of the Model 17C was about $28000

\[ \text{C(Nt)} - \text{C(NOx)} = \text{C(NH₃)} \]  

(2-2)
**Electro-chemical sensor:**

Electro-chemical or electrochemical sensors (e.g. Drager PAC IIIH NH₃ sensor in the project of Gates et al., 2005) are available for measuring common toxic gases. EC sensors are small in size, require low power input and are inexpensive. EC sensors measure the gas concentration by detecting the electrical signal change when gases presence creates an electrochemical reaction. The electrical signal change is proportional to the concentration of gases detected by the sensor. The EC sensors are often designed with higher perceptivity to the specific gas being measured to increase the accuracy of the measurement and to reduce the interference from other contaminants.

The specific electrochemical reaction of Drager PAC IIIH (Liang et al., 2005, Gates et al., 2005) is not available. However, normally, the EC sensors for NH₃ measurement depend on the oxidation reaction of NH₃, which converts the NH₃ to nitrogen (N₂) and hydrogen protons (H⁺). Every two molecules of NH₃ produce six electrons (e⁻) after oxidation. The current phase change (equation 2-3) is used to determine the concentration of NH₃ to which the EC sensor is exposed. Hydrogen protons obtained from this reaction will then react with oxygen to generate water (Robert, 2009).

\[
2 \text{NH}_3 \rightarrow \text{N}_2 + 6 \text{H}^+ + 6 \text{e}^- \quad (2-3)
\]

\[
\text{O}_2 + 4\text{H}^+ + 4 \text{e}^- \rightarrow 2\text{H}_2\text{O} \quad (2-4)
\]

Organic ingredients (e.g. organic gel electrolyte mixture) are necessary for the NH₃ EC sensors to allow the occurrence of the oxidation reaction. Short lifespan issues exist in NH₃ EC sensor due to the inevitable consumption of the organic ingredients under continuous exposure and reaction with NH₃. A “17,520 ppm-h sensor” has one-year lifespan when exposed to NH₃ with constant concentration of 2ppm (365 days x 24 h/day x 2 ppm). The lifespan of the same sensor
will be six months if NH₃ concentration is 4 ppm and three months if 8 ppm. The sensor should be replaced once exceeding the lifespan.

NH₃ EC sensors will lose accuracy with continuous exposure to NH₃. Therefore, when utilizing NH₃ EC sensors in poultry houses where the ambient NH₃ concentration is 20-30 ppm, rejuvenating or purging is necessary for reducing the drift of accuracy over time. Specific information for utilizing EC sensors for detecting NH₃ concentration in poultry houses is given in Section 1.3.

**Infrared Gas Analyzer:**

Infrared gas analyzer consists of non-dispersive infrared (NDIR) sensors that measure the gas concentration as a function of infrared light absorbance. Chemical bonds hold atoms to become molecules and absorb energy from infrared radiation at specific wavelengths. By absorbing energy, the chemical bond vibrates at the same frequency, and the amplitude of vibration increases after the absorption is completed. Consequently, molecules that absorb energy from the radiation at certain wavelengths are heated and gain higher temperature than other molecules without heating by the light. IR sensors only detect the absorbable wavelength of the radiation of the sampling gases.

IR absorbance provides a “fingerprint” for identifying unknown contaminants in sampling gases if their chemical compounds share the same wavelength of radiation with different frequency. For those occurring at specific wavelength of the contaminants, the IR absorbance appears as absorbance peaks that are unshared with others. Those absorbance peaks are proportional to the concentration of contaminants, for instance, 0-1.53 μm is proportional to 0-100 ppm of NH₃. Although non-linear, this proportional relationship can be computed by portable gas detectors equipped with microprocessors.
Auxiliary components such as optical filters, thermopile detectors and microphones provide additional functions to the IR analyzer. Optical filters are used to select the wavelength for adapting the absorbance requirement of molecules of the target sampling gases. Thermopile detectors are used to measure the specific amount of light absorbed by different contaminants. Microphones are used to measure the pressure change caused by the heating process of the molecules resulted from radiation absorption at in specific wavelength (Robert. 2009)

Fourier transform infrared spectrometry (FTIR) can be used to detect the gas contaminants emitted from livestock and poultry houses (Amon et al., 2007). FTIR converts the measured concentration into a plume profile with wind data (ventilation or velocity of air flow) integrated across this plume to calculate the flux and emission rate of the building.

IR analyzers and FTIR instruments provide high resolution and duration to the measurement of NH$_3$ concentration with wide testing range and long lifespan when expose to high concentration of NH$_3$. However, the application of the IR analyzer and FTIR instruments is limited in studies of multi-point measurements in animal housing, mainly due to the large scale, high costs assembly difficulties and high maintenance requirements from the environmental impacts.

**Photo-acoustic:**

Photo-acoustic (PA) or Photo-acoustic spectroscopy (PAS) (e.g. Innova Model 1412 Photoacoustic Gas-Monitor used in the projects of Burns et al., 2008; Heber et al., 2008 and Li et al., 2008) was widely applied in gas concentration monitoring since 1973 (Rosencwaig, 1980). A photo-acoustic effect is the basic principle for detection. The effect is a non-radiative de-excitation process in the gas sample (e.g. NH$_3$ gas), which causes thermal emission. When the gas sample is excited by absorbing intermittent (modulated) light, the heat pulses will be emitted with the same frequency as the intermittent (modulated) light, thus the heat pulses can be measured as acoustic
signals (Buschmann et al., 1984). As shown in Figure 1.1, NH₃ gas or other sampling gas is drawn into the measurement chamber of the monitor, and excited by the light from the IR source which is modulated by the Parabolic Mirror and Optical Filter, and pulsed by the Chopper Wheel. The generated heat pulses from the gas, as acoustic signals, are received by the microphones and transformed into gas concentration through the processing algorithm in the system. The PAS analyzers can provide high resolution with quick response time for measuring NH₃, however the massive preparatory operations and the cost (about $70K for one Innova 1412) of PAS analyzers limited their application for multiple-points gas measurement in animal building. PAS analyzers with IR as their light source are classified as PAIR analyzers which are included in IR analyzers.

Figure 1.1 Principle of Innova 1412 PA Gas Monitor Measurement System adapted from website of manufacturer (source: Lumasense Inc. website http://www.lumasenseinc.com/EN/products/gas-monitoring-instruments/photoacoustic-gas-monitor-innova-1412i/)
### Table 1.2 Review of instruments for NH\textsubscript{3} measurements (2001-2007)

<table>
<thead>
<tr>
<th>Year</th>
<th>Type of Instrument</th>
<th>Range (ppm)</th>
<th>Resolution</th>
<th>Accuracy (ppm)</th>
<th>Response (s)</th>
<th>Accuracy Drifts</th>
<th>Operation</th>
<th>Cost ($)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-2006</td>
<td>TEI Model 17C(CL)</td>
<td>0.005-50</td>
<td>0.5% full scale</td>
<td>±1 % full scale</td>
<td>120</td>
<td>Zero(24h): 1 ppb</td>
<td>Power: 90 -110 VAC @ 50/60 Hz</td>
<td>28000</td>
<td>Herber et al. (2001); Herber et al. (2006)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Span(24h): 1% full scale</td>
<td>105-125 VAC @ 50/60 Hz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>210-250 VAC @ 50/60 Hz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>500 Watt Refresh: Model 111 Zero-Air Supply System</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Output: 4-20 mA, RS-232, RS-485</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2003-2008</td>
<td>PMU Pac IIIH (EC)</td>
<td>0-200</td>
<td>1 ppm</td>
<td>±10%</td>
<td>120-180</td>
<td>Zero &amp; Span(24h): Changes over time</td>
<td>Power: 9V alkaline battery, 600 h operation Refresh: DPST time relay, 3-way solenoid valve. 2min ammonia, 8min zero gas Output: RS-232 port</td>
<td></td>
<td>Gates et al. (2005); Amaral et al. (2007); Burns et al. (2007); Amaral et al. (2008); Gates et al. (2008a); Green et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hygiene: 420.7 Sensor: 335.75 Total: 756.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004-2007</td>
<td>High resolution FTIRspectrometry(IR)</td>
<td>Detect limit: 0.5-Unclear from the paper</td>
<td>Spectral resolution: 0.25 cm(^{-1})</td>
<td>Unclear from the paper, usually ±2 ppm or 3% of reading</td>
<td>Unclear from the paper, usually &lt;= 20</td>
<td>Unclear from the paper, it depends on the type of instruments</td>
<td>Operating with 8 m light path, the absorption spectra is quantified by several calibration methods and the absorption peaks are integrated to increase the accuracy</td>
<td>Unclear from the paper, it depends on the type of the instrument, usually 20,000-1,000,000</td>
<td></td>
</tr>
</tbody>
</table>
Table 1.2 (cont’d) Review of instruments for NH₃ measurements (2004-2009)

<table>
<thead>
<tr>
<th>Year</th>
<th>Type of Instrument</th>
<th>Range (ppm)</th>
<th>Resolution</th>
<th>Accuracy (%)</th>
<th>Response (s)</th>
<th>Accuracy Drifts</th>
<th>Operation</th>
<th>Cost ($)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004-</td>
<td>Model 1312 Photoacoustic Multi-Gas Monitor Innova Air</td>
<td>Detect limit: 0.2-Unclear from the paper</td>
<td>Unclear from the paper, may equal to detect limit</td>
<td>Repeatability: 1% of measured value</td>
<td>25-75, depends on the monitor setup</td>
<td>Zero(3 months): ± 0.2 ppm</td>
<td>Power: 100-127V and 200-240V (50-400Hz) ± 10% AC Pumping rate: 30cm3/s (flushing sampling tube) and 5cm3/s (flushing measurement chamber) Output: RS-232</td>
<td>Daily Rental: 450.00 Weekly Rental: 900.00 Monthly Rental: 2,200.00 Replacemen t Cost: 38,000.00</td>
<td>Philippe et al. (2007)</td>
</tr>
<tr>
<td>2007</td>
<td>Tech Instruments(PA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Scan(3 months): ± 2.5% of measured value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2008-</td>
<td>MAEMU-Model 1412 Innova Photoacoustic Gas-Monitor(PA)</td>
<td>0.2-200</td>
<td></td>
<td></td>
<td></td>
<td>Zero(3 months): ± 0.2 ppm</td>
<td>Power: 100-127V and 200-240V (50-400Hz) ± 10% AC Pumping rate: 30cm3/s (flushing sampling tube) and 5cm3/s (flushing measurement chamber) Output: RS-232</td>
<td>Unclear from the paper, may be same rental price with Model 1312 Burns et al. (2008), Heber et al. (2008), Li et al. (2008)</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Scan(3 months): ± 2.5% of measured value</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[a]: CL = chemiluminescence gas analyzer for NH₃ measurement; EC = electrochemical sensor for NH₃ measurement; IR = infrared gas analyzer for NH₃ or multi-gas measurement including NH₃; PA = photo-acoustic analyzer.
1.2.2 Measurements of CO$_2$

Table 1.3 summarizes the measurements and instruments for CO$_2$ concentration. From 2001-2009, the instruments utilized in different studies are all IR or PAS gas analyzers. The principle of these analyzers for NH$_3$ concentration measurement were introduced in Section 1.2.1, and the principle of CO$_2$ analyzers only require different parameters for instrument settings, thus the introduction of CO$_2$ analyzers for measuring CO$_2$ is omitted. The IR CO$_2$ sensor (model GMT 220, Vaisala Inc., 194 S Taylor Ave, Louisville, CO 80027) from the original Portable Monitoring Unit (PMU) system (Gates et al., 2005) was retained in this project, as its reasonable price and satisfactory performance.

In studying air quality in animal housing, especially in poultry or layer hen housing, emission of NH$_3$ and CO$_2$ are two of the basic concerns. There are mainly four ways for measuring gas concentration: CL, EC IR and PAS monitors. For testing NH$_3$ concentration, to build a low cost instrument with portable property, EC sensors or monitors may show more advantages. However, the accuracy drift and short lifespan of EC sensors are two big challenges for practical application. By taking a periodic control of testing NH$_3$ and purging the sensor, these two drawbacks can be overcome to some extent. In the PMU system, which is introduced in next section, the application of an EC sensor is the main technique for cost reduction. The measurement on CO$_2$ concentration retained the same sensor from current PMU system which can still provide competitive performance.
Table 1.3 Review of instruments for CO₂ measurements (2001-2007)

<table>
<thead>
<tr>
<th>Year</th>
<th>Type of Instrument[^a]</th>
<th>Range (ppm)</th>
<th>Resolution</th>
<th>Accuracy (%)</th>
<th>Response (s)</th>
<th>Accuracy Drifts</th>
<th>Operation</th>
<th>Cost ($)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>Model 3600, Mine Safety Appliances (IR)</td>
<td>0-5000</td>
<td>±2% full scale</td>
<td>Noise: ±5% full scale</td>
<td>97% of a step-change in 12 s</td>
<td>Short-term: ± 1%</td>
<td>Power: 10VA Pumping rate: 1 liters per min Output: 4 to 20 mA, or 0 to 1 Volt, or 0 to 10 Volts</td>
<td>$7000</td>
<td>Heber et al. (2001); Heber et al. (2006b); Ni.et al. (2008)</td>
</tr>
<tr>
<td>2003</td>
<td>PMU Model GMT222, Vaisala 0-5000, Inc., Woburn, MA (IR)</td>
<td>0-10000</td>
<td>1ppm</td>
<td>± (1.5% of range + 2 % of 30 reading)</td>
<td>Long-term (2 years): ±5% full scale</td>
<td>Power: 24 VAC/DC Out: 0 to 20 or 4 to 20 mA and 0 to 10V</td>
<td>$638</td>
<td>Liang et al., (2005); Gates et al., (2005); Amaral et al., (2007); Burns et al., (2007) Amaral et al., (2008) Gates et al., (2008a)</td>
<td></td>
</tr>
<tr>
<td>2004-2007</td>
<td>High resolution FTIR spectrometry (IR)</td>
<td>275-Unclear</td>
<td>spectral resolution : 0.25 cm⁻¹</td>
<td>Unclear from the paper, usually ±2ppm reading</td>
<td>Unclear from the paper, usually &lt;= 20</td>
<td>Unclear from the paper, it depends on the type of instruments</td>
<td>Operating with 8 m light path, the absorption spectra is quantified by several calibration methods and the absorption peaks are integrated to increase the accuracy</td>
<td>Unclear, it depends on the type of the instrument, usually $20,000-$1M</td>
<td>Amon et al. (2007)</td>
</tr>
<tr>
<td>2004-2007</td>
<td>Model 1312 Photoacoustic Multi-Gas Monitor Innovia Air Tech Instruments (PA)</td>
<td>Detect limit 340-Unclear from the paper</td>
<td>Unclear from the paper, may equal to limit</td>
<td>Repeatability: 1% of measured value</td>
<td>25-75, depends on the monitor setup</td>
<td>Zero (3 months): ± 0.2 ppm Span (3 months): ± 2.5% of measured value</td>
<td>Power: 100-127V and 200-240V (50-400Hz) ± 10% AC Pumping rate: 30cm3/s (flushing sampling tube) and 5cm3/s (flushing measurement chamber) Output:RS-232</td>
<td>Daily Rental: $450 Weekly Rental: $900 Monthly Rental: $2.2k Replacement Cost: $38-$50k</td>
<td>Philippe et al. (2007)</td>
</tr>
</tbody>
</table>
**Table 1.3 (cont’d) Review of instruments for CO₂ measurements (2008-2009)**

<table>
<thead>
<tr>
<th>Year</th>
<th>Type of Instrument[a]</th>
<th>Range (ppm)</th>
<th>Resolution</th>
<th>Accuracy (% of measured value)</th>
<th>Response (s)</th>
<th>Accuracy Drifts</th>
<th>Operation</th>
<th>Cost ($)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008-2009</td>
<td>MAEMU-Model 1412 Innova Photoacoustic Gas-Monitor (PA)</td>
<td>340-10000</td>
<td>Unclear from the paper, may equal detect limit</td>
<td>27-150, depends on the monitor setup</td>
<td>Zero(3 months): ±0.2 ppm</td>
<td>Power: 100-127V and 200-240V (50-400Hz) ± 10% AC</td>
<td>Lens in Innova Model 1312/1412 $65-75k complete</td>
<td>Burns et al. (2008), Heber et al. (2008), Li et al. (2008)</td>
<td></td>
</tr>
<tr>
<td>2008-2009</td>
<td>MAEMU-Model 1412 Innova Photoacoustic Gas-Monitor (IR)</td>
<td>340-10000</td>
<td>Unclear from the paper, may equal detect limit</td>
<td>27-150, depends on the monitor setup</td>
<td>Zero(3 months): ±0.2 ppm Span(3 months): ±2.5% of measured value</td>
<td>Pumping rate: 30cm³/s (flushing sampling tube) and 5cm³/s (flushing measurement chamber) Pumping rate: 30cm³/s (flushing sampling tube) and 5cm³/s (flushing measurement chamber) Output: RS-232</td>
<td>Lens in Innova Model 1312/1412 $65-75k complete</td>
<td>Burns et al. (2008), Heber et al. (2008), Li et al. (2008)</td>
<td></td>
</tr>
</tbody>
</table>

[a]: CL = chemiluminescence gas analyzer for NH₃ measurement; EC = electrochemical sensor for NH₃ measurement; IR = infrared gas analyzer for CO₂ or multi-gas measurement including NH₃; PA = photo-acoustic analyzer.
1.3 Portable Monitoring Unit Development

1.3.1 The First Generation PMU (PMU I)

The PMU was first developed by Xin et al., (2002) and Gates et al., (2005) and was mainly used for monitoring the gas concentration of NH$_3$ and CO$_2$ in concentrated animal feeding operations (CAFOs) as part of emissions research. These applications required portable, maneuverable, reliable and inexpensive instruments. These requirements are caused by a need to measure multiple points of gas concentration (e.g. near ventilation fans in representative areas within a building. The PMU fulfilled these requirements by applying low cost NH$_3$ EC sensor Drager PAC IIIH for measuring the NH$_3$ concentration and designing a gas sampling system within a protective enclosure.

The NH$_3$ EC sensor (Drager PAC IIIH) was selected from an evaluation conducted by Xin et al., (2002) that compared it with the other two NH$_3$ EC sensors (Polytron II-HC and Polytron II-LC) and was motivated by obsolescence of the Polytron series of sensors. The evaluation was performed by comparing these three sensors to test zero gas (0 ppm) and NH$_3$ gas with 45.8 ppm for 20 h (30 s data interval), as shown in Figure 1.2. From the test results, the PAC IIIH was selected as best of the three sensors evaluated due to its steady readings and reasonable response time. The time for PAC IIIH to change from 0 ppm to 45.8 ppm was 4 min. In comparison, the time to change from 45.8 ppm to 0 ppm was 6 min. A 48 h test (30 s sampling interval) with 45.8 ppm NH$_3$ gas and normal fresh air was performed to PAC IIIH. The 48 h test applied 5 min sampling time and 10 min purging time, in anticipation of higher level of NH$_3$ in field condition that will increase the purging time. The maximum measurement for each sampling time gradually increased from 46 ppm to 51 ppm, indicating a drift caused by the saturation of the sensor. The drift of the sensor was deemed acceptable, since it was lower than the value of its accuracy (± 5.6
To gain a reasonable estimate of NH$_3$ concentration from the original data of the sensor, a processing method was developed as (1) subtracting the maximum and minimum readings in each 5 min sampling time, then (2) taking an average of the first 3 min readings (or finding the maximum reading). By using this processing method, the estimated uncertainty in NH$_3$ concentration was less than 1 ppm at 45.8 ppm NH$_3$.

![Diagram](image)

*Figure 1.2 EC sensors evaluation adapted from Xin et al., (2002)*

Next, Xin et al., (2002) designed a measurement system to apply two PAC IIIH sensors for NH$_3$ measuring in animal building, which aimed to reduce the measurement’s variance by 2 and standard deviation by $\sqrt{2}$. The system utilized GMT 222 from Vaisala, Inc. for measuring CO$_2$ concentration, and applied a HOBO data logger for recording the measuring data. The periodic switching between sampling and purging was controlled by a 3-way solenoid valve that was operated by a manually adjustable on-off cycle timer. The voltage output when switching between sampling (1.2 VDC) and purging (0 VDC) was also recorded by a HOBO data logger for checking the working status of the system. A low-heat output pump with metal enclosure was placed...
upstream of the CO₂ sensors to avoid diluting sample gas with ambient air through leakage. The pump was placed downstream of the NH₃ sensor to guarantee the NH₃ in sampling air was measured before reacting with the material of the pump. In the tubing connection, the sampling air was separated into three pathways to NH₃ sensor, CO₂ sensor and the bypass. The three pathways merged into one at the exhaust port. The two reasons for using the bypass line were (1) to shorten the delivery time with a high flow rate sampling while guaranteeing acceptable flow rate to the sensor and (2) to reduce the work load of the pump. The components in PMU system except the pump and power supply were fixed on an aluminum backplane. The backplane was mounted within a protective box that allowed easy transportation and could be mounted on the wall of animal building. Filters were connected at the end of all sampling and exhaust tubes. The schematic illustration of the first generation PMU is shown in Figure 1.3. The material cost of one PMU was about $3400. The performance evaluation was taken by comparing PMU with an EPA-approved measurement method of CL NH₃ analyzer (model 17C, Thermo-Environmental Instruments, Franklin, MA) in a field test. The results of these two approaches showed reasonable agreement.

1.3.2 Modification, Second Generation (PMU II) and Field Application

The first generation of PMU utilized a negative pressure sampling system (to NH₃ sensor) in which the pump was placed downstream of the sensor. However, the leakage in this negative pressure system could create a drift problem (the air around the leakage was mixed in the tube with the sampling air) in NH₃ sensor readings. Compared with the leakage problem, gas absorption onto inner surfaces, or any reaction problem which may occur on the material of the pump, was neglected. Therefore, in the second version of PMU system, as described in Gates et al., (2005), the pump was moved to the upstream of the sensor, which meant both NH₃ and CO₂ sensor were placed downstream of the pump and were under positive pressure. The pump used Teflon-coated
internal parts. In addition, a pressure sensor (Setra Static Pressure Sensor) and a temperature sensor (HOBO -TMC6-HD) were added into the system. The pressure sensor was used for measuring the static pressure difference between animal building and ambient environment. A photo of the second generation PMU is shown in Figure 1.4. The second generation PMU was evaluated by comparing it with the same CL NH₃ analyzer that was used to evaluate the first generation PMU, an Innova model 1312, and 2% certified calibration gas. The results between these methods showed reasonable agreement (0.1 ppm difference with 3.2 ppm standard deviation). The material cost of the second generation PMU was $4500.

Figure 1.3 First generation PMU schematic illustration adapted from Xin et al., (2002)
After it was developed, the second generation PMU was then applied (Gates et al., 2005) in several projects for measuring NH\textsubscript{3} emission rate (ER) in broiler and laying hen houses (e.g. Li et al., 2005; Wheeler et al., 2006; Liang et al., 2006; Xin et al., 2009; Gates et al., 2008b; Topper et al., 2008; Casey et al., 2010). The data of fan capacity, fan on-off times, CO\textsubscript{2} concentration and pressure difference were utilized for calculating the ventilation rate (VR) of the fans in the building (Gates et al., 2004, 2005). The ER of NH\textsubscript{3} was obtained as the product of VR and NH\textsubscript{3} concentration. Liang et al., (2005) measured the NH\textsubscript{3} emission from ten commercial layer houses (six high-rise or HR houses and four manure-belt or MB houses) in Iowa (IA) and Pennsylvania (PA) by using the second generation PMU. The results showed the NH\textsubscript{3} emission was influenced by different house structure, manure handling scheme, diet nutrition management and geographical position. Wheeler et al. (2006) measured the NH\textsubscript{3} emission from twelve commercial broiler houses
periodic 48 h test over a one-year span in Kentucky and Pennsylvania. The measured broiler houses represented a variety of building types and different management (built up or new litter each flock) and climate conditions (cold or mixed humid). Topper et al. (2008) used these measurements to evaluate the differences between built-up and new litter on ER. Their study concluded that litter management can reduce the emission of NH$_3$.

1.3.3  Upgrading Motivation

The experience of utilizing the second version PMU in field research also involved some inconveniences. The second version PMU system was not very user friendly in two aspects. The first one was the manually adjusted on-off timer for switching between sampling and purging operations. Since multiple PMUs were applied in the field, the timer and data logger in PMU systems had to be manually reset with same start time, and they were hard to consistently synchronize. The second inconvenience was the requirement of manually downloading and processing data from different sensors and data loggers in multiple PMUs when finishing each field test. These two inconveniences wasted a huge amount of time during each test and decreased the efficiency of the measurement.

Despite these limitations, with patience and endurance the second version PMU system was still acceptable, and applied in several projects. The primary reason for a further upgrade of the second generation PMU was the obsolescence of the NH$_3$ sensor (PAC IIIH). For applying PMU in future research, a new eligible NH$_3$ sensor was needed to replace the PAC III sensor for measuring NH$_3$ concentration. Therefore, this provided an opportunity to include additional features to overcome those inconvenience in manual operations.
CHAPTER 2

OBJECTIVES

This project aimed to develop a new PMU system with a replacement of the previous NH$_3$ sensor and a redesign to create a more user friendly system compared to the previous version. The objectives of this project were:

(1) To redesign the PMU system:

(2) To evaluate the new system.

To accomplish these two objectives, specific tasks includes:

(1) Select and evaluate a new NH$_3$ EC sensor with acceptable performance;

(2) Optimize the data acquisition and control system in current PMU to enable data centralization, wireless communication and real-time processing functions;

(3) Fabricate an upgraded PMU system based the second generation PMU (Gates et al., 2005).

(4) Evaluate the performance of the upgraded PMU in a commercial animal housing and debug the system.

(5) Establish an SOP for operating the upgraded PMU in future applications.
CHAPTER 3
MATERIALS AND METHODS

This chapter specifically describes three upgrade processes to the second version PMU system (the first and second versions were introduced in Chapter 1): 1) select the potential NH₃ EC sensor to be applied in upgraded PMU system by comparing difference EC sensors; 2) evaluate the selected NH₃ EC sensor to determine a reasonable sampling and purging time for measuring NH₃ and 3) upgrade the second version PMU to the third version (PMU III) based on the sensor evaluation. The performance evaluation of the PMU III was done with two field tests that were conducted in a commercial laying hen house, and is included in this section.

3.1 Selection of NH₃ EC Sensors

3.1.1 Criteria

Despite the relatively low cost of EC sensors, saturation and accuracy drift issues occurred when continually exposing the sensor to NH₃ (as shown in Figure 3.1, the drift issue occurs within 7 min after exposing the sensor to NH₃). This saturation is similar to what was encountered with EC sensors used in the earlier model PMU. However, using EC sensors to measure NH₃ concentration can decrease the material cost of instruments for studies in CAFOs, and was chosen for the PMU system by solving the saturation problem with periodic purging with zero gas.

The periodic measurements of EC sensor to the same NH₃ concentration evaluates in stability and acceptable repeatability, and determines whether the sensor’s accuracy is reliable with enough purging time. However, periodic purging will interrupt the continuous measurement of
NH₃ concentration as shown in Figure 3.2. Consequently, if the purging time is extremely long, the measured data will provide limited useful results to evaluate the air quality in animal housing.

Figure 3.1 Simulated saturation problem of EC sensors’ test

Figure 3.2 Simulated periodic test with NH₃ and zero gas
General experience on EC sensors’ application is that a longer sampling time requiring longer purging time, and the sampling time generally consists of a response time (0-2 min in Figure 3.1) and a stable reading time (2-7 min in Figure 3.1). Therefore, a rapid response time can provide more stable readings within a limited sampling time. The evaluation and identification of NH₃ sensors mainly focused on selecting an eligible EC sensor that could offer stable and accurate data collection with acceptable response time (equal or less than 5 min). The evaluation criteria of the EC sensors were defined and listed as follows:

**Measuring Range (0 - 100 ppm):**

The measuring range is the minimum and maximum NH₃ concentration that can be measured by the sensor. The measuring range for the EC sensors selected in this project was 0-100 ppm.

**Resolution (≤1 ppm):**

The resolution is the sensitivity of EC sensor to detect NH₃ gas. For field test requirement in this study, the NH₃ sensor is required to be sensitive to 1 (or lower) ppm of NH₃.

**Saturation Time or Life Span (≥ 100 h*100 ppm):**

The saturation time is the life span of the sensor before it is no longer sensitive and accurate, and is seen when the sensor losing its accuracy even with enough purging. The saturation time of the EC sensors selected in this study should be larger than 100 h * 100 ppm.

(The above three criteria are the basic specifications of EC sensors, and should be always presented in the data sheet or manuals, thus they will not be tested in the sensors’ evaluation.)

**Uncertainty (≤ 2%) and Repeatability (≤ 5%):**
The uncertainty is the difference between the average of readings EC sensor and the concentration of reference NH₃ gas. The repeatability is the standard deviation of these multiple measurements. The uncertainty and the repeatability requirements of the EC sensor in this project are equal or less than 2% and 5% of full scale of the measuring range (i.e. 2ppm and 5ppm).

**Response Time (≤ 5 min):**

The response time is defined as the time needed by the EC sensor to obtain stable readings after exposure to NH₃ gas. The definition of stable readings will be given in the Evaluation Procedures (Section 3.1.3).

**Stable Reading Time (≥ 5 min) and Drift Point:**

The stable reading time is the time duration of stable reading before saturation and accuracy drift. The drift point is defined as the time at which saturation occurs and the signal drifts. As shown in Figure 3.1, the drift point is 7 min after the start of measuring NH₃, and can either increase or decrease.

Furthermore, to interface with the microprocessor, the EC sensor must output electrical signals which are compatible with the microprocessor.

3.1.2 **Hardware and Software**

The evaluation of EC sensors simulated the periodic sampling to NH₃ gas and purging with fresh air. The evaluation system was developed as shown in Figure 3.3, and operated by a control box called the Intelligent PMU (IPMU). IPMU includes a microprocessor Arduino Mega 2560, an SD card shield, a Real Time Clock (RTC) and an LCD screen. In the evaluation system, IPMU controls a 3 way Solenoid Valve for switching the gas line between Zero Gas and Span Gas (NH₃ reference gas with different level of concentration). During the evaluation, the signal output from EC sensor were collected by IPMU, displayed on LCD screen and stored into SD card. When
utilizing the IPMU for sensor evaluation, the circuit connection needs to follow the illustration in Figure 3.4; the sensors’ signal output need to be transformed to voltage signal or other acceptable communication protocol (such as I2C and RS232) that can be received by the microprocessor. Moreover, an Arduino software program (EC Sensor Evaluation Station, Appendix A) is required to be uploaded to the Mega 2560 microprocessor. Specific introductions of IPMU control box, solenoid controlling, LCD displaying and data logging are given in Microprocessor and different function modules (Section 3.3.1).

The tubing connections of the evaluation system is shown in Figure 3.3. Pure nitrogen or normal air can be selected as the Zero Gas for purging the EC sensors, while a gas mixture of certified NH₃ (in nitrogen or normal air) can be selected as the Span Gas for measuring NH₃ concentration. In this project, normal air was applied as the Zero Gas; 1% or 2% certified concentrations of 25 (24.3), 50 (54) and 100 (99.3) ppm NH₃ mixture with nitrogen were selected as the Span Gas. A flow rate meter was installed at the downstream of the solenoid valve to modify the flow rates for different EC sensors. The EC sensor was exposed to NH₃ gas or normal air through natural diffusion in a sealed container, since NH₃ gas density is higher than normal air, the diffusion is upward. The diffusion container was replaced by an adapter for some NH₃ sensors equipped with their own tubing adapter for testing.
Figure 3.3 Tubing connection of EC sensor evaluation system

Figure 3.4 Circuit connection of EC sensor evaluation system
3.1.3 Evaluation Procedures

The evaluation process aimed to test the accuracy, repeatability, response time, stable reading time and drift point of the NH₃ EC sensors as defined previously. The specific procedures are described as follows:

1) Set up the tubing and circuit connections as shown in Figure 3.3 and 3.4. Use normal air as Zero Gas and 25 (24.3) ppm NH₃ as Span Gas;

2) Power the IPMU control box and EC sensor, then warm up the sensor for a period of time (different sensors may require different time to warm up);

3) Start data logging with 10 s interval;

4) Switch the solenoid valve to supply Zero Gas to the EC sensor by typing “0 & Enter” in the program of microprocessor;

5) When the reading of EC sensor becomes steady, switch the solenoid valve to supply Span gas by typing “1 & Enter” in the program of microprocessor, and adjust the flow rate required by the EC sensor using the flow rate meter, of which the default setting is 0.3L/min. (“Steady” is defined as the change between two readings of the sensor within 1 s as less than 0.5% of the sensor’s full span reading. With some sensors, 0.1 mA is 0.5% of its full span current output (20 mA));

6) Wait until “Steady Now” displays on the LCD screen, which means the measuring of NH₃ gas has stabilized; (The definition of “Steady” is applied in the program of microprocessor that can return a string “Steady Now” on LCD screen indicating the sensors are ready to obtain stable readings of the NH₃ concentration being measured)
7) Wait until the signal begins to drift as shown in Figure 3.1. If the drift is not seen, wait for 10 min to gain enough stable readings (Drift of accuracy means the difference of two readings within 1 s is higher than 1% of the sensor’s full span reading. The definition of drift is also applied in the program of microprocessor that will return a string “Drift Now” on LCD screen);

8) Switch the solenoid to purge the EC sensor with Zero gas until it regains the steady readings of 0 ppm;

9) Continue purging the sensor with Zero Gas for another 20 min;

10) Repeat procedure 5) to 9) for a minimum of five replicates before change the Span Gas to 50 and then 100 ppm NH₃, and conduct five replicates of the testing for each concentration level of NH₃ gas.

The IPMU box generates a data string to record the status of Zero Gas purging, Span Gas sampling, stable or steady readings and drift readings when data logging to SD card during the evaluation procedure of the EC sensor. An illustration of the data string is shown in Figure 3.5. The data string can be processed by R studio, MATLAB or Excel to compute the criteria values.
Data processing and analysis procedures are described as follows:

1) Calculate the equilibrium value by taking an average of the readings between the first “Steady Reading” and the first “Drift Reading”. Those readings have the format “…1, 1, 0…” in a data string as shown in Figure 3.5. If the sensor does not drift, take the average of the first 20 stable readings. For each level of NH\textsubscript{3} concentration, the 5 replicated tests will give 5 equilibrium values;

2) Take an average of the 5 equilibrium values. The average value will be considered as the adjusted test result for NH\textsubscript{3} gas. The difference between adjusted test result and the actual concentration of NH\textsubscript{3} gas (24.3, 54 and 99.3 ppm) is defined as the uncertainty of the sensor reading. The overall uncertainty of EC sensor is the average of the uncertainty values to different NH\textsubscript{3} concentrations. The standard deviation between the 5 equilibrium values and the adjusted test result is defined as the repeatability of the sensor reading.
Again, the overall repeatability is the average of the repeatability values to different NH$_3$ concentration. The uncertainty and repeatability were expressed as ppm in results section;

3) The response time of each replicate is calculated by counting the data number from the first data when exposing to NH$_3$ to the data which has the nearest value to the adjusted test result, and multiply the number with the data logging interval. For example, in Figure 3.5, if the adjusted test result of 25 (24.3) ppm reference NH$_3$ gas is 24.5 ppm, the response time is 5 (the data number from sampling NH$_3$ to the appearing of 24.5 ppm) multiplied by 10 s data logging interval, which is 50 s. The response time to each level of NH$_3$ concentration is gained by taking an average of the response times from the 5 replicates. Since EC sensor has different response time to different level of NH$_3$ concentration, the evaluation of response time is performed by comparing the maximum response time;

4) The stable reading time is the number of data with the format “…1, 1, 0…” as shown in Figure 3.5. For example, in Figure 3.5, the stable reading time is 5 (data numbers 5-9) multiplied by 10 s interval, which is 50 s;

### 3.2 Evaluation of the Eligible EC Sensor (HONEYWELL EC FX)

HONEYWELL EC FX was selected as the eligible sensor that might be applied in the new PMU system based on results of NH$_3$ EC sensors selection. More evaluations were required to demonstrate its reliability in long-term measurements, and to establish the time scenario of periodic operations (sampling and purging). Therefore, a long-term laboratory evaluation of HONEYWELL EC sensor simulated the working condition in field tests, and evaluated the sensor’s performance during 48 h continuous measurement.
3.2.1 **Sensor Calibration**

Prior to the long-term evaluation of the EC sensor, three HONEYWELL EC FX sensors were calibrated following the directions in the sensor’s manual book. The adapted procedures are as follows:

1) Add a 220 ohm resistor between GND and SIG port as shown in Figure 3.6;
2) Power on the sensor with 24 VDC, warm up the sensor for at least 1 h;
3) Use the VDC mode of voltage meter and place the leads on GND and SIG port on the signal processing circuit, as shown in Figure 3.6;
4) Expose the sensor to normal air or pure nitrogen with calibration adaptor (EC-FX-CA) with 0.3 L/min flow rate (in this project, normal air is preferred for zero calibration);

![Figure 3.6 Calibration diagram of HONEYWELL EC sensor](image)

5) Check the voltage reading on the meter, the sensor is considered as self-calibrated for zero gas, if the voltage is at 0.88 V ± 0.05 V;
6) Press the button on the EC sensor’s signal processing circuit for 3 s to start the calibration mode of the sensor with green LED slowly flashing (1 s interval);
7) Carefully place and align the calibration adaptor on the sensor;

8) Expose the sensor to full-scale gas (100 ppm NH₃ mixed with pure nitrogen) with 0.3L/min flow rate until the rate of output voltage increase is approximately 0.02 V/sec (which may occur from 10 to 30 s after the sensor has been exposed to NH₃);

9) Immediately adjust the sensor by turning the span pot to change the full-scale voltage output close to 4.4V;

10) Remove the calibration adaptor and the NH₃ calibration gas, otherwise the voltage will keep change after step 9).

11) Note that the time of exposing the sensor to full-scale NH₃ gas must be limited to no more than 2.5 min. If the sensor’s voltage output begins to decrease with 0.04 V/sec which means the saturation and accuracy drift has occurred, step 2) to 10) should be repeated after exposing the sensor to fresh air and waiting the sensor to decrease to 0.88V (may take 5-15 min).

12) Set up an evaluation system for the sensor as shown in Figure 3.4, periodically exposing the sensor to 100 (99.3) ppm NH₃ gas for less than 2.5 min and purging the sensor with normal air within the remaining time of 1 h (57.5 min);

13) Do at least 3 replicates of step 12) and collect the voltage data at 30 s sampling interval;

14) Choose the 3 peak values from each replicate (hourly reading) as the stable reading of full-scale gas, and take average of these stable readings as the adjusted voltage reading of the full-scale gas;

15) Take the average of the voltage data in purge time as the adjusted voltage reading of zero gas (normal air);
16) Repeat step 12) to 14) with 25 (24.3) and 50 (54) ppm reference NH₃ gas, and calculate the difference between adjusted voltage reading of NH₃ gases and adjusted voltage reading of zero gas, that is defined as voltage output of the sensor to the different level of NH₃ concentration;

17) Perform a four points linear regression with $X = [0 \text{ ppm}, 25 \text{ ppm}, 50 \text{ ppm}, 100 \text{ ppm}]$ (the reference gas concentrations in calibration should be their “exact” value on the product’s label, in this project, they were 24.3, 54 and 99.3 ppm) and $Y = [0, \text{ mean voltage output corresponding to } 24.3, 54 \text{ and } 99.3 \text{ ppm}]$;

18) Derive the conversion equation from voltage to gas concentration by inverting the linear regression. (If the linear regression is $Y = AX + B$, then the conversion equation is $X = Y/A - B/A$);

19) Compute calibration statistics to characterize the goodness of fit of the (inverted) conversion equation: a) compute “expected” concentration of the reference NH₃ gases, using the measured voltage values from steps 15 and 16. The difference between the calculated gas concentration and “exact” value of the gas concentration is considered a measure of the nonlinearity error; b) calculate the standard error of the conversion equation, by using the standard error of the regression ($SE_{y|x}$) and the slope, i.e. $SE_{x|y} = SE_{y|x}/A$. This value is considered to be the best estimate of the sensor standard uncertainty.

3.2.2 Long-term Laboratory Test

The objectives of long-term laboratory test with HONEYWELL EC FX sensors were: (1) to quantify the response time, stable reading time and purging time; (2) to finalize the scenario of periodic sampling NH₃ and purging the sensor in field test and (3) to demonstrate the reliability and stability of the sensor during long-term testing with the developed scenario. The PMU system
was developed for 2 days (48 h) measurement of air quality in animal building, therefore, the further evaluation to HONEYWELL EC FX sensor was based on 48 h laboratory tests for evaluating the long-term performance of the sensor in field application (e.g. the average response time of the sensor during 48 h laboratory test).

According to the sensor manual, four specifications are important to its performance: 1) the steady state reading, which is defined as readings in which the current output change is 0.1 mA/sec or less (or, as a voltage output, it is 22 mV/sec with 220 ohm resistor; or, as a gas concentration, it is approximately 0.5 ppm/sec which is the average of calculated gas concentration by the conversion equation from calibration); 2) the unstable reading with drift in accuracy is defined by current output change exceeding 0.2 mA/sec upon the onset of the steady rate (0.044 V/sec and 1 ppm/sec); 3) the limitation of exposing the sensor to NH₃ in each periodic test is 2.5 min, and 4) the response time to reach 90% of full-scale reading (100 ppm) should be less than 30 s.

To verify these specifications in long-term measurements, sensors were connected in the evaluation system (Figure 3.3 and 4) to run three parallel 48 h tests with 25 (24.3), 50 (54) or 100 (99.3) ppm NH₃ gas. Since the maximum duration of exposing the sensor to NH₃ should be limited to 2.5 min, which in turn means the sampling time is limited, the long-term evaluation mainly focused on determining the necessary purging time of the sensor. However, to enhance sensor saturation, which can evaluate the error or drift in accuracy with the overload of NH₃ gas in each sampling time during 48 h test, the periodic sampling and purging were set as 8 min sampling with NH₃ reference gas and 52 min purging with normal air. This arrangement of sampling and purging time to the sensor measured NH₃ with an hour interval during the 48 h test. The data logging interval during the 48 h test was 30 s.
After the long-term test was finished, the voltage readings were transformed to NH$_3$ concentration based on the calibration results and the conversion equations. The stable readings were defined as the difference between two consecutive readings of a sensor within 1 s being less than or equal tp 0.02 V (0.5 ppm). Since the data logging interval in the 48 h laboratory test was 30 s, it could not capture the change of a sensor’s readings within 1 s; thus, the stable reading was redefined as the first peak reading and the two readings after the peak reading (3 stable readings) in each cycle. This definition was established by considering that the stable reading time would always start after reaching a peak (or maximum) value in each cycle, and the other two readings (30 s interval) after the peak reading are the data during the 60 s stable reading time.

Next, the 48 h equilibrium test result of NH$_3$ concentration was calculated by taking the average of the stable readings from each 48 h laboratory test with different level of NH$_3$. The average of the three stable reading in each hour were also calculated and defined as the 1 h equilibrium test results. Thus in each 48 h laboratory test, 48 averages (1 h equilibrium test results) were obtained, and the standard deviation between these 1 h equilibrium test results and corresponding 48 h equilibrium test result is considered as the repeatability of each 48 h test.

To gain more reasonable analyses on the sensor’s saturation error and drift during the 48 h laboratory testing, measurements from the first (0 – 2), middle (23-25) and last (46-48) two hours were selected and the average NH$_3$ of each two hour stable readings (6 stable readings) was used for the equilibrium test result of the first, middle and last two hour measurement. If the equilibrium test results of the middle two hour measurement is higher than the other two equilibrium test results of the first and last two hour measurement, the drift is neglected as a data fluctuation. Otherwise, the drift is obtained by calculating the difference between the equilibrium test results of the first and last two hour measurement.
Additionally, the theory of dynamic response (Doebelin et al., 2007) was applied to the long-term evaluation to provide more reliable analyzing of the sensor’s response time. Without saturation and accuracy drift, EC sensor can be considered as a measurement device with 1st order dynamic response, and the practical condition of intermittent measuring NH₃ concentration can be considered as an unit step function input. The adapted general equation (Doebelin et al., 2007) of 1st order dynamic response to unit step function input is:

\[ X(t) = 1 - e^{-t/\tau} \]  

(3 – 2)

Where:

\( X(t) \) = a value between 0 and 1, which is the percentage of finishing the response to unit step function (dimensionless)

\( (X (t) = [V (t)-V (t=0)] / (V (t) \text{ max} - V (t=0)), \) an example of voltage output response from an EC sensor

\( t \) = the time after the sensor has been exposed to NH₃ gas (s)

\( \tau \) = time constant, indicates the response speed of the sensor. (“tau” in Figure 3.7)
Figure 3.7 shows the relationship between a unit step input and a 1st order response, which can illustrate the EC sensor’s response to NH$_3$ gas, the unit step input is the status of exposing the sensor to NH$_3$ (0 means 0 ppm, 1 means exposing to some concentration of NH$_3$). To apply the 1st order response theory in the long-term evaluation, $X(t)$, the response in Figure 3.7, is the percentage of completing the measuring of NH$_3$ (e.g. 100% means the sensor has obtained a reasonable reading of NH$_3$ concentration). Mathematically, $X(t)$ is calculated by dividing the value of the sensor’s real-time reading minus initial reading, by the magnitude of the step change in NH$_3$ concentration and expressed as a percentage. The first, middle and last one hour real-time readings (30 s interval) were selected from the 48 h laboratory test to calculate three $X(t)$s for different stages of the measurement. The time ($t$) for the value real-time reading to reach 95% (by Equation 3-2, 95% =1-exp [-t/3$\tau$]) and 99.3% (by Equation 3-2, 99.3% =1-exp [-t/5$\tau$]) of the value of the equilibrium result (or $X(t)$ to be equal to 95% or 99.3%) can be defined as the response time of the
sensor. In the long-term laboratory evaluation, the 30 s data interval was difficult to find the accurate time to obtain \( X(t) = 95\% \), and therefore \( X(t) = 99.3\% \) was considered as the standard to gain a response time with \( 5\tau \). The value of \( \tau \) was calculated by dividing the response time of \( X(t) = 99.3\% \) with 5, and consequently the response time of \( X(t) = 95\% \) was obtained by multiplying \( \tau \) with 3. From the three values of \( X(t) \), three response times of \( X(t) = 95\% \) were obtained, and their average was defined as the overall response time of the sensor.

3.3 PMU System Upgrading and Field Test

From the results and conclusions in previous procedures, the HONEYWELL EC FX sensor was selected for the new PMU system with periodic sampling and purging in each cycle. With the replacement of the obsolete \( \text{NH}_3 \) sensor, the system upgrade of the second version PMU system was initiated, and retained most of the previous components. Circuitry and tubing connections were modified and reorganized to integrate with the Arduino microprocessor for providing a user-friendly system and reducing manual works. The upgraded PMU is the third version from its first generation, and will be called PMU III.

3.3.1 Microprocessor and Different Function Modules

The user-friendly PMU III system was developed by applying a microprocessor as the controller to the system. Specifically, the microprocessor provides three functions: (1) simplifying the complicated operations to reset the on-off timer (Omron Twin Timer) in original PMU system, (2) centralizing and logging the data from different sensors in one SD card with wireless transfer function and (3) processing the data of \( \text{NH}_3 \) concentration to display an estimated concentration in each hour.
Arduino Mega 2560:

The Mega-2560 (Arduino, Ivrea, Italy), as shown in Figure 3.8, is the microprocessor utilized in PMU III system. It has 256 KB onboard memory, 54 digital input (or output) pins, 16 analog input (or output) pins, ICSP pins and I2C (SDA and SCL) pins. Mega-2560 can communicate with circuit through electrical signals or protocols such as voltage signal, SPI and I2C protocols. The program libraries for different applications with extension modules (e.g. Real Time Clock and LCD screen) are open source that can substantially reduce the work on programming. To upload programs on the Mega-2560, a programming interface (Arduino IDE) is required. Moreover, 6-12 VDC is the necessary power supply to the microprocessor. The programming on Mega-2560 that is developed to enable several auto-control functions is introduced in a separate section in this chapter (Section 3.3.3).

![Figure 3.8 Arduino Mega 2560 controller (source: www.arduino.cc)](image)

Real Time Clock (RTC) module:

An RTC module, as shown in Figure 3.9, is applied to provide the real time data that has a format as “year/month/day hour: min: second”. The module number of RTC in this project is DS3231 AT24C3 ((DS3231, USPRO, [www.amazon.com/shops/AT6S5L77ZENZI](http://www.amazon.com/shops/AT6S5L77ZENZI)), with a
program library from its manufacturer (www.adafruit.com, the library provided by the website is developed for module DS1307, and it is also effective on DS3231). RTC module requires I2C communication with Mega-2560 through SDA and SCL pins connecting with DIO 20 (SDA) and DIO 21 (SCL) on Mega-2560. In addition, the GND and VCC pins have to be connected to the GND and 5 V pins on Mega-2560 for power supply. The protocol for I2C communication is patented by the semiconductor company NXP of Philips. The real time data provided by RTC module will be recorded in SD card for combining the measuring result with a timeline in field test. Moreover, in the final version of program, PMU III system is controlled by the virtual timer (Section 3.3.3) that use the RTC as the timeline.

![DS3231 AT24C3 RTC module](source: ADAFruit)

**Solid State Relay (SSR) and 3 way Solenoid Valve:**

SSR, as shown in Figure 3.10, is an electronic switching device when applying a low external voltage to open or close a high voltage circuit (SainSmart 2 Channel SSR 5A DC-DC 5V-220V, SainStore, [http://www.amazon.com/s/ref=bl_sr_car?ie=UTF8&field-brandtextbin=SainSmart&node=1077068](http://www.amazon.com/s/ref=bl_sr_car?ie=UTF8&field-brandtextbin=SainSmart&node=1077068)). In this project, SSR allows Mega-2560 output 5 V digital signal to control a 3 way solenoid valve which is powered by 24VDC. Electromechanical Relay (EMR) is another option as a switching device, however, it usually generates a voltage shock
that resets the Mega-2560 at the moment of switching. Thus SSR is preferred to offer the switching
function without disturbing the measurement. The low external voltage side of SSR module is
connected to Mega-2560 via VCC- 5 V, GND- GND and SIG to DIO 19. (For details comparing
to SSR and EMR, see the website resource: http://www.ssousa.com/appnote040.asp). The high
voltage side is connected to the 3-way solenoid valve (Type 6014, Burkert Contromatic Corp.,
2915 Whitehall Park Dr., Charlotte, NC, USA) in series. As shown in Figure 3.11, the solenoid
valve is controlled by the Mega-2560 through SSR. When SSR receives a HIGH signal input from
Mega-2560, it will connect the circuit and allow the 3 way Solenoid Valve to be powered on.
Conversely, when it receives a LOW signal, it will disconnect the circuit and powered off the
solenoid. Since heat can accumulate when the valve is continually powered on, the normally-
closed (NC) side of the 3-way solenoid valve is connected to the fresh air tubing. Therefore, each
hour, the 3-way solenoid valve is energized for only a couple of minutes to sample barn air, and
de-energized the rest time of an hour.

Compared to the PMU II design, another 3-way solenoid valve is applied at the upstream
of the NH₃ EC sensor to provide a delay of exposing the EC sensor to the barn air when sampling.
In field test, the sampling point may be placed far from PMU system that will spend time on pre-
sampling the barn air from the sampling point to the sensor in PMU system. During the pre-
sampling time, the residual barn air from last sampling time will be piped to the sensor which is
meaningless for measuring, and will waste the limited exposing time of the NH₃ sensor (2.5 min).
For example, if the barn air is piped through a tube with 0.25 inch diameter and 6 L/min flow rate,
it will require about 1 min for passing the distance of 50 meter. Therefore, a second 3-way solenoid
valve prevents the sensor from being exposed to barn air during this pre-sampling time.
Figure 3.10 SainSmart 2 Channel SSR (source: SainSmart)

Figure 3.11 Schematic of 3 way solenoid valve
(adapted from website resource: http://www.fas.ch/info_tech_fonctions.asp?Langue=english)

**Wireless and SD shield and X-Bee shield:**

To fulfill the requirement of centralized data recording and wireless transfer of the data in PMU III system, a Wireless and SD shield (Arduino Proto Wireless SD Shield, code A000065, Arduino, http://store.arduino.cc) is applied in this project as the extension module to Mega-2560. When the module is installed on Mega-2560 as shown in Figure 3.12, it can provide data logging to SD card through SPI protocol by connecting to ICSP pins on Mega-2560. The module can also remotely communicate with computer through X-Bee radio signal transmitters. The communication requires two X-Bee transmitters (X-Bee 2mW Wire Antenna - Series 2, ZigBee Mesh, Karlsson Robotics, www.amazon.com/shops/karlssonrobotics), one is combined with Mega-2560 by installing on the Wireless and SD shield, while the other is connected to the computer through a USB shield as shown in Figure 3.13.
Before using X-Bee transmitters for wireless communication, their settings and addresses must be initialized, and connected with each other. X-Bee transmitters allow point-to-point or point to multiples communications. To use the point-to-point communication in this project, XCTU is applied as the computer software for initializing the settings on the transmitters. For setup, four parameters are necessary to configure the transmitters: ID, CH, DH and DL. The ID and CH parameters are the identification and channel of the transmitters’ communication, and they must be the same in the settings of two connected transmitters. DH and DL are the high and low address destination numbers of X-Bee transmitters. Transmitters also have SH and SL as their high and low address serial number that cannot be changed on the settings list. When connecting, the DH and DL numbers of one X-Bee transmitter should match the SH and SL numbers of the other one that will be connected, and vice versa. For example, the SH and SL of X-Bee A are 13A200 and 407B0DEC. If X-Bee B is to communicate with A, the DH and DL of B should be 13A200 and 407B0DEC, meanwhile A should use B’s SH and SL as its DH and DL settings. The setting profile is presented in APPENDIX B as a reference. In addition, the switch button on the Wireless/SD shield should be switched to “Micro” mode to enable the wireless function. Otherwise, it will stay in “USB” mode for uploading program to the microprocessor. When two X-Bee transmitters are successfully communicating with each other, the PMU III system can be wirelessly controlled by computer to reset the system or download data. The introduction of the wireless control commands is included in Section 3.3.3.
Liquid Crystal Display (LCD) screen:

A 4x20 character LCD screen (SainSmart LCD Module For Arduino 20 X 4, PCB Board, White On Blue, SainStore, https://www.amazon.com/gp/aag/details/ref=aag_m_ss?ie=UTF8&asin=&isAmazonFulfilled=1&isCBA=&marketplaceID=ATVPDKIKX0DER&seller=A10EAPE4CAYC9P#aag_legalInfo) is utilized for displaying the system status and measuring results. The LCD module is developed for communicating with microprocessor through I2C protocols which enables Mega-2560 to move the cursor and display strings on the LCD screen. The LCD module requires a 5 VDC power supply. Figure 3.14 shows a photo of SainSmart LCD Module for Arduino 20 X 4 which is the module applied in this project. By programming the LCD screen with Mega-2560,
real-time measuring results and system status (e.g. sampling or purging) can be displayed and refreshed with 1 s interval.

![SainSmart LCD Module for Arduino 20 X 4 (source: SainSmart)](image)

**Figure 3.14 SainSmart LCD Module for Arduino 20 X 4 (source: SainSmart)**

**Temperature Sensor:**

A thermistor (HOBO-TMC6-HD, Onset Computer Corp, 470 MacArthur Blvd., Bourne, MA, 02532 USA) temperature sensor is used for measuring the temperature of sampling barn air. It is retained from the second version PMU system with redesigning the circuit connection to output electrical signal that can be received by the microprocessor. Figure 3.15 is the diagram of the cable wiring to read the sensor’s output by a HOBO data logger. However, in the redesign of circuit, the cable is removed to enable a communication with microprocessor. In specific, the red and black wires are connected to the 5 V and GND pins on Mega-2560 as a power supply to the sensor. The white wire is connected to AIO 9 on Mega-2560 so that the microprocessor can receive analog reading signal. The analog reading function on Mega-2560 can measure a voltage input between 0 and 5 V. The measured result of voltage is provided as an integer number between 0 and 1023, which means the resolution of voltage reading is 10 bits. The conversion equation between the analog reading and voltage input is shown as follows:

\[ V_{input} = \frac{V_{analog}}{1023} \times 5 \]
\[ V = \frac{A}{1023} \times 5 \quad (3-3) \]

Where

\( V = \) Voltage input to microprocessor (V)

\( A = \) Analog reading number (between 0 and 1023)

The voltage input from equation (3-3) can be used to calculate the temperature measured by the sensor following two steps (Davis, 2003): (1) calculate the temperature-sensitive resistance of the thermistor (equation 3-4) and (2) calculate the temperature by using the conversion equation between temperature and resistance, which is provided by the datasheet of the sensor (equation 3-5).

\[ Rt = Ro \times \frac{V}{5 - V} \quad (3-4) \]

\[ T = \frac{1}{\ln \left( \frac{Rt}{Ro} \right) \beta + \frac{1}{To}} - 273.15 \quad (3-5) \]

Where

\( T \quad = \) Measured temperature (°C)

\( Rt \quad = \) The resistance of the thermistor (Ω)

\( Ro \quad = \) Constant 10 kΩ

\( V \quad = \) Voltage input to microprocessor (V) from equation (3-1)

\( \beta \quad = 4261 \quad (\ln (1/K)) \)

\( To \quad = 298.15 \quad (K) \)
Pressure sensor:

To measure the pressure difference between animal building and ambient atmosphere, a differential pressure sensor (Model 264, Setra Inc, 159 Swanson Rd, Boxborough, MA 01719) was also retained from the second version PMU (Figure 3.16). The pressure sensor can measure the pressure difference by connecting the one port to the outdoor atmosphere and the other port to the animal building space. The sensor provides a current output signal proportional to pressure, and utilizes a power loop circuit that combines the power supply and current output in series as shown in Figure 3.17. The pressure sensor requires a separate power supply which is higher than 9 V (in this project, a 24 VDC power supply is applied for the pressure sensor). The measurement range is 0 to 0.5 inch of water column (wc), or 0 to 124.54 Pa, and the output current is between 4 and
20 mA. Since the Mega-2560 is unable to directly measure a current signal, a 220 Ω resistor is added in series to the GND side for transforming the current output to a voltage signal between 0.88 and 4.4 V. The conversion equation between analog reading number and the voltage input from the pressure sensor is the same as equation (3-3). To calculate the pressure, equation (3-6) is shown as below:

\[
P = \frac{V - V_0}{V_{\text{span}} - V_0} \times (P_{\text{span}} - P_0)
\]  

(3 - 6)

Where

\begin{align*}
P & = \text{Measured pressure difference (Pa)} \\
V & = \text{Voltage input to microprocessor (V) from equation (3-3)} \\
V_0 & = 0.88 \text{ (V)} \\
V_{\text{span}} & = 4.4 \text{ (V)} \\
P_{\text{span}} & = 124.54 \text{ (Pa)} \\
P_0 & = 0 \text{ (Pa)}
\end{align*}

Figure 3.16 Setra 264 Pressure Sensor (adapted from Setra’s 264 manual)
NH₃ sensor:

The HONEYWELL EC FX selected for use in the PMU III system is shown in Figure 3.18. It requires a 24 VDC power supply that connects to the GNC and VCC pins on the sensor’s circuit. The sensor provides a current signal (4 – 20 mA) that is proportional to the NH₃ concentration (0 – 100 ppm). The current signal is also received by the microprocessor with a 220 Ω resistor connected to the GND and SIG pins on the sensor’s circuit to create a voltage signal. Equation (3-3) is used to obtain the voltage input from the sensor. The measured NH₃ concentration is then calculated based on the calibration results to the sensor (Section 4.2.1).
**CO2 sensor:**

The PMU also uses a CO₂ monitor (model GMT 220, Vaisala Inc., 194 S Taylor Ave, Louisville, CO 80027) as shown in Figure 3.19; it is an IR sensor retained from PMU-II. Similar to the NH₃ EC sensor, GMT 220 also requires a 24 VDC power supply and provides a current output (4-20 mA) that is proportional to CO₂ concentration (0 - 5000 ppm), and the current output is converted to a voltage signal for the microprocessor by adding a 220 Ω resistor between the SIG and GND pin on the sensor’s circuit. The conversion equation between voltage and CO₂ concentration is similar as equation (3-6) and shown as follows:

\[
C = \frac{V - V_o}{V_{span} - V_o} \times (C_{span} - C_o) \quad (3-7)
\]

Where

- \(C\) = Measured CO₂ concentration (ppm)
- \(V\) = Voltage input to microprocessor (V) from equation (3-3)
- \(V_o = 0.88\) (V)
- \(V_{span} = 4.4\) (V)
- \(C_{span} = 5000\) (ppm)
- \(C_o = 0\) (ppm)
3.3.2 System Schematic

Circuit connection:

Figure 3.20 illustrates the circuit connections between the functional modules and the microprocessor as mentioned in Section 3.3.1, is presented below. The PMU III system circuitry consists of four parts: (1) a control box called Intelligent PMU (IPMU) that groups microprocessor, RTC module, LCD screen and Wireless SD shield together; (2) the sensors which are applied for measuring the air quality (NH₃, CO₂ concentration, temperature and pressure difference) and providing voltage signal inputs to the control box; (3) sampling controls, that consist of Relay and 3-way Solenoid Valves for switching between the gas line of sampling and purging (the air pump retained from the second version PMU is applied to provide air flow in the PMU III system) and (4) the power supplies in the PMU III system including two voltage transformers that convert 120VAC to 12 VDC (to power the microprocessor) and 24 VDC (to power the sensors, solenoid valves and relay). The sampling pump is directly powered by 120 VAC.
Figure 3.20 Circuit system of PMU III (not drawn to scale)
**System sketch and tubing connection:**

The rearrangement of previous components in the second version PMU system, including the sensors, air flow meters, 3 way solenoid valve, pump and power supply, was performed to leave space for installing the control box (IPMU), replacing the NH\textsubscript{3} sensor and adding another 3 way solenoid valve. The metal board retained from the second version PMU is applied for fixing the components in the protective case. Figure 3.21 shows the system sketch with the rearrangement. The tubing connection is also illustrated in Figure 3.21 with the arrows to express the direction of air flow when the air pump is opening. As mentioned in the review of the second version PMU, the air pump is placed at the upstream of the sensors to avoid the leakage problem, and a bypass line is used to reduce the pressure accumulation in the tube when applying different air flow rate (11 L/min for pre-sampling the air into PMU system and 0.3 L/min for exposing the air to the sensors) Additionally, the material of tubes and connectors are recommended to have resistance to NH\textsubscript{3} with 100 ppm for minimizing the measurement error caused by the reaction between NH\textsubscript{3} and the material.

3.3.3 **Software and Programming**

Arduino IDE (Version 1.5.5) is the computer software for editing and uploading program on the microprocessor Arduino Mega-2560. The microprocessor is controlled by a program to cooperate with different modules for accomplishing different functions of PMU III system. The programing includes: (1) a virtual timer to control sample and purge times and the data logging interval; (2) a series of subprograms that implement the extension modules including RTC, Wireless SD shield, LCD screen and Relay and (3) an algorithm for real- time processing of the measured voltage data into NH\textsubscript{3} concentration. As shown in Figure 3.22, the Setup and Main Loop are the two principal functions in the programming of Arduino microprocessor.
In the Setup function, the intervals for sampling, pre-sampling (sampling = pre-sampling + measuring) and purging can be reset or the default settings may be applied, which are the values used with the previous measurement. The initial time on the RTC can also be adjusted in the Setup function if it is inaccurate.

In the Main Loop function, the virtual timer is designed with two global variables (RTC-pre and RTC-cur). RTC-pre records the beginning time (e.g. 00:00:00) of each working cycle from the RTC module, while RTC-cur records the real time (e.g. 00:01:10) from RTC module. Their
difference can be calculated in the program to obtain the seconds elapsed (e.g. 00:01:10 – 00:00:00 = 70 s) from the beginning. Next, the elapsed seconds is compared with different time intervals to start or end different operations as the manual on/off timer did in previous PMU system. The virtual timer is running in each working cycle in Main Loop to control the 3 way Solenoid Valve I (as shown in Figure 3.21) switching between sampling and purging. The 3 way Solenoid Valve II (as shown in Figure 3.21) is also controlled by the virtual timer to delay the exposing time of NH$_3$ sensor to the barn air, by considering the time cost on pre-sampling the barn air from the sampling point to the sensor (specific explanation is included in Section 3.3.1, Solid State Relay (SSR) and 3 way Solenoid Valve).

Furthermore, in Main Loop, the data logging interval is change with different working status to gain more stable readings (1 s/data) during sampling time and less purging readings (600 s/data) to save the space on SD card. In addition, the data logging interval is 10 s when pre-sampling the barn air and waiting for sensor’s response. Therefore, to simplify the resetting procedures in PMU III (the Setup function), the data logging intervals are defaulted as constant values which can only be changed by editing the program.

An illustration of the system timeline during one cycle (1 hour) measuring is shown in Table 3.1. In each 1-h cycle, the system starts by sampling the barn air for 360 s. The sampling time includes a pre-sampling time of 180 s, a response time (30 s, in the setting of field test) and a stable reading time (the sum of response time and stable reading time must be lower than the 150 s allowable NH$_3$ exposure time). A stable reading is marked as such when the difference between two sensor voltage readings (within 1 s) is less than 0.02 V (0.5 ppm as NH$_3$ concentration). After sampling barn air for 330 s, the solenoid valve will be switched to purge the sensor with fresh air for the remaining 3240 s (1 hour working cycle = 3600 s = 360 s sampling + 3240 s purging).
Moreover, a data string will be generated from the program when the data logging function is activated, which includes the mark of system working status as shown in Table 3.1.

Besides, the wireless control function is programmed in both Setup and Main Loop functions to allow wirelessly resetting the time intervals, interrupting the working cycle, downloading the measuring results and formatting the SD card. The commands list of wireless control is presented in Table 3.2.

The second version of the complete program is attached in APPENDIX C and named as IPMU V2. After assembling the circuit and tubing system of PMU III following Figure 3.20 and 15, IPMU V2 is required to upload on the microprocessor (Mega-2560) for activating the electrical system of PMU III.

**Table 3.1 System timeline with operations in one cycle**

<table>
<thead>
<tr>
<th>System operations during different time intervals</th>
<th>Sampling (360 s)</th>
<th>Purging (3240 s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-sampling (180 s)</td>
<td>Measuring-sensor response and becoming steady (30 s)[a]</td>
<td>Measuring-taking stable readings (150 s)[a]</td>
</tr>
<tr>
<td>3-way Solenoid Valve I</td>
<td>To barn air</td>
<td>To barn air</td>
</tr>
<tr>
<td>3-way Solenoid Valve II</td>
<td>To bypass</td>
<td>To NH₃ sensor</td>
</tr>
<tr>
<td>Data logging interval (sec)</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Record of status in data string</td>
<td>[1]</td>
<td>[2]</td>
</tr>
</tbody>
</table>

[a] the settings of sensor response time and stable reading time, which were applied in the field test, may be adjusted in future applications to provide more reasonable data of the measurement.

**Table 3.2 Wireless commands list**

| Serial Command | Setup               | Main Loop                                                      |
|----------------|---------------------|                                                               |
| “1” + “Enter” | “Yes” or “+”       | “Download data” (only available in purging time)              |
| “2” + “Enter” | “No” or “-”         | “Delete data” (only available after downloading data)         |
| “3” + “Enter” | “Set” or “Next”     | “Reset the system”                                            |
| “4” + “Enter” |                     | “Skip the purging to testing barn air”                        |
| “5” + “Enter” |                     | “Skip the testing to purging with fresh air”                  |
| “6” + “Enter” |                     |                                                               |
| “7” + “Enter” |                     |                                                               |
| “8” + “Enter” |                     |                                                               |
Figure 3.22 Program flow chart
3.3.4 Field Test in Laying Hen House

Two 48 h field tests were performed to demonstrate the feasibility and improve any problems encountered with the PMU III system. A caged layer barn, with manure belts, in a commercial egg farm in the midwest USA was chosen for the field test. The barn had 2 floors (12 tiers of cages layers) with dimensions of 540 ft (164.59 m) length x 91 ft (27.74 m) width, with about 425,000 laying hens. Figure 3.23 is a layout of the ventilation fans in the laying hen building. Since the field test was conducted during winter time, the ventilation system for summer time that is constructed on the north and south side of the building (cooling pads and summer ventilation fans) is not discussed. Ventilation fans for winter time were located on the west and east walls of the building. The minimum ventilation variable fans (marked with “M” in Figure 3.23) normally ran continuously. When temperature in the building increased above the temperature control point, a series of ventilation stages are activated by the control system. The fans with different stage levels (marked with “I” to “V” in Figure 3.23) are activated stage-by-stage for gradual regulation of building temperature. On the west side of the building a manure storage room is connected to the west wall of the main animal building. Therefore, except for the first six ventilation fans, which directly exchange air with outdoor atmosphere, the rest of ventilation fans exchange air with outside environment through the windows of the manure storage room. The air exchange from the fans is also applied for drying the manure in the storage room. The sampling points of the two field tests with PMU III are noted in Figure 3.23 which are the first and last two minimum ventilation fans on west wall of the laying hen housing.
Figure 3.23 Building and fans layout
The performance evaluation of PMU III in the 48 h field tests evaluated two aspects of the upgraded system: (1) whether the microprocessor and the uploaded program could accomplish the objectives of optimizing the previous system (optimization included the virtual timer control, centralized data logging, wireless data transform and real-time data processing of NH₃ concentration), and (2) whether the replaced NH₃ EC sensor provided reasonable NH₃ concentration measurements. The first field test was performed with PMU III-1 (Figure 3.24), while the second was performed with PMU III-2 (Section 3.3.2). The differences between PMU III-1 and 2 are listed in Table 3.3.

Before each field test, the sensors were calibrated, the circuit and tubing connections were rechecked, and the protective case was disinfected. Dust filters were connected to the sampling and exhaust ports of PMU. The sampling points were near the center of the minimum ventilation fans as shown in Figure 3.25. The purging tube connected to the outdoor air. The PMU system was powered by the power supply in the building during the 48 h test.

<table>
<thead>
<tr>
<th>Table 3.3 Differences between PMU III-1 and PMU III-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difference of</td>
</tr>
<tr>
<td>Number of NH₃ EC sensor</td>
</tr>
<tr>
<td>Number of 3 way Solenoid Valve</td>
</tr>
<tr>
<td>Sampling time (s)</td>
</tr>
<tr>
<td>Pre-sampling time (s)</td>
</tr>
<tr>
<td>Measuring time (s)</td>
</tr>
<tr>
<td>Purging time (s)</td>
</tr>
<tr>
<td>Data logging interval (s)</td>
</tr>
<tr>
<td>Time reference of the virtual timer</td>
</tr>
<tr>
<td>Real-time data processing</td>
</tr>
</tbody>
</table>
Figure 3.24 PMU III-1

Figure 3.25 Photo of barn air sampling points in first field test
3.3.5 **Reliability Check**

Since the field tests were only performed with PMU III system, and without another measurement system to serve as a reference, the reliability of the NH₃ test results had to be checked with another approach. A laboratory reliability check was performed after the system finished the first 48 h field test. The reliability check used the same procedures as the 48 h test in further evaluation, and only required 12 h. The reliability check test results were compared to the previous 48 h test results to check whether the sensor has same performance of measuring reference NH₃ gases after finishing field test.
CHAPTER 4
RESULTS AND DISCUSSION

The sections in this chapter correspond to the sections in Chapter 3 to provide the results and discussion for each process in this project. Since the reliability check was performed under similar laboratory conditions and followed the same procedures as the 48 h test, the results of the reliability check is presented after the results of 48 h laboratory test.

4.1 Selection Results of NH$_3$ EC Sensors

Four EC sensors from different manufacturers were tested (Table 4.1). Sensor response time was tested as the first and most important criteria, and only one EC sensor, HONEYWELL EC FX, demonstrated an acceptable response time of less than 5 min; thus, other evaluations were canceled for the disqualified EC sensors. It should be noted that the unsatisfactory results of the three disqualified sensors were only effective to show the performance and quality of the sensors that were delivered to the project.

Table 4.1 lists the information from each sensor’s evaluation and their data sheets. HONEYWELL EC FX sensor has less than 1 min response time to NH$_3$ with 25 (24.3) ppm and less than 5 min to NH$_3$ with 100 (99.3) ppm. The sensor’s stated accuracy is equal or less than 5% (5 ppm) of its full-scale range (100 ppm), which will generate a ±5 ppm error to the measuring results. The average repeatability of the fifteen replicates when exposing the sensors to certified NH$_3$ concentration of 24.3 ppm, 54 ppm and 99.3 ppm NH$_3$ was found equal or less than 3% (3 ppm) of the full-scale range, so that the difference between multiple measuring results with same NH$_3$ concentration is ±3 ppm. These properties indicate that HONEYWELL EC FX sensor has
approximately similar performance as Dragger PAC IIIH with the evaluated criteria. However, a further evaluation was necessary to demonstrate its feasibility for longer time (48 h at least) application.

Table 4.1 Information and conclusion about EC NH₃ sensors’ evaluation

<table>
<thead>
<tr>
<th>Product Type/Model</th>
<th>Response Time (min)</th>
<th>Accuracy (%)</th>
<th>Repeatability (%)</th>
<th>Saturation Time (h*ppm)</th>
<th>Resolution (ppm)</th>
<th>Signal Output</th>
<th>Accept (✓) /Reject (×) /Check (?)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environmental Sensors Company Model ZDL-800</td>
<td>≥ 20 (25ppm NH₃)</td>
<td>Have not tested &amp; No information from data sheet</td>
<td>Have not tested &amp; 5% repeatability drift within 6 months (from data sheet)</td>
<td>Warranty is one year, without specific information about h*ppm</td>
<td>0.1</td>
<td>Voltage 0-5V</td>
<td>×</td>
</tr>
<tr>
<td>Aeroqual Series 300 Monitor (S-300) with Sensor Head</td>
<td>No Response (25ppm NH₃)</td>
<td>Have not tested &amp; ±10% full span (from data sheet)</td>
<td>Have not tested &amp; No information from data sheet</td>
<td>No information from data sheet</td>
<td>0.1</td>
<td>Voltage 0-5V</td>
<td>×</td>
</tr>
<tr>
<td>MQ137 Ammonia Detection Sensor Module</td>
<td>≥ 10 (25ppm NH₃) &amp; Only response to higher than 10 ppm NH₃ concentration</td>
<td>Have not tested &amp; No information from data sheet</td>
<td>Have not tested &amp; No information from data sheet</td>
<td>No information from data sheet</td>
<td>No information from data sheet</td>
<td>Voltage (No unique range and nonlinear relationship with gas concentration)</td>
<td>×</td>
</tr>
<tr>
<td>Honeywell EC-FX-NH₃ (0-100ppm)</td>
<td>≤ 1 (25ppm NH₃) &amp; ≤ 5 (100ppm NH₃)</td>
<td>≤ 5 of full scale</td>
<td>≤ 3 (within 5hr)</td>
<td>No information from data sheet</td>
<td>No information from data sheet</td>
<td>Current 4-20mA (0.88-2.4V with 220 Ω resister)</td>
<td>✓?</td>
</tr>
</tbody>
</table>
4.2 Evaluation Results of Eligible EC Sensor (HONEYWELL EC FX)

Three HONEYWELL EC FX sensors (marked as “H_EC_1”, “H_EC_2” and “H_EC_3”) were calibrated and evaluated with long-term test following the procedures in Section 3.2. Because the three evaluated sensors had high consistency of performance, H_EC_1 was selected as an example to show the plots of data processing, while other sensors’ plots were included in APPENDIX E.

4.2.1 Calibration Results

Certified reference NH\textsubscript{3} gas with 24.3 ppm ± 1%, 54 ppm ± 2% and 99.3 ± 2% was used for sensor calibration. The original data from calibration procedures to H_EC_1 are plotted in Figure 4.1, and the marked points in these plots demonstrate those selected for processing. After each measurement period, the voltage output decreased under the normal level of zero gas, and then returned to initial stage, which indicated a recovery of the sensor’s accuracy. However, this signal reduction behavior was not observed with 25 ppm reference gas.

![Figure 4.1 Calibration data of H_EC_1](image)
Figures 4.2 and 4.3 show the linear regression and conversion equation of $H_{EC\_1}$ calibration with the specifications of all three sensor’s calibrations listed in Table 4.2. From the calibration results, the relation between NH$_3$ concentration and the sensor’s voltage output is about 25 ppm/V. The standard error of the conversion equation is about ±3 ppm that is contained in the error caused by the sensor’s accuracy (±5 ppm). The nonlinearity error to each level of NH$_3$ concentration (e.g. 3.8, 2.4 and 0.6 ppm at 54ppm for sensors 1, 2 and 3 respectively) is also within the manufacturer’s accuracy claim (±5 ppm). The standard errors of the conversion equations were ±3.4, ±3.2 and ±2.3 ppm for Sensors 1, 2 and 3, respectively, which are the uncertainties of the measurement results with the HONEYWELL sensors. The uncertainty indicated the sensor provides better accuracy than the manufacturer’s claim (±5 ppm). Therefore, the calibration demonstrated the acceptable linearity of the conversion equation between voltage signal of the sensor and the NH$_3$ concentration, so that in future application, the calibration can be simplified to a two points calibration which only require 0 and 100 ppm NH$_3$ to calibrate the zero and full scale reading of the sensor. In addition, the calibration results are utilized in the 48 h laboratory test for calculating the NH$_3$ concentration.
Figure 4.2 Linear Regression of H_EC_1

Figure 4.3 Conversion Equation of H_EC_1
### Table 4.2 Specifics of calibration

<table>
<thead>
<tr>
<th>Specifics of</th>
<th>H_EC_1</th>
<th>H_EC_2</th>
<th>H_EC_3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linear Regression</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average voltage output of 0 ppm (V)</td>
<td>0.836 ± 0.107</td>
<td>0.838 ± 0.075</td>
<td>0.876 ± 0.078</td>
</tr>
<tr>
<td>Average change in voltage output of 25 (24.3) ppm (V)</td>
<td>0.882 ± 0.084</td>
<td>0.806 ± 0.057</td>
<td>0.797 ± 0.076</td>
</tr>
<tr>
<td>Average change in voltage output of 50 (54) ppm (V)</td>
<td>2.302 ± 0.150</td>
<td>2.252 ± 0.084</td>
<td>2.123 ± 0.082</td>
</tr>
<tr>
<td>Average change in voltage output of 100 (99.3) ppm (V)</td>
<td>3.869 ± 0.090</td>
<td>3.927 ± 0.011</td>
<td>3.898 ± 0.053</td>
</tr>
<tr>
<td>Linear regression slope (A (V/ppm))</td>
<td>0.0396</td>
<td>0.0403</td>
<td>0.0398</td>
</tr>
<tr>
<td>Linear regression intercept (B) (V)</td>
<td>0.007</td>
<td>-0.044</td>
<td>-0.064</td>
</tr>
<tr>
<td>Degree of freedom</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.9957</td>
<td>0.9962</td>
<td>0.9981</td>
</tr>
<tr>
<td>Standard error of linear regression (x</td>
<td>y) (V)</td>
<td>0.136</td>
<td>0.1301</td>
</tr>
<tr>
<td><strong>Conversion Equation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters^{b)}(1/A,B/A)</td>
<td>(25.253,0.178)</td>
<td>(24.814, -1.092)</td>
<td>(25.126, -1.608)</td>
</tr>
<tr>
<td>Standard error of conversion equation (y</td>
<td>x) (ppm)^{b)}</td>
<td>3.434</td>
<td>3.226</td>
</tr>
<tr>
<td>Nonlinearity error of 0 ppm (ppm)</td>
<td>-0.177</td>
<td>1.09</td>
<td>1.61</td>
</tr>
<tr>
<td>Nonlinearity error of 25 (24.3) ppm (ppm)</td>
<td>-2.159</td>
<td>-3.210</td>
<td>-2.660</td>
</tr>
<tr>
<td>Nonlinearity error of 50 (54) ppm (ppm)</td>
<td>3.790</td>
<td>2.390</td>
<td>0.640</td>
</tr>
<tr>
<td>Nonlinearity error of 100 (99.3) ppm (ppm)</td>
<td>-3.219</td>
<td>-2.460</td>
<td>-0.45</td>
</tr>
<tr>
<td>Overall nonlinearity error (ppm)</td>
<td>-0.437</td>
<td>-0.548</td>
<td>-0.215</td>
</tr>
</tbody>
</table>

^{a)} conversion equation: \[ X_{PPM} = 1/A \times [Y_{\Delta V} - B/A] \]

^{b)} \text{SE } x|y = \text{SE } y|x / A, the standard error of conversion equation estimates the uncertainty of the sensor.
4.2.2 Long-term Laboratory Test Results

The original signal output data from H_EC_1 was converted to NH$_3$ concentration based on the conversion equation from the sensor calibration, and plotted in Figure 44 with the equilibrium results in 48 h laboratory test of three sensors listed in Table 4.3. The difference between equilibrium results and reference NH$_3$ concentration are acceptable (equal or less than 5 ppm) with 100 and 25 ppm. However, the results of 50 ppm NH$_3$ measurement have differences which are out of acceptable range.

The first, middle and last two h measuring data were selected to evaluate the drift of signal output during long-term measurement, as shown in Figure 4.5. The equilibrium results of first, middle and last two h measurement were calculated and listed in Table 4.4. With the standard of drift and fluctuation defined in Section 3.2.2, the change of equilibrium results were classified in Table 4.5.

In Table 4.4, the first two h equilibrium results with 50 ppm NH$_3$ are all contained in 5 ppm acceptable range, which means the unacceptable 48 h equilibrium results might be caused by the drift and fluctuation. By analyzing the drift and fluctuation, the drift of H_EC_1 is contained in 5 ppm acceptable range, and the fluctuation of H_EC_2 &3 can be minimized with some signal smoothing filter (e.g. moving average filter) in future application. Moreover, the long-term measurement was an over load measurement, which means decreasing the measuring time and increasing the purging time in each cycle, as recommended by the manual (2.5 min measuring of NH$_3$), may possibly solve the problem of unacceptable difference. Therefore the sensor’s performance is still qualified in current stage.
### Table 4.3 Equilibrium results of 48 h test

<table>
<thead>
<tr>
<th>Equilibrium Result of H_EC_1 48 h test (ppm)</th>
<th>H_EC_2 48 h test (ppm)</th>
<th>H_EC_3 48 h test (ppm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 (99.3) ppm ±STD</td>
<td>97.6 ±4.8</td>
<td>98.2 ±1.0</td>
</tr>
<tr>
<td>Nonlinearity error of 100 ppm</td>
<td>-2.2</td>
<td>-3.2</td>
</tr>
<tr>
<td>50 (54) ppm ±STD</td>
<td>62.0 ±3.4</td>
<td>63.8 ±2.4</td>
</tr>
<tr>
<td>Nonlinearity error of 50 ppm</td>
<td>3.8</td>
<td>2.4</td>
</tr>
<tr>
<td>25 (24.3) ppm ±STD</td>
<td>20.6 ±1.6</td>
<td>21.6 ±0.6</td>
</tr>
<tr>
<td>Nonlinearity error of 25 ppm</td>
<td>-3.2</td>
<td>-2.4</td>
</tr>
</tbody>
</table>

### Table 4.4 Equilibrium results of first, mid and last two h test

<table>
<thead>
<tr>
<th>Equilibrium Result of H_EC_1 (ppm)</th>
<th>H_EC_2 (ppm)</th>
<th>H_EC_3 (ppm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 (99.3) ppm [0-2 h]</td>
<td>98.0</td>
<td>98.6</td>
</tr>
<tr>
<td>[23-25 h]</td>
<td>98.8</td>
<td>95.6</td>
</tr>
<tr>
<td>[46-58 h]</td>
<td>96.8</td>
<td>97.0</td>
</tr>
<tr>
<td>50 (54) ppm [0-2 h]</td>
<td>57.4</td>
<td>55.6</td>
</tr>
<tr>
<td>[23-25 h]</td>
<td>60.0</td>
<td>62.8</td>
</tr>
<tr>
<td>[46-48 h]</td>
<td>62.2</td>
<td>62.4</td>
</tr>
<tr>
<td>25 (24.3) ppm [0-2 h]</td>
<td>24.2</td>
<td>21.8</td>
</tr>
<tr>
<td>[23-25 h]</td>
<td>21.8</td>
<td>22.2</td>
</tr>
<tr>
<td>[46-48 h]</td>
<td>20.4</td>
<td>21.0</td>
</tr>
</tbody>
</table>

### Table 4.5 EC sensor’s drift and fluctuation in 48 h test

<table>
<thead>
<tr>
<th>Drift or Fluctuation</th>
<th>H_EC_1 (ppm/48 h)</th>
<th>H_EC_2 (ppm/48 h)</th>
<th>H_EC_3 (ppm/48 h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 (99.3) ppm</td>
<td>F</td>
<td>-1.6</td>
<td>F</td>
</tr>
<tr>
<td>50 (54) ppm</td>
<td>+4.8</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>25 (24.3) ppm</td>
<td>-3.8</td>
<td>F</td>
<td>-1.4</td>
</tr>
</tbody>
</table>

[a]: F - Fluctuation
Figure 4.4 48 h test results of H_EC_1
Figure 4.5 [0-2], [23-25], [46-48] h test results of H_EC_1
The plots in Figure 4.5 shows that, with 50 (54) and 100 (99.3) ppm NH₃, the sensor’s reading started to quickly decline at about 2-3 min after exposing to NH₃, and the last reading of each sampling period could be lower than 50% of the stable readings. The declination of the sensor’s reading demonstrated the saturation of the sensor after overtaking the limited exposing time (2.5 min, warned on the data sheet). However, with 25 (24.3) ppm NH₃, the declination or saturation is not as remarkable as the case with high level NH₃ concentration. Since the overall exposing time of the sensor to NH₃ is limited, accurately evaluating the response time of the sensor is necessary for estimating how many stable readings can be obtained under a constant data logging interval during each sampling period.

To apply the 1st order dynamic response theory as mentioned in Chapter 3, the sensor’s readings in first, middle and last one hour were selected, processed by dividing the 48 h equilibrium results and expressed as the response in percentage on Y-axis, as shown in Figure 4.6. The 5 τ response time was counted as the time before the response achieving 99.3%. If the maximum response is less than 99.3% (e.g. the first hour with 50 ppm NH₃ in Figure 4.5), the time before obtain first stable reading was counted as a replacement. The value of τ, response time with 95% response, overall response time can be obtained following the processes described in Section 3.2.2, and listed in Table 4.6. The overall response time with 95% response (3 τ) is about 1 min, which means the sensor can provide 1.5 min stable readings before expiring its limited measuring time to NH₃.
Table 4.6 EC sensor’s response time in 48 h test

<table>
<thead>
<tr>
<th>Response time of</th>
<th>H_EC_1 (sec)</th>
<th>H_EC_2 (sec)</th>
<th>H_EC_3 (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>* t = 3 τ</td>
<td>* t = 3 τ</td>
<td>* t = 3 τ</td>
</tr>
<tr>
<td>100 (99.3) ppm</td>
<td>60</td>
<td>56</td>
<td>58</td>
</tr>
<tr>
<td>[0-1 h]</td>
<td>120</td>
<td>90</td>
<td>60</td>
</tr>
<tr>
<td>[23-24 h]</td>
<td>60</td>
<td>60</td>
<td>58</td>
</tr>
<tr>
<td>[47-48 h]</td>
<td>60</td>
<td>90</td>
<td>60</td>
</tr>
<tr>
<td>[Average]</td>
<td>60</td>
<td>90</td>
<td>56</td>
</tr>
<tr>
<td>50 (54) ppm</td>
<td>60</td>
<td>90</td>
<td>60</td>
</tr>
<tr>
<td>[0-1 h]</td>
<td>~90</td>
<td>~60</td>
<td>58</td>
</tr>
<tr>
<td>[23-24 h]</td>
<td>60</td>
<td>90</td>
<td>60</td>
</tr>
<tr>
<td>[47-48 h]</td>
<td>60</td>
<td>90</td>
<td>56</td>
</tr>
<tr>
<td>[Average]</td>
<td>60</td>
<td>90</td>
<td>56</td>
</tr>
<tr>
<td>25 (24.3) ppm</td>
<td>60</td>
<td>90</td>
<td>56</td>
</tr>
<tr>
<td>[0-1 h]</td>
<td>150</td>
<td>180</td>
<td>58</td>
</tr>
<tr>
<td>[23-24 h]</td>
<td>60</td>
<td>120</td>
<td>56</td>
</tr>
<tr>
<td>[47-48 h]</td>
<td>60</td>
<td>120</td>
<td>56</td>
</tr>
<tr>
<td>[Average]</td>
<td>60</td>
<td>120</td>
<td>56</td>
</tr>
<tr>
<td>Average of 48 h test</td>
<td>60</td>
<td>100</td>
<td>56</td>
</tr>
<tr>
<td>Average time constant τ</td>
<td>20</td>
<td>18.667</td>
<td>18.667</td>
</tr>
</tbody>
</table>

[a] *“*:* an estimated response time of "t = 3 τ" based on the time constant τ of 48 h test;
[b] "~": the response time to achieve first stable reading, since the average of stable readings is lower than equilibrium result of 48 h test.

In summary, the further evaluation demonstrate the feasibility of the HONEYWELL EC FX NH₃ sensor with acceptable linearity, accuracy, drift (or fluctuation) and response time. The 8 min sampling with purging time (52 min) that was applied during 48 h laboratory test can provide basic scenario of periodic measurement, which means, in field application, PMU system can apply the NH₃ sensor with 8 min sampling time and 52 min purging time to obtain one measurement of the barn air quality in each hour. However, the studying was still continuing to find the best scenario which can keep the reliability and stability (acceptable drift and fluctuation) of the HONEYWELL EC FX sensor during 48 h test.
Figure 4.6 [0-1], [23-24], [47-48] h dynamic response of H_EC_1 (The red lines and points are the data that reach 99.3% of equilibrium results)
4.2.3 **Reliability Check**

The HONEYWELL EC FX NH$_3$ sensor had obtained acceptable evaluation results from the 48 h laboratory test, and was applied in the field test with PMU III system. However, since the field test was only performed with PMU III, which cannot compared the measurement results with other approved equipment (e.g. model 17C, Thermo-Environmental Instruments, Franklin, MA). Therefore, another 12 h laboratory test with same procedures and NH$_3$ gas of the 48 h laboratory test was performed after finishing the field test to check the consistency of the sensor’s performance, and to indirectly demonstrate the reliability of the sensor (H_EC_2 and H_EC_3) in the field test. The equilibrium test results of the sensor in each hour during the 12 h laboratory test were plotted in Figure 4.7. The average of these equilibrium test results were taken as the 12 h equilibrium test result, and be compared with the 48 h equilibrium test result as listed in Table 4.7. The differences between the 12 h and 48 h equilibrium test results are contained in the acceptable range (the error caused by the accuracy ±5 ppm). Moreover, another approach for evaluating the drift during the 12 h laboratory test is utilizing the hypothesis T test to demonstrate whether the drift can be neglected. Specifically, the data points plotted in Figure 4.7 were given linear regression, and the hypothesis T test was accomplished to check whether the slope of the linear regression result could be zero with a 95% confidence interval, so that the drifting of the measurement results during 12 h laboratory test can be zero and neglected. The results of hypothesis T test were also listed in Table 4.7. The drifting in four of the six 12 h laboratory test (H_EC_2 with 100 ppm NH$_3$, H_EC_3 with 25, 50 and 100 ppm NH$_3$) can be neglected, and the maximum drifting was –4.4 ppm during the 12 h laboratory test which is also contained in the acceptable range. Therefore the consistency of the sensor before and after the field test were demonstrated, and the reliability of the sensor in field test was indirectly approved.
**Table 4.7 Reliability Check**

<table>
<thead>
<tr>
<th>Analyzing results</th>
<th>H_EC_2</th>
<th>H_EC_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean of 100 (99.3) ppm (ppm) in 12 h test</td>
<td>98.0</td>
<td>101.00</td>
</tr>
<tr>
<td>Equilibrium of 100 (99.3) ppm in previous test (ppm)</td>
<td>98.2</td>
<td>100.0</td>
</tr>
<tr>
<td>Difference (ppm)</td>
<td>-0.2</td>
<td>+1.0</td>
</tr>
<tr>
<td>95% T test result[a] (reject H0 or H1)</td>
<td>H1</td>
<td>H1</td>
</tr>
<tr>
<td>Drift (Slope*Time ppm/12 h)</td>
<td>~0[b]</td>
<td>0</td>
</tr>
<tr>
<td>Mean of 50 ppm (54 ppm) in 12 h test</td>
<td>65.0</td>
<td>61.8</td>
</tr>
<tr>
<td>Equilibrium of 50 (54) ppm in previous test (ppm)</td>
<td>63.8</td>
<td>61.8</td>
</tr>
<tr>
<td>Difference (ppm)</td>
<td>+1.2</td>
<td>+0.0</td>
</tr>
<tr>
<td>95% T test result (reject H0 or H1)</td>
<td>H0</td>
<td>H1</td>
</tr>
<tr>
<td>Drift (Slope*Time ppm/12 h)</td>
<td>-4.4</td>
<td>0</td>
</tr>
<tr>
<td>Mean of 25 ppm (ppm) in 12 h test</td>
<td>24.2</td>
<td>22.8</td>
</tr>
<tr>
<td>Equilibrium of 24.3 (24.3) ppm in previous test (ppm)</td>
<td>21.6</td>
<td>21.8</td>
</tr>
<tr>
<td>Difference (ppm)</td>
<td>+2.6</td>
<td>+1.0</td>
</tr>
<tr>
<td>95% T test result (reject H0 or H1)</td>
<td>H0</td>
<td>H1</td>
</tr>
<tr>
<td>Drift (Slope*Time ppm/12 h)</td>
<td>-2.8</td>
<td>0</td>
</tr>
</tbody>
</table>

[a] Test H 0: Beta 1 = 0 vs. H 1: Beta 1 ≠ 0 at the 5% level of significance.
[b] The drift can be neglected based on the T test result.
Figure 4.7 The linear regression of 12 h reliability check
4.3 Field Test Results of PMU Evaluation

4.3.1 First Field Test with PMU III-1

The first version of PMU III (PMU III-1) was applied in the first field test to a commercial laying hen housing. The original data from NH$_3$, CO$_2$, temperature and pressure sensors were taken with 30 s data logging interval, downloaded from PMU system through wireless communication and plotted in Figure 4.8. The periodic raising and declining of the data from NH$_3$ and CO$_2$ sensors was the response of the sensor to periodic sampling (barn air) and purging (fresh air). The measuring results during purging time of the NH$_3$ sensor demonstrated the same status of the sensor after purging (back to 0 ppm) in each working cycle. As plotted in Figure 4.9, the processing of the field test data was performed by taking an average of the stable readings (same definition as the 48 h laboratory test) from NH$_3$ sensor in each sampling time, and taking the averages of the data from other sensors which corresponded to (in the same data string of) the stable readings from the NH$_3$ sensor. The mean of the processed NH$_3$ results from two HONEYWELL EC FX sensors were considered as the final processed result of NH$_3$ measurement.

The results from the first field test taken between Feb 9 and 11 showed that the upgraded PMU (PMU III-1) was successfully implementing systems including timer control, data centralization and data transform functions, which could reduce manual work with the upgraded system in future applications. However, some defects were found and modified as follows:

The expected measurement interval was 1 h (60 min, 3600 s) in the 48 h field test, which would obtain 5760 data with 30 s data logging interval (48 * 3600 / 30 = 5760). However, the measurement with PMU III-1 in 48 h field test only gained 4905 data, that the actual measurement interval is about 70 min (5760 / 4905 * 60 = 70.243). The error of the measurement interval may have been caused by an inaccurate virtual timer in the PMU III-1 program, which utilized the
Millisecond() function in the microprocessor as a reference to control the timeline of the system. When several functions (e.g. data logging and real-time data processing) in the program were being performed, and the Millisecond() function was periodically invoked between these functions, a cumulative delay would occur that led to a drift in the system time and an inaccurate time reference for the virtual timer. Therefore, a hardware real time clock (RTC) (DS3231, USPRO, www.amazon.com/shops/AT6S5L77ZENZI) was applied in PMU III-2 as the time reference for the virtual timer, which could independently record the time without being disturbed by other functions in the program.

The NH₃ measurements result from real-time data processing, and were compared with the post processed NH₃ measurement results (from Figure 4.9) and plotted in Figure 4.10. The results from real-time and post processing can provide similar varying tendency during the 48 h field test. However, some meaningless results (close to zero or negative during sampling time) from the real-time processing happened at about 8 and 36 h into the field test. The appearance of these meaningless results may be caused by the insufficient time for pre-sampling the sampling barn air to the sensor and waiting for the sensor’s response, which contained a large amount of meaningless sensor’s readings (measuring the air in the tube retained from purging or previous sampling) into the average calculation. As a solution, the second 3 way Solenoid Valve was applied to prevent the exposing of the NH₃ sensor to sampling air before enough pre-sampling time, as described in Chapter 3, and a response time was considered in the program to allow a 30 s response of the sensor without processing the data. Moreover, to collect more stable readings during the sampling time and save the space on SD card during the purging time, the data logging interval was modified from constant (30 s) to variable (1 s in stable reading time, 10 s in pre-sampling and response time, 600 s in purging time)
In addition, by comparing the processed results from the two HONEYWELL EC FX sensor in the first field test as shown in Figure 4.11, the high consistency between the two sensors (which can also be approved by analyzing the results in further evaluation) demonstrated that applying one HONEYWELL EC FX NH$_3$ sensor is feasible in future application. Therefore, only one NH$_3$ sensor was used in PMU III-2.

4.3.2 Second Field Test with PMU III-2

Following the same procedures as was done with the first field test, the PMU III-2 was applied to measure the air quality near the other two minimum ventilation fans of the laying hen housing in a second field test. The original data and processed results were plotted in Figure 4.12 and 6. The mean concentration of the second field test is about twice that of the first field test, which may be caused by backflow of ventilation air from the manure storage room, which elevated NH$_3$ in that section of the barn. The mean static pressure difference between animal building and outdoor atmosphere was about half of the result from the first field test, which may have been decreased by the back pressure from the manure storage room. The measured difference between the first and second field test demonstrated the sensitivity of PMU III system to the barn environment and could be applied for evaluating and improving the air quality in animal building.

Moreover, with the modification to the virtual timer of PMU III-1, PMU III-2 measured the barn air with precise 1 h sampling intervals and collected enough data during the 48 h field test. The real-time and post processed measurement results of NH$_3$ concentration were compared and plotted in Figure 4.14. The linear regression (Figure 4.15) between the real-time and post processed results showed an offset (intercept 11.6 ppm) which may be caused by the insufficiency response time. The response time of sensor should be at least 1 min based on laboratory evaluation, and the second field test only applied 30 s which was one of the sensor’s specifications claimed on the
manual (Response Time (T90): <30 s, 90% full scale). By adding the offset to the real-time processed results, the real-time and post processed results showed reasonable agreement. Therefore, increasing the waiting time for sensor’s response in the virtual timer can obtain reasonable real-time processing results in future applications.

In summary, although the PMU III could be further optimized in some aspects, the fundamental redesign and new sensor evaluation has been completed with PMU III-2, allowing for field studies of NH₃ and CO₂ concentration in animal housing.
Figure 4.8 The original data of first field test
Figure 4.9 The processed data of first field test
Figure 4.10 Comparison between real-time processed data and post processed data in first field test

Figure 4.11 Consistency between the processed NH₃ concentration from H_EC_1 and H_EC_2 in first field test
Figure 4.12 The original data of the second field test
Figure 4.13 The processed data of second field test
Figure 4.14 Comparison between real-time processed data and post processed data in second field test

Figure 4.15 Consistency between real-time and post processed data in second field test
CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary of the Project

In this project, the second generation PMU system (Gates et al., 2005) was upgraded by replacing the obsolete NH$_3$ EC sensor and developing a user friendly control system. To identify a suitable NH$_3$ EC sensor, an evaluation system was developed with specific criteria and procedures to test the performance of EC sensors, which focused on the response time of sensor. Calibration and further evaluation (48 h laboratory test with three levels of NH$_3$ concentration) was performed on the eligible EC sensor (HONEYWELL EC FX with approximately ±5 ppm accuracy and 5 min response time) to provide an assessment of its feasibility for use in field tests. The theory of 1$^{st}$ order dynamic response to unit step input was utilized to find the time constant and to estimate the reliable response time of the EC sensor.

From the results of these further evaluations, and the instructions from the sensor’s datasheet, the time intervals of sampling barn air and purging the sensor with fresh air were finalized as 360 s sampling corresponding to 3240 s purging in one working cycle. Based on the time control scenario, the virtual timer in the microprocessor program was designed to control the system. Moreover, other optimized functions were added to the existing PMU system by installing electrical components and microprocessor (Arduino Mega 2560). These optimized functions include RTC, LCD display, centralized data logging, wireless control, wireless data transform and real-time data processing algorithm for outputting NH$_3$ concentration after each sampling time.
The first version upgraded PMU (PMU III-1) was tested in a commercial laying hen house and modified to fix the defects in data loss and inaccurate real-time processing. The time for pre-sampling the barn air from the sampling points to the PMU system was considered in the program as a delay to expose the NH$_3$ sensor to the barn air and begin the measurement of NH$_3$ concentration. The second field test with the modified upgraded PMU (PMU III-2) gained enough data strings. By adding the offset (see Figure 4.7), the real-time processed data of NH$_3$ concentration from the modified system showed consistency with the post processed data (Slope = 0.94 ppm/ppm, R-squared = 0.88). The reliability of the NH$_3$ sensor in field test was checked by taking another 12 h laboratory test following the same procedures as the 48 h test. The results of the reliability test demonstrated consistency (difference is less than 5 ppm) of the applied EC sensors before and after the field test, which indicates the results of field test is reasonable. Based on the experience on designing and field application of the third version PMU, an SOP for instruction of utilizing this system in further studies is presented in APPENDIX D.

The overall material cost of the modified upgraded PMU system is about $4500, which is similar as the previous version. The upgrading fee is $1200 for replacing the NH$_3$ EC sensor and optimizing the control system.

5.2 Conclusions

In conclusion, this project achieved the objectives and tasks in Chapter 2 with following progresses:

(1) NH$_3$ EC sensor (HONEYWELL EC FX) was selected as the replacement of the original EC sensor with an acceptable response time and demonstrated stability (the uncertainty is about ±3 ppm and the maximum drift is about 4.8 ppm, which are all contained in the
manufacturer’s accuracy claim, ±5 ppm. The periodic measurement scenario of the HONEYWELL EC FX sensor in new PMU was established as 5.5 min sampling and 54.5 min purging.

(2) The DACS in second generation PMU system (Gates et al., 2005) was redesigned and upgraded with following optimizes:

a. The time settings of the system (i.e. sampling time, pre-sampling time, purging time and data logging interval) can be reset by applying a microprocessor (Arduino Mega 2560) with a design of “virtual timer” in the programming of the microprocessor (see Section 3.3.3).

b. The measurement results from the sensors in the system can be recorded by one SD card, and remotely downloaded by computer during the purging time, which was enabled by adding a Wireless SD shield to the microprocessor and two X-Bee radio transmitters.

c. The NH$_3$ concentration of the sampling barn air can be real-time processed by an algorithm designed for using in the microprocessor program.

(3) The third generation PMU (PMU III) was fabricated based on the protective enclosure and metal board form the second generation PMU (Gates et al., 2005) with reorganizing the circuit, tubing connection and components positions.

(4) Two field test were performed to evaluate the performance (the implement of optimized functions and the reliability of the replaced NH$_3$ EC sensor) of the PMU III system in field application, which demonstrated the feasibility of the PMU III for utilizing in future studying.
(5) A SOP was established and attached in APPENDIX D for operating the upgraded PMU in future applications.

5.3 **Recommendation for Future Work**

The PMU system is available for air quality studies in animal housing after the upgrading in this project; however, limited by the time issue, several aspects can still be improved in future work.

5.3.1 **Further Study on NH$_3$ EC Sensor**

More study on HONEYWELL NH$_3$ EC sensor can be taken following the same procedures in 48 h laboratory evaluation with less purging time to find the minimum measurement interval (with less purging time). If the sensor can stabilize (acceptable saturation or drift error) with 0.5 h purging time in each working cycle, it can capture twice the data compared with the current version. Additionally, other EC sensors which were not included in the sensor’s evaluation and identification may provide better performance than HONEYWELL EC sensor. Furthermore, performing the field test with other instruments (e.g. model 17C, Thermo-Environmental Instruments, Franklin, MA) and demonstrating the agreement between PMU and the reference instrument is necessary to practically evaluate the performance of upgraded PMU system.

5.3.2 **Further Optimization on PMU system**

The upgraded PMU (PMU III-2) can be still optimized in many aspects including:

1) Redesign the positions of different components, replace the current protective case with a smaller size and make the system more portable as shown in Figure 5.1.

2) Use more than one NH$_3$ EC sensor in each PMU to provide additional measurement results, even with the current measurement interval (1 NH$_3$ measurement result /h). For example,
in Figure 5.2, the two HONEYWELL FX EC sensors can measure NH₃ in rotation and provide a 0.5 h interval of measurement.

3) Redesign the tubing connections and add another separate air pump for measuring other air quality parameters (CO₂, temperature and pressure difference), these parameters can be continually measured without interruption caused by periodic purging of the NH₃ sensor.

---

**Figure 5.1 The design of PMU with small protective case**

---

**Figure 5.2 The design of PMU with two NH₃ sensors**
5.3.3 Enhance the Wireless Function

The wireless function of the upgraded PMU system can be extended by using a more powerful wireless module than X-Bee transmitter (e.g. mobile phone or radar). Thus a PMU management station can be developed to centralize and organize air quality measurements from multiple PMUs in the building or in the farm. Moreover, with a wireless module that can offer communication with satellites, a regional network of air quality detection can be established. The regional network (e.g. PMU network in Midwest) will allow the researcher download the real-time data from PMU without entering the farm. Ideally, if the regional network can cover enough buildings and farms in the measuring area, the overall data of gas pollutant emission can be analyzed by cooperating with the meteorological data in this area, and the air pollutants from animal industry in this area can be real-time estimated as a topographic map.
REFERENCES


APPENDIX A ARDUINO CODE FOR EC SENSOR EVALUATION SYSTEM

//EC sensor evaluation station program by Boyu
//Apply a EC sensor with 100 ppm full scale and 4-20mA(0.88-4.4V) signal output
//as an example. To apply the program for evaluating other sensors,
//details in the program should be modified.
#include <Wire.h>
#include <SD.h>
#include <SPI.h>
Sd2Card card;
SdVolume volume;
SdFile root;
int card_check = 0;
const int chipSelect = 4;//Load SD shield library and Define CS pin.
int data_interval = 10000; //The datalogging interval (milliseconds)
int dataNUM = 0;
int Sensor_pin = A2;
int Sensor_V = 0;
int Sensor_PPM = 0;
int Sensor_pre = 0;
in Sensor_cur = 0;
in SAMPLE = 0;
in STABLE = 0;
in DRIFT = 0;
in RELAY = 5;//Relay control,HIGH- sampling, LOW - purging;
#include <RTClib.h>
RTC_DS1307 RTC;//Load RTC library
String Timestring = "";
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);//Load LCD library.
String LCDstring = "";
void setup() {
  Serial.begin (9600);
  Serial.println("-Evaluation System-");
  Serial.println("---for EC Sensors---");
  Serial.println("-------by Boyu-------");
  Serial.println("----Version 1.0-----");
  lcd.init();lcd.backlight();//Initialization of LCD.
  LCDstring = "-Evaluation System-";lcd.setCursor(0,0);lcd.print(LCDstring);
  delay(500);
  LCDstring = "---for EC Sensors---";lcd.setCursor(0,1);lcd.print(LCDstring);
  delay(500);
  LCDstring = "-------by Boyu-------";lcd.setCursor(0,2);lcd.print(LCDstring);
  delay(500);
  LCDstring = "----Version 1.0-----";lcd.setCursor(0,3);lcd.print(LCDstring);
  delay(500);
  lcd.init();
  LCDstring = "_____Sys-Check_____";lcd.setCursor(0,0);lcd.print(LCDstring);
  Wire.begin();
  //Check SD Card
  //Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
  pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
  Serial.println("Checking SD Card... ");
  LCDstring = "Checking SD Card...";lcd.setCursor(0,1);lcd.print(LCDstring);
  delay(2000);
  while(card_check == 0) {
    if (!SD.begin(chipSelect)) {
      Serial.println("Card failed, or not present.@_@");
      Serial.println("initialization failed. Things to check:");
      Serial.println("* is a card is inserted?");
      Serial.println("* Is your wiring correct?");
      }
Serial.println("* did you change the chipSelect pin to match your shield or module?");
  LCDstring = "Card failed!   ";
  delay(1000);
  }
  else
  {
    card_check = 1;
    Serial.println("Card initialized.^-^-"); 
    LCDstring = "Card initialized!";
    delay(1000);
  }
  }
lcd.setCursor(0,1);lcd.print(LCDstring);

void loop() {
  Timeinformation();
  lcd.init();
  int AnalogREAD;
  AnalogREAD = analogRead(Sensor_pin);
  Sensor_V = AnalogREAD*5/1023;
  LCDstring = "Voltage = "; LCDstring += Sensor_V;
lcd.setCursor(0,0);lcd.print(LCDstring);
  Serial.print(LCDstring); Serial.print(" ");
  Sensor_PPM = (Sensor_V - 0.88)/(4.4-0.88)*100;
  LCDstring = "PPM = "; LCDstring += Sensor_PPM;
lcd.setCursor(0,1);lcd.print(LCDstring);
  Serial.println(LCDstring);
  if(millis()%10000 == 0)
  {dataNUM = dataNUM +1;
    DATALOGGING();
  }
  Sensor_cur = Sensor_PPM;
  if(abs(Sensor_cur - Sensor_pre) <0.5 & SAMPLE == 0)
  {
    LCDstring = "Steady NOW";lcd.setCursor(0,2);lcd.print(LCDstring);
    Serial.print(LCDstring); Serial.print(" ,");
    LCDstring = "[PLEASE SAMPLE]";lcd.setCursor(0,3);lcd.print(LCDstring);
    Serial.println(LCDstring);
    SAMPLE = 0;STABLE = 1;DRIFT = 0;}
  else if(abs(Sensor_cur - Sensor_pre) <0.5 & SAMPLE == 1 & STABLE == 0)
  {
    LCDstring = "Drift NOW";lcd.setCursor(0,2);lcd.print(LCDstring);
    Serial.print(LCDstring); Serial.print(" ,");
    LCDstring = "[PLEASE PURGE]";lcd.setCursor(0,3);lcd.print(LCDstring);
    Serial.println(LCDstring);
    SAMPLE = 0;STABLE = 0;DRIFT = 1;}
  else if(abs(Sensor_cur - Sensor_pre) >1 & SAMPLE == 1 & STABLE == 0)
  {
    LCDstring = "Drift NOW";lcd.setCursor(0,2);lcd.print(LCDstring);
    Serial.print(LCDstring); Serial.print(" ,");
    LCDstring = "[PLEASE PURGE]";lcd.setCursor(0,3);lcd.print(LCDstring);
    Serial.println(LCDstring);
    SAMPLE = 1;STABLE = 0;DRIFT = 1;}
  else if(SAMPLE == 1 & STABLE == 0 & DRIFT == 1)
  {
    LCDstring = "Drift NOW";lcd.setCursor(0,2);lcd.print(LCDstring);
    Serial.print(LCDstring); Serial.print(" ,");
    LCDstring = "[PLEASE PURGE]";lcd.setCursor(0,3);lcd.print(LCDstring);
    Serial.println(LCDstring);
    SAMPLE = 1;STABLE = 0;DRIFT = 1;}
  if(Serial.available() > 0 && Serial.read() == '0')
  {
    digitalWrite(RELAY,LOW);
    lcd.init();
    LCDstring = "PURGING...";lcd.setCursor(0,3);lcd.print(LCDstring);
    Serial.println(LCDstring);
    SAMPLE = 0; STABLE = 0; DRIFT = 0;"}
else if(Serial.available() > 0 && Serial.read() == '1')
{
digitalWrite(RELAY,HIGH);
  lcd.init();
  LCDstring = "SAMPLING...";lcd.setCursor(0,3);lcd.print(LCDstring);
  Serial.println(LCDstring);
  SAMPLE = 0; STABLE = 0; DRIFT = 0;
  delay(1000);
  Sensor_pre = Sensor_cur;
}

void DATALOGGING()
{ String dataString = "";
  Serial.print("Data Logging: ");
  dataString += String(dataNUM);
  dataString += ";
  dataString += Timestring;
  dataString += ";
  dataString += String(SAMPLE);
  dataString += ";
  dataString += String(STABLE);
  dataString += ";
  dataString += String(DRIFT);
  dataString += ";
  dataString += String(Sensor_PPM);
  // open the file
  File dataFile = SD.open("datalog.txt", FILE_WRITE);
  // if the file is available, write to it
  if (dataFile)
  {
    dataFile.println(dataString);
    dataFile.close(); // print to the serial port too:
    Serial.println(dataString);
    Serial.println("<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<");
    delay(500);
  } // if the file isn't open, pop up an error
  else
  {
    Serial.println("error opening datalog.txt");
    LCDstring = "....Datalog Error....";lcd.setCursor(0,3);lcd.print(LCDstring);
  }
}

void Timeinformation()
{ DateTime now = RTC.now(); //Read Time data from RTC
  Timestring += String(now.year(),DEC);
  Timestring += "/";
  if(now.month()<10)
  {
    Timestring += "0";
    Timestring += String(now.month(),DEC);
  }
  else
  {
    Timestring += String(now.month(),DEC);
  }
  Timestring += "/";
  if(now.day()<10)
  {
    Timestring += "0";
    Timestring += String(now.day(),DEC);
  }
  else
  {
    Timestring += String(now.day(),DEC);
  }
  Timestring += " ";
  if(now.hour()<10)
  {
    Timestring += "0";
  }
Timestring += String(now.hour(),DEC);
}
else
 {Timestring += String(now.hour(),DEC);
  Timestring += ":";
  if(now.minute()<10)
   {Timestring += "0";
    Timestring += String(now.minute(),DEC);
   }
  else
   {Timestring += String(now.minute(),DEC);
   }
  Timestring += ":";
  if(now.second()<10)
   {Timestring += "0";
    Timestring += String(now.second(),DEC);
   }
  else
   {Timestring += String(now.second(),DEC);
   }
  Timestring += ";"; //Combine data information into a Timestring for SD data and LCD display.
}
APPENDIX B SETTING PROFILE FOR X-BEE TRANSMITTER

X-BEE Transmitter Connected to PC

<?xml version="1.0" encoding="UTF-8"?>

<data>
  <profile>
    <description_file>XB24-B_ZigBee_1047.xml</description_file>
    <settings>
      <setting command="ID">1526</setting>
      <setting command="SC">FFFF</setting>
      <setting command="SD">3</setting>
      <setting command="NJ">FF</setting>
      <setting command="DH">13A200</setting>
      <setting command="DL">40B18ADD</setting>
      <setting command="ZA">0</setting>
      <setting command="SE">E8</setting>
      <setting command="DE">E8</setting>
      <setting command="CI">11</setting>
      <setting command="NI">0x20</setting>
      <setting command="BH">0</setting>
      <setting command="AR">FF</setting>
      <setting command="DD">30000</setting>
      <setting command="NT">3C</setting>
      <setting command="NO">0</setting>
      <setting command="PL">4</setting>
      <setting command="PM">1</setting>
      <setting command="EE">0</setting>
      <setting command="EO">0</setting>
      <setting command="KY">0</setting>
      <setting command="BD">7</setting>
      <setting command="NB">0</setting>
      <setting command="RO">3</setting>
      <setting command="D7">1</setting>
      <setting command="D6">0</setting>
      <setting command="SP">20</setting>
      <setting command="D0">1</setting>
      <setting command="D1">0</setting>
      <setting command="D2">0</setting>
      <setting command="D3">0</setting>
      <setting command="D4">0</setting>
      <setting command="D5">1</setting>
      <setting command="P0">1</setting>
      <setting command="P1">0</setting>
      <setting command="P2">0</setting>
  </settings>
</profile>
</data>
X-BEE Transmitter Connected to Arduino Microprocessor

<?xml version="1.0" encoding="UTF-8"?>

<data>
  <profile>
    <description_file>XB24-B_ZigBee_1247.xml</description_file>
    <settings>
      <setting command="ID">1526</setting>
      <setting command="SC">FFFF</setting>
      <setting command="SD">3</setting>
      <setting command="NJ">FF</setting>
      <setting command="DV">0</setting>
      <setting command="DH">13A200</setting>
      <setting command="DL">40B41271</setting>
      <setting command="ZA">0</setting>
      <setting command="SE">E8</setting>
      <setting command="DE">E8</setting>
      <setting command="CI">11</setting>
      <setting command="NI">0x20</setting>
      <setting command="BH">0</setting>
      <setting command="AR">FF</setting>
      <setting command="DD">30000</setting>
      <setting command="NT">3C</setting>
      <setting command="NO">0</setting>
      <setting command="PL">4</setting>
      <setting command="PM">1</setting>
      <setting command="EE">0</setting>
      <setting command="EO">0</setting>
      <setting command="KY"></setting>
      <setting command="BD">7</setting>
      <setting command="NB">0</setting>
      <setting command="RO">3</setting>
    </settings>
  </profile>
</data>
<setting command="D7">1</setting>
<setting command="D6">0</setting>
<setting command="SM">0</setting>
<setting command="ST">1388</setting>
<setting command="SP">20</setting>
<setting command="SN">1</setting>
<setting command="SO">0</setting>
<setting command="D0">1</setting>
<setting command="D1">0</setting>
<setting command="D2">0</setting>
<setting command="D3">0</setting>
<setting command="D4">0</setting>
<setting command="D5">1</setting>
<setting command="P0">1</setting>
<setting command="P1">0</setting>
<setting command="P2">0</setting>
<setting command="LT">0</setting>
<setting command="RP">28</setting>
<setting command="PR">1FFF</setting>
<setting command="IR">0</setting>
<setting command="IC">0</setting>
<setting command="V+">0</setting>
<setting command="CT">64</setting>
<setting command="GT">3E8</setting>
<setting command="CC">2B</setting>
</settings>
</profile>
</data>
APPENDIX C ARDUINO CODE FOR PMU III (IPMU V2)

//Intelligent Portable Monitoring Unit (IPMU) V2 R1 by Boyu for PMU system upgrading.
//1hour test interval
//No purge in sampling time, even collect stable data, since field test need more
//response time of the NH3 sensor.
//Wait for 30sec after mixing the gas in tube for sensor to response.
The sampling time will be 180 sec for mixing and 150 sec for testing.

//Intelligent Portable Monitoring Unit (IPMU) V2 R0 by Boyu for PMU system upgrading.
//Redesign the time control method to make it depending on RTC.
//Only apply one NH3_sensor, EEPROM function is given up;
//RTC will be used to start sampling Barn Air in each half hour.
//Another solenoid (II) is added to help NH3 sensor avoid "pre-saturation" problem in field test with long tubing.
The Barn Air sampling will be separated in to 3 section
//Section 1: RTC invoke Barn Air sampling by switching Solenoid I
//Section 2: Wait for 3min for mixing the gases in tube, Solenoid II close the gas line to NH3 sensor.
//Section 3: After 3min Solenoid II open the gas line to NH3 sensor, waiting for 10 stable readings with 1sec interval, switching to Purge.

//Intelligent Portable Monitoring Unit (IPMU) V1 R1 by Boyu for PMU system upgrading.
//Use 5min+25min for S/P
//Use 90sec mixing+160sec reading+50sec saturation
//Only Delta Filters

//Intelligent Portable Monitoring Unit (IPMU) V1 R0 by Boyu for PMU system upgrading.
//It still has some problem with SD initialization when reset the system, which will be
//solved in future, however, re-plug in can always solve the failed initialization issue.
// NH3 and CO2 concentration, Temperature and Pressure Difference between house and atmosphere
//will be allowed to test with this system.
//More wireless control functions are added in this version.
//Command list : ( Serial Command from the COM Window to Arduino Mega by USB or Micro)

//1-Blue Button on panel, "Yes" or "+".
//2-Red Button on panel, "No" or "-".
//3-Yellow Button on panel, for Time set.
//4-Data read function.
//5-Data delete function.
//6-Reset function.
//7-Skip purging function.
//8-Skip sampling function.

//Conversion between voltage and actual NH3 ppm depends on two FILTERs;

#include <EEPROM.h> //System self memory library.
//const int sampling_switch = 1;//address for recording sampling status.
//const int sampling_time = 2;//address for recording sampling time .
//const int data_memory = 3;//address for recording catalog number.
//const int Barn_time_memory = 4;//address for recording Barn_time.
//const int Purge_time_memory = 5;//address for recording Purge_time.
//const int Datalog_time_memory = 6;//address for recording Datalog_time.
//int system_status = 0;//sampling status status parameter.
//int check_time = 0;//check time parameter for EEPROM memory, 1 record/min.

int system_time = 0;//system time parameter.
int Timespan = 0;
int WaitforMIX = 180;
int Barn_time = 330;
int Purge_time = 3270;
const int Relay_SorP = 18; //Define the pin for Relays
const int Relay_NH3 = 19;
const int PurgeLED = 5;
const int SampLED = 6;//LED signal for purge and sampling.

#include <Wire.h>
#include <SD.h>
#include <SPI.h>
Sd2Card card;
SdVolume volume;
SdFile root;
int card_check = 0;
const int chipSelect = 4;//Load SD shield library and Define CS pin.
int IDrecord = 0;
int dataNUM = 0;//Data number.
int Datalog_Barn = 300;//Data log interval.
int Datalog_Stable = 1;
int Datalog_Fresh = 600;
int StableReading_NUM = 20;
int Relay_I = 0;
int Relay_II = 0;
int Sensortesting = 0;

#include <RTClib.h>
RTC_DS1307 RTC;//Load RTC library
String Timestring = "";
int hour2sec = 0;
int min2sec = 0;
int sec = 0;
long cur_RTC = 0;
long pre_RTC = 0;
int before_RTC = 1;
int check_interval = 0;

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);//Load LCD library.
String LCDstring = "";
int NH3_display = 1;

int button;
inetAddress button = 0;
int waitforchoose = 0;
int timechoose = 0;
inetAddress Dayset;
inetAddress Het;
inetAddress Minset;
inetAddress Secset;
int jump = 0; // for jump out of some setting loops, since Arduino will be confusing in some
// Time setting loops.

Serial and Wireless Control Parameters for Data download and other functions////////////
int USB_read;
int waitfor_datatransform = 0;

Voltage Reading and Calculations/////////////////////////////////////////////////

int NH3_analogpin = A3;
int CO2_analogpin = A5;
int Temp_analogpin = A9;
int Pressure_analogpin = A10;

///////////////////////////////////////////////////////////////////////////////////////////

double CO2_reading;
double CO2_voltage;
double CO2_PPM;

///////////////////////////////////////////////////////////////////////////////////////////

double NH3_reading;
double NH3_voltage;
double NH3_PPM;

///////////////////////////////////////////////////////////////////////////////////////////

double Temp_reading;
double Temp_voltage;
double Temp_resistor;
double Temp_oC;

///////////////////////////////////////////////////////////////////////////////////////////

double Pressure_reading;
double Pressure_voltage;
double Pressure_P;

///////////////////////////////////////////////////////////////////////////////////////////

Data Process////////////////////////////////////////////////////////////////////////////////

int Process_Status = 0; // 1 - for processing Sampling Data;
// 2 - for processing Purging Data.
double NH3_vo
ltage_pre = 0;
double NH3_voltage_cur = 0;
double NH3_sample_SUM = 0;
int NH3_sample_NUM = 0;
double NH3_sample_MEAN = 0.88;
double NH3_purge_SUM = 0;
int NH3_purge_NUM = 0;
double NH3_purge_MEAN = 0.88;
// double draft = 1;
int Sample_Data_init = 1;
int Purge_Data_init = 1;

SYSTEM SETUP {___________________________

void setup() {
  Serial.begin(115200);
lcd.init();lcd.backlight(); // Initialization of LCD.
  LCDstring = "-------IPMU V2-------";lcd.setCursor(0,1);lcd.print(LCDstring);
delay(1000);
  LCDstring = "-------by Boyu-------";lcd.setCursor(0,2);lcd.print(LCDstring);
delay(1000);
  Serial.println("IPMU_V2 by Boyu");
}
Serial.print("NH3_ID:##### ");Serial.print("CO2_ID:##### ");Serial.println("Pressure_ID:##### ");

lcd.init();

LCDstring = "NH3_ID:#####";lcd.setCursor(0,0);lcd.print(LCDstring);

LCDstring = "CO2_ID:#####";lcd.setCursor(0,1);lcd.print(LCDstring);

LCDstring = "Temp_ID:#####";lcd.setCursor(0,2);lcd.print(LCDstring);

LCDstring = "Pressure_ID:#####";lcd.setCursor(0,3);lcd.print(LCDstring);
delay(1000);

lcd.init();

LCDstring = "Reset time? Y/N...";lcd.setCursor(0,0);lcd.print(LCDstring);

int waittime = 0;

if(setcheck == 3)
{
goto setfinish;
}

Serial.println("Reset time? Y/N...Waiting...");

//The codes for printing on LCD
//will always be like an LCDstring with words or numbers, a Cursor setting function
to choose the print

//position, and a print function, just like this example.

LCDstring = "Waiting...";lcd.setCursor(0,1);lcd.print(LCDstring);
while(waittime<100)
{

    {USB_read = 0;
     USB_read= Serial.read();
     attachInterrupt(4,Plus,HIGH);attachInterrupt(5,Minor,HIGH);//wake up the buttons.
     if( Serial.available()&&USB_read>0&&USB_read == '1'| button == 1)
     {
         EEPROM.write(sampling_switch,0);
         setcheck = 0;
         button = 0;
         waittime=110;
     }
     else if( Serial.available()&&USB_read>0&&USB_read =='2'| button == 2)
     {
         setcheck = 1;
         Serial.println("Continue!");
         LCDstring = "Continue!";lcd.setCursor(0,2);lcd.print(LCDstring);
         waittime=110;
     }
     delay(100);
    waittime = waittime+1;
    LCDstring=String(10-waittime/10);LCDstring+="sec";
    if(10-waittime/10>=10)
    {
lcd.setCursor(11,1);
    }
    else
    {
        lcd.setCursor(11,1);lcd.print("0");
        lcd.setCursor(12,1);lcd.print(LCDstring);
        if(waittime==100)
        {
            LCDstring = "Continue...";lcd.setCursor(0,2);
            setcheck = 1;
        }
    }
}

button = 0;// initialize the button;

// Loop check point; Continue/Restart. Wait for 10sec, then go to next step.

//system_status = EEPROM.read(sampling_switch);

//switch (system_status)

//{case 0://sampling_switch = 0, which means Restart the loop, this is a manual
//restart operation.
    // Serial.println("New loop...");
    //LCDstring = "New loop...";
    //EEPROM.write(sampling_time,0);
    //EEPROM.write(data_memory,0);
    if(setcheck ==1)
else if(setcheck == 0)

SET TIME

{lcd.init();
 button = 0;
 Serial.println("Curent set is below:");
 LCDstring = "Curent set is below:";lcd.setCursor(0,0);lcd.print(LCDstring);
 Serial.print("Barn(sec): ");Serial.println(Barn_time);
 LCDstring = "Barn(sec): ";LCDstring += String(Barn_time);lcd.setCursor(0,1);lcd.print(LCDstring);
 Serial.print("Purge(sec): ");Serial.println(Purge_time);
 LCDstring = "Purge(sec): ";LCDstring += String(Purge_time);lcd.setCursor(0,2);lcd.print(LCDstring);
 Serial.print("WaitforMIX(sec): ");Serial.println(WaitforMIX);
 LCDstring = "WaitforMIX(sec): ";LCDstring += String(WaitforMIX);lcd.setCursor(0,3);lcd.print(LCDstring);
 delay(1000);
 lcd.init();
 Serial.println("Still reset? Y/N");
 LCDstring = "Still reset? Y/N";lcd.setCursor(0,0);lcd.print(LCDstring);
 while(waittime<200)
 {USB_read= Serial.read();
  if (USB_read == '1'| button == 1)
  {Serial.println("Set Time Now...");
   LCDstring = "Set Time Now...";lcd.setCursor(0,1);lcd.print(LCDstring);
   DateTime now = RTC.now();
   Dayset = now.day();
   Het = now.hour();
   Minset = now.minute();
   Secset = now.second();
   Timeset();
   waittime = 210;
  }
  else if (USB_read == '2'|button == 2)
  {Serial.println("Old options...");
   LCDstring = "Old options...";lcd.setCursor(0,1);lcd.print(LCDstring);
   LCDstring = "";
   setfinish://A track back after settime:///!
   waittime = 210;
  }
  delay(100);
  waittime = waittime+1;
  if( waittime == 200)
  {Serial.println("Time out for set...");
   LCDstring = "Time out for set...";lcd.setCursor(0,1);lcd.print(LCDstring);
   LCDstring = "use Old options...";lcd.setCursor(0,2);lcd.print(LCDstring);
   LCDstring = "New loop...";
  }
 }
//case 2://if sampling_swith = 2, Continue with sampling Fresh air.
//Serial.println("Old loop...Purging.");
//LCDstring = "Old loop...Purging.";
//delay(1500);
//break;
//}
lcd.setCursor(0,3);lcd.print(LCDstring);
lcd.init();
//Barn_time = EEPROM.read(Barn_time_memory)*20;
//Purge_time = EEPROM.read(Purge_time_memory)*20;
//Datalog_time = EEPROM.read(Datalog_time_memory)*20;//Initialize the Time span (sec) for Sampling/Purging/Data logging.
//EEPROM.write(Barn_time_memory, Barn_time/20);
//EEPROM.write(Purge_time_memory,Purge_time/20);
//EEPROM.write(Datalog_time_memory,Datalog_time);
Serial.print("Barn(sec): ");Serial.println(Barn_time);
LCDstring = "Barn(sec): ";LCDstring += String(Barn_time);lcd.setCursor(0,0);lcd.print(LCDstring);
Serial.print("Purge(sec): ");Serial.println(Purge_time);
LCDstring = "Purge(sec): ";LCDstring += String(Purge_time);lcd.setCursor(0,1);lcd.print(LCDstring);
Serial.print("WaitforMIX(sec): ");Serial.println(WaitforMIX);
LCDstring = "WaitforMIX(sec): ";LCDstring += String(WaitforMIX);lcd.setCursor(0,2);lcd.print(LCDstring);
//system_time = (EEPROM.read(sampling_time))*60+60;//get the value of systemtime from Arduino memory.
//dataNUM = EEPROM.read(data_memory);//get the value of datanumber from Arduino memory.
delay(2000);
Serial.println("___PMU Sys-Check____");
lcd.init();
LCDstring = "___PMU Sys-Check____";lcd.setCursor(0,0);lcd.print(LCDstring);
delay(2000);
Wire.begin();
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
Serial.print("Checking SD Card... ");
LCDS
//Check SD Card
//Wire connection for separate SD card with Mega 2560 are:CS-49,MISO-50.MOSI-51.SCK-52
pinMode(SS, OUTPUT);//This pin should not be wired, it is a shield.
LCDstring = "Checking RTC...";lcd.setCursor(0,2);lcd.print(LCDstring);
delay(2000);
if (! RTC.isrunning())
{Serial.println("RTC is NOT running! @_@");
LCDstring = "RTC isn't running!";
delay(1000);
// following line sets the RTC to the date & time this sketch was compiled
RTC.adjust(DateTime(__DATE__, __TIME__));
}
else
{Serial.println("and RTC is OKAY!! ^-^ ");
LCDstring = "RTC is OKAY!!";
delay(1000);
}
lcd.setCursor(0,2);lcd.print(LCDstring);
delay(2000);
Serial.println("____________________");
LCDstring = "Test will begin ...";lcd.setCursor(0,3);lcd.print(LCDstring);
lcd.init();
delay(2000);
pinMode(Relay_SorP,OUTPUT);
pinMode(Relay_NH3,OUTPUT);
pinMode(PurgeLED, OUTPUT);
pinMode(SampLED, OUTPUT);

//Stay in work loop after time set
int workready = 0;
while(workready <1)
{loop();
    detachInterrupt(4);detachInterrupt(5);detachInterrupt(0);//Cancel the buttons to avoid wrong manual operations.
} //a double check to start the main loop, since the setup function is too complicated
in this program.

SYSTEM SETUP

MAIN BODY 

void loop () {
    system_time = cur_RTC - pre_RTC;
    Timeinformation();//Real Time Clock.
    //Voltage Readings
    //-------------------------------
    CO2_reading = analogRead(CO2_analogpin);
    CO2_voltage = CO2_reading*5/1023;
    CO2_PPM = (CO2_voltage-0.88)*5000/(4.4-0.88);
    //Serial.print("CO2 v: ");Serial.print(CO2_PPM);
    //-------------------------------
    NH3_reading = analogRead(NH3_analogpin);
    NH3_voltage = NH3_reading*5/1023;
    //Serial.print(" NH3 v: ");Serial.print(NH3_voltage);
    //-------------------------------
    Temp_reading = analogRead(Temp_analogpin);
    Temp_voltage = Temp_reading*5/1023;
    Temp_resistor = 10*Temp_voltage/(5-Temp_voltage);
    Temp_oC = 1/(log(Temp_resistor/10)/4261+1/298.15)-273.15;
    //Serial.print(" T: ");Serial.print(Temp_oC);
    //-------------------------------
    Pressure_reading = analogRead(Pressure_analogpin);
    Pressure_voltage = Pressure_reading*5/1023;
    Pressure_P = (Pressure_voltage-0.88)*0.5*249.9/(4.4-0.88);
//Serial.print("P: "); Serial.println(Pressure_P);
//-------------------------------------------------------------------------------
-------------------
LCDscreen();
if(NH3_display<2)
{NH3_display = NH3_display + 1;
}
else
{NH3_display = 1;
}
if(system_time< Barn_time)
{//Timespan=Barn_time;
if(Sample_Data_init ==1)
{NH3_sample_SUM = 0; NH3_sample_NUM = 0;
Sample_Data_init = 0;
Purge_Data_init = 1;
}
Datalog_time = Datalog_Barn;//The datalog interval in Barn Air sampling time is
10sec.
LCDstring = "Barn Air Sampling...";lcd.setCursor(0,3);lcd.print(LCDstring);
// Relay_SorP  HIGH-Sampling
//         LOW-Purging
// Relay_NH3 HIGH-Bypass
//         LOW-NH3 sensor
Relay_I = 1;
Relay_II = 0;
Process_Status = 1;
if(system_time>WaitforMIX)
{digitalWrite(Relay_NH3,LOW);
digitalWrite(Relay_SorP,HIGH);
Relay_II = 1;
while(Sensortesting <30)
{Serial.print("Give 30 sec for sensor response ");
LCDstring = "<NH3 sensor warm up>";lcd.setCursor(0,3);lcd.print(LCDstring);
delay(1000);
Sensortesting = Sensortesting+1;
Serial.println(Sensortesting);
}
Serial.println("NH3 sensor is testing... ");
LCDstring = "<NH3 sensor testing">;lcd.setCursor(0,3);lcd.print(LCDstring);
Processing(Process_Status);
}
else if (system_time<=WaitforMIX)
{if((cur_RTC - pre_RTC)%10 == 0)
{Serial.println("Wait for mixing...");
digitalWrite(Relay_NH3,HIGH);
digitalWrite(Relay_SorP,HIGH);
Relay_II = 0;
}
LCDScreen = ".Wait for mixing...";lcd.setCursor(0,3);lcd.println(LCDScreen);
}
//if(NH3_sample_NUM>0&NH3_sample_NUM<10)
//{Datalog_time = Datalog_Stable;//The datalog interval will be 1sec during
stable reading time.
// Serial.println("Collect Stable Readings");
//}
//else if(NH3_sample_NUM>9)
//if((cur_RTC - pre_RTC)%10 == 0)
//{Serial.println("Collect 10 Stable Readings");
//LCDScreen = "Gain Stable Reads”;lcd.setCursor(0,3);lcd.print(LCDScreen);
//Serial.println("More purge is giving...");


//LCDstring = "More purge is giving";lcd.setCursor(0,3);lcd.print(LCDstring);
//}
//digitalWrite(Relay_NH3,LOW);digitalWrite(Relay_SorP,LOW);
//Relay_I = 0;
//Relay_II = 1;
//}
int r = random(430,440);
NH3 = r*0.01*draft;
//simulation voltage of 100 ppm

Barn air sampling.

if(system_time==Barn_time)
{
    Serial.println("_______________");
    EEPROM.write(sampling_switch,0);
    LCDstring = "Switch!";lcd.setCursor(0,3);lcd.print(LCDstring);

digitalWrite(Relay_SorP,LOW);digitalWrite(Relay_NH3,LOW);digitalWrite(SampLED,LOW);digitalWrite(PurgeLED,HIGH);
    Relay_I = 0;
    Relay_II = 1;
} //Switch Relay for purging.

if(system_time<Barn_time+Purge_time&&system_time>Barn_time)
{
    if(Purge_Data_init ==1)
    {
        NH3_purge_SUM = 0; NH3_purge_NUM = 0;
        Sample_Data_init = 1;
        Purge_Data_init = 0;
    }

    Sensortesting = 0;
    Datalog_time = Datalog_Fresh;//The datalog interval in Purging time is 10min.
    LCDstring = "Fresh Air Purging...";lcd.setCursor(0,3);lcd.print(LCDstring);

digitalWrite(Relay_SorP,LOW);digitalWrite(Relay_NH3,LOW);digitalWrite(SampLED,LOW);digitalWrite(PurgeLED,HIGH);
    Relay_I = 0;
    Relay_II = 1;
} //Switch Relay for purging.

if(system_time==Purge_time + Barn_time)
{
    system_time = 0;
    //Timespan=Barn_time;
    EEPROM.write(sampling_switch,1);
    Serial.println("_______________");
    LCDstring = "Switch!";lcd.setCursor(0,3);lcd.print(LCDstring);

digitalWrite(Relay_SorP,LOW);digitalWrite(Relay_NH3,LOW);digitalWrite(SampLED,HIGH);digitalWrite(PurgeLED,LOW);
} //Switch Relay for sampling.

if(system_time%Datalog_time==0)
{
    dataNUM = dataNUM+1;
    //if(EEPROM.read(sampling_switch)==1)
// {Serial.println("|Barn Air| ");}
//else
// {Serial.println("|Fresh Air|");}
//EEPROM.write(data_memory, dataNUM);
//////Wait for MIX and Stop for Saturation//////
Serial.print("NH3 Stable Readings Num= "); Serial.println(NH3_sample_NUM);
Serial.print(" Estimate PPM= "); Serial.println(NH3_PPM);
datalog(); //Datalogging.
LCDstring = "....Datalogging....."; lcd.setCursor(0,3);lcd.print(LCDstring);
// if(system_time<=Barn_time)
// {draft = draft - 0.04; }//simulation of draft
// else
// {draft = 1;}

// It is time for data logging.

// if(check_interval % 60==0)
// {check_time = int(((cur_RTC - pre_RTC))/60);
// EEPROM.write(sampling_time, check_time);
// Serial.print("Time memory = ");Serial.println(Timespan-check_time*60);
// Serial.println("<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<");
// LCDstring = "/\Do Self-memory//";lcd.setCursor(0,3);lcd.print(LCDstring);
// } // It is time for Arduino self memory to avoid work status losing if the system
// is reset by accident.
while(check_interval <1000)
{check_interval = check_interval + 100;
 Wireless(); //Xbee wireless control and data transform.
 USB_read = 0;
delay(100); //a 100 system dely will be sensitive enough for maunal controls.
}  
Timestring =""; //Refresh the Timestring for next sec Time data logging.
check_interval = 0; //Refresh the check_interval 
cur_RTC = hour2sec + min2sec + sec;

if(before_RTC ==1)
{pre_RTC = cur_RTC;
 before_RTC = 0;
}

if(cur_RTC<pre_RTC)
{pre_RTC = cur_RTC - system_time;
}
if(cur_RTC-pre_RTC < Barn_time)
{if((cur_RTC - pre_RTC)%10 == 0)
{Serial.print("Pre: "); Serial.print(pre_RTC); Serial.print(" Cur: ");
Serial.println(cur_RTC);
Serial.print("RTC timer in Barn Sampling "); Serial.println(Barn_time - (cur_RTC-pre_RTC));
//EEPROM.write(sampling_switch,1);
}
}
else if(cur_RTC - pre_RTC >= Barn_time + Purge_time)
{pre_RTC = cur_RTC;
}
else
{if((cur_RTC - pre_RTC)%10 == 0)
{Serial.print("Pre: "); Serial.print(pre_RTC); Serial.print(" Cur: ");
Serial.println(cur_RTC);
Serial.print("RTC timer in Fresh Purging "); Serial.println(Barn_time + Purge_time - (cur_RTC-pre_RTC));
//EEPROM.write(sampling_switch,0);
}
}  } MAIN BODY } MAIN BODY   //
void datalog()
{
    String dataString = "";
    Serial.print("Data Logging: ");
    dataString += String(dataNUM);
    dataString += ",");
    // append unix timestamp to dataString
    dataString += Timestring;
    // read a thermistor append to the string
    dataString += "|";
    if(Relay_I == 1)//Record of the on/off of Relay for Purge or Barn Air sampling
    {dataString += "1,";
    }
    else
    {dataString += "0,";
    }
    if(Relay_II == 1)//Record of the on/off of Relay for NH3 sensor
    {dataString += "1,";
    }
    else
    {dataString += "0,";
    }
    dataString += String(StableReading_NUM);
    dataString += "|");
    dataString += String(NH3_voltage);
    dataString += "|");
    dataString += String(NH3_PPM);
    dataString += "|");
    dataString += String(CO2_PPM);
    dataString += "|");
    dataString += String(Temp_oC);
    dataString += "|");
    dataString += String(Pressure_P);
    dataString += ";");
    // open the file
    File dataFile = SD.open("datalog.txt", FILE_WRITE);
    // if the file is available, write to it
    if (dataFile)
    {while(IDrecord < 1)
        {dataFile.println("NH3_ID:#####  CO2_ID:#####  Temp_ID:#####
Pressure_ID:######");
        IDrecord = 1;
        }
        dataFile.println(dataString);
        dataFile.close(); // print to the serial port too:
        Serial.println(dataString);
        Serial.println(">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>");
        delay(500);
        } // if the file isn't open, pop up an error
    else
    {Serial.println("error opening datalog.txt");
        LCDstring = "....Datalog Error...."; lcd.setCursor(0,3);lcd.print(LCDstring);
        }
    }
}

void dataread()
{
    File dataFile = SD.open("datalog.txt", FILE_READ); // if the file is available, write
to it
    if (dataFile)// read from the file until there's nothing else in it:
    {String Data_read;
        while (dataFile.available())
            {Serial.write(dataFile.read());
                delay(1);
                }
                // close the file:
            dataFile.close();
            Serial.println("---------Done---------");
            LCDstring = "---------Done---------";lcd.setCursor(0,3);lcd.print(LCDstring);
            }
waitfor_datatransform = 1;

// if the file isn't open, pop up an error
else
  {Serial.println("error opening datalog.txt");
  }
}

void LCDscreen()
{
  LCDstring = Timestring;lcd.setCursor(0,0);lcd.print(LCDstring);
  LCDstring = " |CO2 |ToC|Ppa";lcd.setCursor(0,1);lcd.print(LCDstring);

  switch (NH3_display)
  {
    case 1:
      lcd.setCursor(0,1); lcd.print("NH3_V ");
      LCDstring = " ";
      LCDstring += String(NH3_voltage);
      break;
    case 2:
      lcd.setCursor(0,1); lcd.print("NH3_C ");
      if(NH3_PPM <0)
        {LCDstring = " Err!";}
      else if(NH3_PPM<10)
        {LCDstring = " ";LCDstring += String(int(abs(NH3_PPM))));}
      else if(NH3_PPM<100)
        {LCDstring = " ";LCDstring += String(int(abs(NH3_PPM))));
      else if(NH3_PPM<1000)
        {LCDstring = " ";LCDstring += String(int(abs(NH3_PPM))));
      break;
  }
  LCDstring += " |";
  if(CO2_PPM<0)
    {LCDstring += "Err!";}
  else if(CO2_PPM<10)
    {LCDstring += " ";LCDstring += String(int(CO2_PPM)));
  else if(CO2_PPM<100)
    {LCDstring += " ";LCDstring += String(int(CO2_PPM)));
  else if(CO2_PPM<1000)
    {LCDstring += " ";LCDstring += String(int(CO2_PPM)));
  else
    {LCDstring += String(int(CO2_PPM))}
  LCDstring += " |";
  if(Temp_oC<0)
    {LCDstring += "Err";}
  else if(Temp_oC<10)
    {LCDstring += " ";LCDstring += String(int(Temp_oC)));
  else if(Temp_oC<100)
    {LCDstring += " ";LCDstring += String(int(Temp_oC)));
  else
    {LCDstring += " ";LCDstring += String(int(Temp_oC)));
  LCDstring += " |";
  if(Pressure_P<10)
    {LCDstring += String(int(Pressure_P)));
  else if(Pressure_P<0)
    {LCDstring += " ";LCDstring += String(int(Pressure_P)));
  else if(Pressure_P<10)
    {LCDstring += String(int(Pressure_P)));
  else if(Pressure_P<100)
    {LCDstring += " ";LCDstring += String(int(Pressure_P)));
  else
    {LCDstring += String(int(Pressure_P))}
  lcd.setCursor(0,2);
  lcd.print(LCDstring);
}


void Processing(int i)
{
    Wireless();
    if (i == 1) // Processing in Sample time
    {
        NH3_voltage_cur = NH3_voltage;
        if (NH3_voltage_cur - NH3_voltage_pre < 0.022 && WaitforMIX > 2 && system_time < Barn_time) // If the difference of two voltage readings is less than 0.022v is will be regard as a stable reading.
        {
            NH3_sample_SUM = NH3_sample_SUM + NH3_voltage_cur;
            NH3_sample_NUM = NH3_sample_NUM + 1;
            NH3_sample_MEAN = NH3_sample_SUM / NH3_sample_NUM;
            NH3_PPM = (NH3_sample_MEAN - NH3_purge_MEAN) * 24.3 - 1.09;
            Datalog_time = Datalog_Stable;
            StableReading_NUM = 1;
        }
        else
        {
            NH3_sample_SUM = NH3_sample_SUM;
            NH3_sample_NUM = NH3_sample_NUM;
            NH3_sample_MEAN = NH3_sample_MEAN;
            StableReading_NUM = 0;
        }
        NH3_voltage_pre = NH3_voltage_cur;
    }
    else if (i == 2) // Processing in Purge time
    {
        NH3_voltage_cur = NH3_voltage;
        NH3_purge_SUM = NH3_purge_SUM + NH3_voltage;
        NH3_purge_NUM = NH3_purge_NUM + 1;
        NH3_purge_MEAN = NH3_purge_SUM / NH3_purge_NUM;
    }
}

void Change()
{
    int waitforchange = 0;
    while (waitforchange < 1)
    {
        USB_read = Serial.read();
        /////////////////"+"////////////////
        if (USB_read == '1' || button == 1)
        {
            button = 0;
            switch (timechoose)
            {
                delay(500);
                case 1:
                    Dayset = Dayset + 1;
                    RTC.set(RTC_DAY, Dayset);
                    Serial.print(Dayset);
                    Serial.print(\"\?");
                    LCDstring = String(Dayset);
                    break;
                case 2:
                    Het = Het + 1;
                    RTC.set(RTC_HOUR, Het);
                    Serial.print(Het);
                    Serial.print(\"\?");
                    LCDstring = String(Het);
                    break;
                case 3:
                    Minset = Minset + 1;
                    RTC.set(RTC_MINUTE, Minset);
                    Serial.print(Minset);
                    Serial.print(\"\?");
                    LCDstring = String(Minset);
                    break;
            }
        }
    }
}
case 4:
Secset = Secset+5;
RTC.set(RTC_SECOND,Secset);
Serial.print(Secset);
Serial.print("? ");
LCDstring = String(Secset);
case 5:
Barn_time = Barn_time+60;
delay(500);
Serial.print(Barn_time);
Serial.print("? ");
LCDstring = String(Barn_time);
break;
case 6:
Purge_time = Purge_time+60;
delay(500);
Serial.print(Purge_time);
Serial.print("? ");
LCDstring = String(Purge_time);
break;
case 7:
WaitforMIX = WaitforMIX+10;
delay(500);
Serial.print(WaitforMIX);
Serial.print("? ");
LCDstring = String(WaitforMIX);
break;
}
LCDstring += "?";
lcd.setCursor(0,3);
lcd.print(LCDstring);
delay(500);
button = 0;
}
//----------"-"/----------
else if (USB_read=='2'|button == 2){button = 0;
switch(timechoose)
{delay(500);
case 1:
Dayset = Dayset-1;
RTC.set(RTC_DAY,Dayset);
Serial.print(Dayset);
Serial.print("? ");
LCDstring = String(Dayset);
break;
case 2:
Het = Het-1;
RTC.set(RTC_HOUR,Het);
Serial.print(Het);
Serial.print("? ");
LCDstring = String(Het);
break;
case 3:
Minset = Minset-1;
RTC.set(RTC_MINUTE,Minset);
Serial.print(Minset);
Serial.print("? ");
LCDstring = String(Minset);
break;
case 4:
Secset = Secset-5;
RTC.set(RTC_SECOND,Secset);
Serial.print(Secset);
Serial.print("? ");
LCDstring = String(Secset);
case 5:
    Barn_time = Barn_time-60;
    delay(500);
    Serial.print(Barn_time);
    Serial.print("? ");
    LCDstring = String(Barn_time);
    break;
    case 6:
    Purge_time = Purge_time-60;
    delay(500);
    Serial.print(Purge_time);
    Serial.print("? ");
    LCDstring = String(Purge_time);
    break;
    case 7:
    WaitforMIX = WaitforMIX-10;
    delay(500);
    Serial.print(WaitforMIX);
    Serial.print("? ");
    LCDstring = String(WaitforMIX);
    break;
}
    LCDstring += "?";
    lcd.setCursor(0,3);
    lcd.print(LCDstring);
    delay(500);
    button = 0;

if(USB_read=='3')
{
    Serial.print("Next...");
    LCDstring = "Next...";
    lcd.setCursor(0,3);
    lcd.print(LCDstring);
    TimeSet();
}
if(jump == 1)
{
    waitforchange = 1;
    Serial.println("Next...");
    LCDstring = "Next...";
    lcd.setCursor(0,3);
    lcd.print(LCDstring);
    button = 0;
    jump = 0;
}

void Timeinformation()
{
    DateTime now = RTC.now();//Read Time data from RTC
    Timestring += String(now.year(),DEC);
    hour2sec = now.hour();
    hour2sec = hour2sec*3600;
    min2sec = now.minute()*60;
sec = now.second();
Timestring += "/";
if(now.month()<10)
{
    Timestring += "0";
    Timestring += String(now.month(),DEC);
}
else
Timestring += String(now.month(),DEC);
if(now.day()<10)
{Timestring += "0";
Timestring += String(now.day(),DEC);
} else
{Timestring += String(now.day(),DEC);
}
Timestring += "/";
if(now.hour()<10)
{Timestring += "0";
Timestring += String(now.hour(),DEC);
} else
{Timestring += String(now.hour(),DEC);
}
Timestring += ":";
if(now.minute()<10)
{Timestring += "0";
Timestring += String(now.minute(),DEC);
} else
{Timestring += String(now.minute(),DEC);
}
Timestring += ":";
if(now.second()<10)
{Timestring += "0";
Timestring += String(now.second(),DEC);
} else
{Timestring += String(now.second(),DEC);
}
Timestring += " "; //Combine data information into a Timestring for SD data and LCD display.

void Timeset()
{
  DateTime now = RTC.now();
  button = 0;
  Serial.println(" ");
  LCDstring = " ";
  lcd.setCursor(0,3);
  lcd.print(LCDstring);
  while(waitforchoose<10)
  {
    USB_read= Serial.read();
    if(timechoose== 8)
    {timechoose = 0;
    }
    attachInterrupt(0,Set,RISING);
    timeset:
    if (USB_read == '3'|button ==3)
    {timechoose = timechoose +1;
      Serial.print(timechoose);
      delay(500);
      reswitch:
      switch(timechoose)
      {case 1:
        Serial.print(" Set DAY: ");
        Serial.print( now.day());
        Serial.print( to: ");
        lcd.init();
        break;
        case 2:
        break;
        case 3:
        break;
        case 4:
        break;
        case 5:
        break;
        case 6:
        break;
        case 7:
        break;
        case 8:
        break;
        case 9:
        break;
      default:
        break;
LCDstring = "Set DAY: ";
lcd.setCursor(0,0);
lcd.print(LCDstring);
LCDstring = String(now.day());
lcd.setCursor(0,1);
lcd.print(LCDstring);
LCDstring = " to: ";
lcd.setCursor(0,2);
lcd.print(LCDstring);
jump = 0;
Change();
delay(500);
button = 0;
break;
case 2:
  Serial.print(" Set HOUR: ");
  Serial.print( now.hour());
  Serial.print(" to: ");
  lcd.init();
  LCDstring = "Set HOUR: ";
lcd.setCursor(0,0);
lcd.print(LCDstring);
LCDstring = String(now.hour());
lcd.setCursor(0,1);
lcd.print(LCDstring);
LCDstring = " to: ";
lcd.setCursor(0,2);
lcd.print(LCDstring);
jump = 0;
Change();
delay(500);
button = 0;
break;
case 3:
  Serial.print(" Set MIN: ");
  Serial.print( now.minute());
  Serial.print(" to: ");
  lcd.init();
  LCDstring = "Set MIN: ";
lcd.setCursor(0,0);
lcd.print(LCDstring);
LCDstring = String(now.minute());
lcd.setCursor(0,1);
lcd.print(LCDstring);
LCDstring = " to: ";
lcd.setCursor(0,2);
lcd.print(LCDstring);
jump = 0;
Change();
delay(500);
button = 0;
break;
case 4:
  Serial.print(" Set SEC(5s +/-): ");
  Serial.print( now.second());
  Serial.print(" to: ");
  lcd.init();
  LCDstring = "Set SEC(5s +/-): ";
lcd.setCursor(0,0);
lcd.print(LCDstring);
LCDstring = String(now.second());
lcd.setCursor(0,1);
lcd.print(LCDstring);
LCDstring = " to: ";
lcd.setCursor(0,2);
lcd.print(LCDstring);
jump = 0;
Change();
delay(500);
button = 0;
break;
case 5:
Serial.print(" Set Barn_time: ");
Serial.print( Barn_time);
Serial.print(" to: ");
lcd.init();
LCDstring = "Set Barn_time: ";
lcd.setCursor(0,0);
lcd.print(LCDstring);
LCDstring = String(Barn_time);
lcd.setCursor(0,1);
lcd.print(LCDstring);
LCDstring = " to: ";
lcd.setCursor(0,2);
lcd.print(LCDstring);
jump = 0;
Change();
delay(500);
button = 0;
break;
case 6:
Serial.print(" Set Purge_time:");
Serial.print( Purge_time);
Serial.print(" to: ");
lcd.init();
LCDstring = "Set Purge_time: ";
lcd.setCursor(0,0);
lcd.print(LCDstring);
LCDstring = String(Purge_time);
lcd.setCursor(0,1);
lcd.print(LCDstring);
LCDstring = " to: ";
lcd.setCursor(0,2);
lcd.print(LCDstring);
jump = 0;
Change();
delay(500);
button = 0;
break;
case 7:
Serial.print(" Set WaitforMIX:"));
Serial.print(WaitforMIX);
Serial.print(" to: ");
lcd.init();
LCDstring = "Set WaitforMIX: ");
lcd.setCursor(0,0);
lcd.print(LCDstring);
LCDstring = String(WaitforMIX);
lcd.setCursor(0,1);
lcd.print(LCDstring);
LCDstring = " to: ");
lcd.setCursor(0,2);
lcd.print(LCDstring);
jump = 0;
Change();
delay(500);
button = 0;
break;
case 8:
  button = 0;
  Serial.println(" Exit?:");
  lcd.init();
  LCDstring = "Exit?";
  lcd.setCursor(0,0);
  lcd.print(LCDstring);
  for(waitforchoose=0;waitforchoose<10;waitforchoose++)
    {int restforwait = 10 - waitforchoose;
     Serial.println(restforwait);
     LCDstring = String(restforwait -1);
     LCDstring += " sec";
     lcd.setCursor(0,1);
     lcd.print(LCDstring);
     USB_read = 0;
     USB_read= Serial.read();
     if(USB_read == '3'| button ==3)
      {button = 0;
       timechoose = 0;
       LCDstring = "Back to Day set";
       lcd.setCursor(0,3);
       lcd.print(LCDstring);
       goto reswitch;
      } break;
    }
  delay(1000);
  const int waitforchoose = 11;
  LCDstring = "Back to sys-check";
  delay(1000);
}
Serial.println("Back to sys-check");
delay(1000);
setcheck = 3;
setup();

void Wireless()
{
  USB_read= Serial.read();
  if(Serial.available()>0 & USB_read == '4')//Wireless Read
    {delay(100);
     if (system_time>Barn_time)
       {Serial.println("Data transform:");
        LCDstring = "---Data transform:--";lcd.setCursor(0,3);lcd.print(LCDstring);
        while(waitfor_datatransform<1)
          {dataread();//Datareading.
           }
       } else
         {Serial.println("Please Wait for Purging...");
          }
    }
  if(Serial.available()>0 & USB_read == '5' & waitfor_datatransform == 1)//Delete old file
    {Serial.println("Removing datalog.txt...");
     SD.remove("datalog.txt");
     if (SD.exists("datalog.txt"))
       {Serial.println("ERROR: datalog.txt still exists.");
        }
else
    {Serial.println("DONE: datalog.txt has been removed.");
        Serial.println("System will be reset in 5 sec");
        delay(5000);
        asm volatile (" jmp 0");
    }

else if (Serial.available()>0 & USB_read == 'D' & waitfor_datatransform == 0)
    {Serial.println("Datafile has not been downloaded before removing, please print R for downloading");
        if(Serial.available()>0 & USB_read == '6') /// Manual Reset
            {Serial.println("Manual Reset");
                LCDstring = "/\Manual Reset/\";lcd.setCursor(0,3);lcd.print(LCDstring);
                delay(1000);
                asm volatile (" jmp 0");
            }
if(Serial.available()>0 & USB_read == '7') /// Manual Skip Purging
    {Serial.println("Manual Skip P");
        Serial.println(" ");
        LCDstring = "/\Manual SKIP P/\";lcd.setCursor(0,3);lcd.print(LCDstring);
        delay(1000);
        if(system_time>Barn_time)
            {system_time = 0;
                pre_RTC = cur_RTC;
            }
    }
if(Serial.available()>0 & USB_read == '8') /// Manual Skip Sampling
    {Serial.println("Manual Skip S");
        Serial.println(" ");
        LCDstring = "/\Manual SKIP S/\";lcd.setCursor(0,3);lcd.print(LCDstring);
        delay(1000);
        if(system_time<=Barn_time)
            {system_time = Barn_time;
                pre_RTC = cur_RTC-Barn_time;
            }
    }
}
APPENDIX D SOPs FOR PMU III

HONEYWELL EC FX Calibration

1) Add a 220 ohm resistor between GND and SIG ports on the sensor’s the signal processing circuit;

2) Power on the sensor with 24 VDC, warm up the sensor for 1 hour without any operations;

3) Use the VDC mode of voltage meter and place the “+” lead on GND and “-” lead on SIG port on the signal processing circuit.

4) Expose the sensor to zero gas (normal air or pure nitrogen) with calibration adaptor (EC-FX-CA) and 0.3 L/min flow rate until the reading on voltage meter is at 0.88 V ± 0.05 V, which means the sensor has automatically calibrated to zero gas;

5) Press the button for 3 s to start calibration mode of the sensor with green LED slowly flashing;

6) Switch the tube on the calibration adaptor to expose the sensor to full-scale gas (100 ppm NH₃ mixed with pure nitrogen) with 0.3L/min flow rate until the rising of output voltage is equal or less than 0.02 V/sec. Then immediately begin adjusting the sensor by turning the span pot to adjust the full-scale voltage output close to 4.4V;

7) Even though, the voltage reading will keep change after adjusting to 4.4 V, remove the adaptor and stop exposing the sensor to NH₃. (Note: the time of exposing the sensor to full-scale NH₃ gas must be limited in 2.5 min. If the sensor’s voltage output begin decreasing with 0.04 V/sec which means the saturation and accuracy drift has happened, step 2 to 10 should be repeated.)

8) Set up the sensor into EC sensor evaluation system (Section 3.1.2) and periodically expose the sensor to full-scale NH₃ gas (100 ppm) for 2.5 min and purging the sensor with zero
gas for 57.5 min, do at least 3 cycles of sampling and purging to collect the voltage data within 10 s interval;

9) Choose 3 peak values from each sampling time as the stable readings of full-scale gas, and take the average of the 9 stable readings from three cycles as the adjusted voltage reading of full-scale NH₃ gas (100 ppm), also take the average of the data in purging time as the adjusted voltage reading of zero gas (normal air);

10) Take linear regression with two points data, \( X = [0 \text{ ppm}, 100 \text{ ppm}] \) and \( Y = [0, \text{ adjusted voltage reading of full-scale NH}_3 \text{ gas - adjusted voltage reading of zero gas }] \) and derive the conversion equation (transform voltage reading to NH₃ gas concentration);

11) Repeat step 8) with 50 ppm NH₃ gas, then calculate the adjusted voltage reading of 50 ppm NH₃ gas with same method in step 9), and utilize the conversion equation from step 10) to calculate the NH₃ gas concentration. If the difference between calculated NH₃ concentration and the 50 ppm is contained in the error caused by the accuracy (±5 ppm), the calibration is accomplished. Otherwise the linearity of the HONEYWELL EC sensor may have been obsolete.

**X-BEE Transmitter Setup**

1) Download XCTU software from the manufacturer’s website and open the software; 

2) Click on “Discover radio modules connected to your machine”;

3) Choose the correct USB Serial Port (check 9600, 19200 and 115200 baud rate) and click “Finish”;

4) Wait for the software to discover the X-Bee transmitter and click “Add selected devices”;
5) Click the LOGO of the discovered transmitter to open the setting page;

6) Click on “Update firmware”, select ZNET 2.5 ROUTER/EC DEVICE AT firmware for the X-Bee transmitter (A) which will be connected to computer USB shield, and select ZNET 2.5 COORDINATOR AT for the transmitter (B) which will be connected to Wireless SD shield, then click “Finish”;

7) Click on “Load” and choose the profile “X-BEE Transmitter Connected to PC” for transmitter A and the profile “X-BEE Transmitter Connected to Arduino Microprocessor” for transmitter B, then confirm the loading;

8) Modify the DH and DL of transmitter A with the SH and SL of transmitter B, and modify the DH and DL of transmitter B with the SH and SL of transmitter A;

9) Click “Write radio setting” to finish the setup.

**Arduino Program Upload**

1) Download Arduino IDE (Version 1.5.5 is applied for this project, other version needs modify with the libraries) from website and open the software;  
   (http://arduino.cc/en/Main/Software)

2) Open the code file “IPMU V2” which has been pasted in APPENDIX C;

3) Connect the microprocessor to computer with USB data cable (if the Wireless SD shield is plugged on microprocessor, the button on the shield must be switch to “USB” side for uploading the program)

4) Click on "Tools", set the "Board" to "Arduino Mega or Mega 2560", set "Processor" to "ATmega2560 (Mega 2560)", set the "Port" to correct USB serial port of Arduino USB cable.
5) Click on “Upload” to upload the program (code) to the microprocessor (if the Wireless SD shield is plugged on microprocessor, the button on the shield must be switch to “Micro” side after uploading the program).

Instruction of PMU III operations in Field Application

1) Connect the “SAMPLE” and “HIGH” inlet port of PMU III to the sampling point of animal building with TEFLO\textsuperscript{N} tube;
2) Connect the “PURGE” and “LOW” inlet port of PMU III to the purging point of outdoor atmosphere with TEFLO\textsuperscript{N} tube;
3) Connect the “EXHAUST” outlet port of PMU III to the outdoor atmosphere with TEFLO\textsuperscript{N} tube;
4) Dust filet must be connected to the end of each sampling, purging or exhausting tube;
5) Power the PMU III system with 120 VAC;
6) Set the flowrate as 9-11 L/min on the bypass line and 0.5 L/min on the sensor line (to NH\textsubscript{3} and CO\textsubscript{2} sensor);
7) Connect the X-Bee transmitter with USB shield to the computer and open the COM Window of Arduino IDE software with 115200 baud rate to communicate with the IPMU control box;
8) Reset the time intervals or calibrate the RTC by typing “1” in the input box of the COM Window and confirm with “Enter” (“2” to skip this step, or the system will wait for 10 s and automatically pass this step);
9) Type in “3” and confirm with “Enter” to switch between the setting options (1-day, 2-hour, 3-min, 4-second, 5-sampling time, 6-pre-sampling time, 7-purging time and 8-exit);
10) Type in “1” and confirm with “Enter” to increase the value of the setting (“2” to decrease);

11) The system will start field measurement after finishing or skipping the reset and calibration function (The connection of RTC module and SD card will be initialized before starting the field measurement, if the initialization of SD card is failed, it need to be re-plugin to the port on Wireless SD shield);

12) Type “4” and confirm with “Enter” to download data from the SD card, which is only available during the purging time;

13) Type “5” and confirm with “Enter” to delete the previous data on SD card, which is only available after downloading the previous data;

14) Type “6” and confirm with “Enter” to interrupt the working cycle of the system and reset the system; (“7” to skip the sampling time and “8” to skip the purging time);

15) Finish the field test by power off the PMU system and disconnect the tubes.
APPENDIX E PLOTS AND FIGURES

Figure E.1 Calibration data of H_EC_2

Figure E.2 Linear Regression of H_EC_2
Figure E.3 Conversion Equation of H_EC_2

Figure E.4 Calibration data of H_EC_3
Figure E.5 Linear Regression of H_EC_3

Figure E.6 Conversion Equation of H_EC_3
Figure E.7 48 h test results of H_EC_2
Figure E.8 48 h test results of H_EC_3
Figure E.9 [0-2], [23-25], [46-48] h test results of H_EC_2
Figure E.10 [0-2], [23-25], [46-48] h test results of H_EC_3
Figure E.11 [0-1], [23-24], [47-48] h dynamic response of H_EC_2 (The red lines and points are the data that reach 99.3% of equilibrium results)