Electrical Capacitance Measurements to Assess European Corn Borer Infestation in Maize

Mavrik D. Thomas
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Engineering Commons

BYU ScholarsArchive Citation
Thomas, Mavrik D., "Electrical Capacitance Measurements to Assess European Corn Borer Infestation in Maize" (2021). Theses and Dissertations. 9661.
https://scholarsarchive.byu.edu/etd/9661

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Electrical Capacitance Measurements to Assess European Corn Borer Infestation in Maize

Mavrik D. Thomas

A thesis submitted to the faculty of Brigham Young University in partial fulfillment of the requirements for the degree of Master of Science

Brian Mazzeo, Chair
Douglas Cook
Brian Jeffs

Department of Electrical and Computer Engineering Brigham Young University

Copyright © 2021 Mavrik D. Thomas All Rights Reserved
ABSTRACT

Electrical Capacitance Measurements to Assess European Corn Borer Infestation in Maize

Mavrik D. Thomas
Department of Electrical and Computer Engineering, BYU
Master of Science

The European Corn Borer (ECB), *Ostrinia nubilalis*, is an agricultural pest which bores small holes in the rind of maize stalks and then proceeds to consume the pith. Because most of the damage to the stalk is internal, it is difficult to quantify the damage to an individual stalk without time-consuming, manual examination. This work explored the hypothesis that internal damage could be detected and quantified using non-destructive, electromagnetic measurements. Laboratory experiments and numerical simulation studies predicted changes in capacitance of stalks due to hollow core ECB damage. A guarded probe device to measure electrical impedance from 500 Hz to 100 kHz was designed and constructed for data collection. A field test with the measurement device was conducted and frequency-swept impedance measurements were taken on field-grown plants with and without ECB damage. Field measurements demonstrated that statistically significant capacitance changes associated with ECB damage could be detected in agreement with numerical simulations of stalk damage. Numerical, laboratory, and field test results all supported the hypothesis that electromagnetic impedance measurements, in particular, capacitance, provide a promising new avenue for ECB damage evaluation. While further research will be needed to further refine this concept, this measurement approach is non-destructive, thus allowing measurements to be performed without sacrificing the infested plants.

Keywords: European Corn Borer, capacitance, impedance, maize, nondestructive testing, agriculture
ACKNOWLEDGEMENTS

The work presented in this thesis could not have happened without the help and support of a number of remarkable individuals. I would like to thank my advisor Dr. Brian Mazzeo for his invaluable help and support. I have had the privilege of working with him for the last four years on a few different research projects and his mentorship during this time has been formative to my education. I also express my gratitude to Dr. Douglas Cook. He was the one who presented to me the opportunity to work on the project that would become my thesis. His expertise and support were invaluable to the success of the project. I would like to thank Brandon Sutherland for providing a consistent supply of plants for lab testing throughout the project and helping in preparations for the greenhouse tests.

I acknowledge and thank the people at Corteva Agriscience for support and funding this project. I would like to thank Timothy Nowatzki and Tim Moriarty from Corteva for setting up field tests and collecting field data.

Lastly, I would like to acknowledge my family and friends. I am profoundly grateful for their love, support, and encouragement as I endeavored to complete my thesis.
TABLE OF CONTENTS

ABSTRACT .................................................................................................................................... II

TABLE OF CONTENTS .............................................................................................................. IV

LIST OF TABLES ........................................................................................................................ VI

LIST OF FIGURES ..................................................................................................................... VII

CHAPTER 1. INTRODUCTION .............................................................................................. 1
  1.1 The European Corn Borer ........................................................................................... 1
  1.2 Nondestructive Evaluation of ECB Damage ......................................................... 3

CHAPTER 2. INITIAL DEVICE DEVELOPMENT AND EXPERIMENTS ......................... 6
  2.1 Initial Hypothesis about Permittivity, ECB Damage, and Capacitance ............... 6
  2.2 Initial Multichannel Impedance Measurement Prototypes .................................... 7
  2.3 Greenhouse Prototype and Testing ......................................................................... 10
  2.4 Initial Maize Stalk Permittivity Estimation ............................................................. 15

CHAPTER 3. CAPACITANCE MEASUREMENT DEVICE TO ASSESS ECB DAMAGE IN MAIZE 18
  3.1 Introduction ................................................................................................................ 18
  3.2 Background .............................................................................................................. 20
  3.3 Laboratory Experiments and Numerical Simulations .............................................. 21
  3.4 Physical 3D Cylinder Model – Hollow PTFE Rod Measurements ...................... 21
  3.5 Maize Stalk Permittivity Measurements ................................................................. 25
  3.6 Parameter Sensitivity Analysis by Numerical Simulation of Hollow Maize Stalks .... 25
  3.7 Prototype Measurement Device ............................................................................. 31
     3.7.1 Measurement Apparatus ................................................................................. 31
     3.7.2 Analog Discovery & Interface Circuitry ....................................................... 32
     3.7.3 Electrodes ...................................................................................................... 33
     3.7.4 Physical Apparatus ....................................................................................... 34
     3.7.5 Device Calibration & Measurement Model ................................................... 36
  3.8 Field Methods .......................................................................................................... 40
     3.8.1 Field Plants .................................................................................................... 41
     3.8.2 Measurements ............................................................................................... 42
  3.9 Field Results ............................................................................................................ 43
  3.10 Discussion ............................................................................................................... 45
  3.11 Conclusion ............................................................................................................. 49

CHAPTER 4. CONCLUSION AND FUTURE WORK .......................................................... 50
LIST OF TABLES

Table 3-1: Summary statistics of maize plants in field experiment. Details about ECB damage are given. ............................................................................................................................... 44

Table 3-2: Statistical comparisons between intact measurement sites and ECB damaged sites for the average parameter values of each population. ................................................................. 46
LIST OF FIGURES

Figure 1-1: ECB adult female (photo courtesy of University of Nebraska–Lincoln Department of Entomology, reproduced with permission) [6]........................................................................ 1

Figure 1-2: ECB larval tunneling through the inside of a maize stalk (photo courtesy of University of Nebraska–Lincoln Department of Entomology, reproduced with permission) [16]........................................................................................................................................... 2

Figure 1-3: Maize stalks split longitudinally to show ECB damage (photo courtesy of Corteva Agriscience, reproduced with permission) [17]. ..................................................................... 3

Figure 2-1: Initial measurement concept. Multiple electrodes placed on both sides of the stalk allow electric fields (yellow) to interrogate different parts of the stalk which could then be combined to create a capacitance map of the stalk................................................................. 8

Figure 2-2: Various electrode arrays used for the initial multichannel prototypes...................... 10

Figure 2-3: A measurement being taken by one of the initial multichannel prototypes............. 9

Figure 2-4: Single pair of electrodes which shows the guard ring on one electrode to reduce the influence of stray fields on the measurement................................................. 11

Figure 2-5: Prototype setup used in greenhouse testing (first milestone).................................. 12

Figure 2-6: Greenhouse results comparing conductance (top) and capacitance (bottom) between intact and damaged measurement locations. The difference of means at each frequency index is shown along with the 95% confidence interval. .......................................................... 14

Figure 2-7: Dry pith cubes used to estimate stalk permittivity.................................................. 16

Figure 3-1: Cross-section view of a maize stalk. ECB damage is characterized by replacement of water-rich pith material with air and frass, altering the permittivity and internal material arrangement. Capacitance measured across the stalk decreases........................................... 20

Figure 3-2: Laboratory and simulation setup for PTFE capacitance measurements. (a) CAD model showing the locations of the guarded electrode (GE), guard ring (GR), and counter electrode (CE) in relation to the rod. (b) The measurement stage with cutouts showing the dimensions of the GE, GR and CE....................................................................................... 23

Figure 3-3: Simulated and measured capacitance values at 5.02 kHz for PTFE rods with increasing hole size, represented as a ratio to stalk diameter. Capacitance values are presented as a ratio to intact rod capacitance. Error bars for measured values are 95% confidence intervals.............................................................................................................. 24

Figure 3-4: Simulated maize stalk model. The dimensions for the guard ring (GR), guarded electrode (GE), counter electrode (CE), and the guarded back......................................................... 26

Figure 3-5: Cross section of maize stalk with parallel electrodes. Simulated parameters are: (a) ECB size, (b) electrode offset, and (c) contact gap. .............................................................. 27

Figure 3-6: Simulated capacitance values for maize stalks with ECB damage from three simulation sets. (a) Capacitance for maize stalks with diameters ranging from 12mm to 24mm electrode offset from stalk. (b) Capacitance for increasing electrode offset from the maize stalk. (c) Capacitance for increasing electrode contact gap from stalk........................................... 29

Figure 3-7: Stalk measurement apparatus mounted on a 25.4mm diameter wood dowel. The trigger actuates the spring-loaded plunger to open a space between the electrode guides. When the trigger is released, the electrode guides close and clamp onto the dowel and center...
the electrodes in the process. The electrodes are pressed against the dowel with small springs to ensure good contact with the dowel.

Figure 3-8: Circuit diagram of measurement apparatus.

Figure 3-9: Stalk measurement PCBs. Top PCB with CE. Bottom PCB with GE and GR.

Figure 3-10: Clamping mechanism. (a) Electrode PCBs connected to holders with springs. (b) Electrode guides with glue on edges. (c) Bottom view of apparatus showing the spring-loaded plunger.

Figure 3-11: Comparison of $N_{b}(\omega)$ to $Y_{\text{measured}}(\omega)$ for various capacitor (left) and resistor (right) loads. The resistors values from 223kΩ to 18.2MΩ are shown, and all capacitor values from 1pF to 59.3pF are shown.

Figure 3-12: The measurement model used to characterize measurement taken with our apparatus. The baseline noise $N_{b}(\omega)$ in our measurements is assumed to be additive. By estimating $N_{b}(\omega)$, we can estimate values for $L_{s}$, $G_{p}$, and $C_{p}$.

Figure 3-13: Model estimated conductance (left) and capacitance (right) compared to known values of resistors and capacitors, respectively. The error bars show the 95% confidence intervals for all measurements (most are so small they are not visible).

Figure 3-14: Photograph of representative set of split maize stalks, with some stalks showing ECB damage (dark areas).

Figure 3-15: Un-parameterized, measured capacitance and conductance values on plants from the field. Each point represents an average capacitance and conductance value measured at a single frequency. Red points correspond to damaged measurement locations and blue points correspond to intact locations. The average parameter fit for each population is shown by the dashed lines: cyan for intact locations and orange for damaged locations.

Figure A-1: Assembly diagram for measurement stage.

Figure A-2: Diagram showing multiple views of the bottom platform.

Figure A-3: Dimensions for the bottom platform.

Figure A-4: Diagram showing multiple views of the top platform.

Figure A-5: Dimension for the top platform.

Figure A-6: Dimensions for the measurement stage CE PCB.

Figure A-7: Manufacturing schematic for the measurement stage CE PCB.

Figure A-8: Dimensions for the measurement stage GE and GR PCB.

Figure A-9: Manufacturing schematic for the measurement stage CE PCB.
CHAPTER 1.  INTRODUCTION

1.1  The European Corn Borer

Maize (*Zea mays*) is an important global cereal crop used in food products, animal feed, and fuel production. In 2019, about 197 million hectares of maize were grown globally, of which, about 52 million hectares were grown in Europe and North America [1]. One challenge to growing maize in Europe and North America is the European Corn Borer, *Ostrinia nubilalis* (ECB). Native to Europe, the ECB is a member of the Crambidae family in the Lepidoptera order of insects (Figure 1-1). The ECB was introduced to North America in the early twentieth century and has since spread across the Midwest [2, 3]. In the United States alone, the ECB is estimated to cause $1-2 billion dollars of loss to maize crops [4, 5].

![Figure 1-1: ECB adult female (photo courtesy of University of Nebraska–Lincoln Department of Entomology, reproduced with permission) [6]](image-url)
ECB larvae feed on and bore into the leaves, ears, and stalks of maize, causing extensive damage to maize plants (Figure 1-2). Plants damaged by ECB tunneling are vulnerable to secondary pathogen infections, such as a fungal infection [6]. A few strategies have been employed to mitigate the impact of the ECB. Traditional methods of pest management have utilized strategic crop planting and harvesting schedules [7, 8], insecticide application [7, 8], and the use of pest-resistant varieties such as Bt maize [5, 8-11]. One challenge to all management strategies is tracking ECB infestation throughout the growing season [5]. Current methods rely on estimating infestation based on ECB larval and moth sample counts [12, 13]. Furthermore, evaluating the effectiveness of pest-resistant varieties has required quantification of ECB damage [10, 14, 15].

Figure 1-2: ECB larval tunneling through the inside of a maize stalk (photo courtesy of University of Nebraska–Lincoln Department of Entomology, reproduced with permission) [16].
Since ECB damage primarily occurs inside of corn stalks and is rarely visible from the outside, current evaluation methods are destructive in nature and require significant manual labor [14, 15]. Such methods commonly entail cutting open maize plants and visually measuring the extent of ECB damage. These destructive measurements are limited in their ability to track ECB damage over time and can only provide a limited view to how the ECB affects plants and crops. An objective, rapid, nondestructive evaluation method for detecting and quantifying ECB damage would be a valuable new tool that would aid in the development of maize varieties that are resistant to ECB infestation.

Figure 1-3: Maize stalks split longitudinally to show ECB damage (photo courtesy of Corteva Agriscience, reproduced with permission) [17].

1.2 Nondestructive Evaluation of ECB Damage

Only a small amount of work has been done in developing nondestructive methods for evaluating ECB damage. One study used X-ray imaging to map ECB cavities within maize
plants, but cost and field deployment concerns were limiting factors for this method [18].

Another study looked at detecting larvae in the roots of sugarcane [19]. Electrical impedance measurements have been used to nondestructively estimate the root mass size of maize plants [20, 21]. These studies exploited known physiological changes and dimensions in plants to develop nondestructive test methods.

Since ECB damage induces physical changes to maize plant structure, there is a possibility that electrical impedance measurements can be used to evaluate ECB damage without damaging the plants. An ideal implementation of a nondestructive impedance method requires a portable setup that can take rapid in situ measurements and accurately quantify ECB damage for maize plants in the field. The work presented in this thesis explores the hypothesis that electrical impedance, particularly electrical capacitance, can be used to quantify ECB damage in maize. Additionally, the presented work seeks to demonstrate proof of concept by developing and deploying a portable impedance analyzer designed to rapidly interrogate the electrical impedance of maize stalks.

One year was allotted by the project sponsor to conduct this study and develop a prototype measurement device. During this time, development on a prototype started from scratch and proceeded at quick pace. Numerical simulations and laboratory experiments were also performed to explore theoretical impedance changes that would result from ECB damage. The findings from these simulations and experiments were used to aid the development of the measurement device.

Two testing milestones were set up to evaluate progress towards the study’s objectives. The first test milestone occurred four months from the start of the program and entailed deploying an initial prototype to test greenhouse grown maize plants at the sponsor’s facilities.
The second milestone was scheduled for six months from the start of the project and called for a prototype to be deployed and used in the field on field-grown plants. While unforeseen difficulties, including COVID-19 and weather damaging the crops to be tested, prevented the exact execution of this plan, the general schedule was followed during this project.

Ultimately, this study explored and verified the hypothesis linking ECB damage and electrical impedance. Prototypes were deployed to take measurements on maize plants and demonstrated the proof of concept by measuring a statistically significant difference in capacitance between intact sections of maize stalk and sections damaged by ECB. Chapter 2 details initial device development and some of the challenges and questions that arose in the process. Chapter 3 presents the work that will be published from this study: the exploration of the relationship between ECB damage and impedance through lab tests and numerical simulations, the final development of a measurement device, and the results and findings from field testing. Chapter 4 is a concluding chapter that summarizes the work and outlines some possible future directions for this work.
CHAPTER 2. INITIAL DEVICE DEVELOPMENT AND EXPERIMENTS

The work presented in this chapter highlights key research foundations and initial experiments that culminated in the final, working device and accompanying numerical simulations that are extensively detailed and presented in Chapter 3. When this research began, there was a need to explore many different approaches to sensing electromagnetic impedance changes and there was also a pressing need to produce a device capable of measurement quickly, as the 2020 maize growing season in North America would provide the only limited window of measurement opportunity for field-grown maize. Additionally, due to COVID travel restrictions that were unforeseen when the project started, measurements would have to be performed by Corteva personnel in Iowa. Thus, there was a rapid exploratory and building phase at the outset of this project that, subsequently, was followed by a more deliberate approach that is detailed later. Additionally, to perform needed numerical simulations, initial permittivity estimates of maize stalk samples were also conducted. Because of the unique execution of this project, a number of technical details are excluded from this chapter because they are detailed in Chapter 3 or because they are no longer relevant to the work as a whole.

2.1 Initial Hypothesis about Permittivity, ECB Damage, and Capacitance

The starting point of this project was a hypothesis about how material property changes due to ECB damage might enable a non-destructive electromagnetic assessment of maize condition. The relative permittivity of a material reflects its ability to polarize when an electric
field is applied. Essentially, polarization is charge displacement due to the applied electric field and different materials will polarize to varying degrees. Some materials, like water, polarize easily. Others, like air, do not.

The ratio of electric charge accumulation due to an applied electric potential between two or more conductors is termed capacitance and is dependent on the relative permittivity of the material between the conductors. Since capacitance depends mainly on physical parameters, such as conductor separation and surface area, as well as the electrical permittivity of the medium between the conductors, capacitance measurements can be used to quantify material changes that may be intrinsic in a homogeneous medium as well as rearrangements of heterogeneous material.

ECB damage is characterized by a change in stalk internal material structure due to larval tunneling. Water-rich, pith material is replaced with air and frass (larval excrement). Since water has a much higher relative permittivity than air (a factor of about 80), the working hypothesis is that this damage would result in a decrease in material permittivity and consequently lead to a measurable decrease in capacitance between probe conductors placed along or across an ECB damaged stalk portion as compared to an intact portion.

2.2 Initial Multichannel Impedance Measurement Prototypes

To test the hypothesis predicting a drop in capacitance in response to ECB damage, several initial prototypes were developed to measure the capacitance of maize stalks. While a professional-grade impedance analyzer is ideal in a laboratory setting, due to the field deployment requirements inherent to the project, a low-cost, low-power, portable unit was necessary for measurement prototypes. A USB-powered Analog Discovery 2 (AD2) (Digilent)
was selected to measure capacitance and conductance across a range of frequencies. The initial approach to quantifying ECB damage sought to use an array of electrodes to create a capacitance map for a small section of maize stalk. The initial idea was that a capacitance map constructed of multiple measurements between electrodes would provide detailed information about any ECB damage in a stalk (Figure 2-1).

Figure 2-1: Initial measurement concept. Multiple electrodes placed on both sides of the stalk allow electric fields (yellow) to interrogate different parts of the stalk which could then be combined to create a capacitance map of the stalk.

A variety of electrode arrays were used for initial prototypes as shown in Figure 4 and Figure 5, but all prototypes ran into a few common challenges. First, the data processing was far more complex than initially anticipated. Each electrode array produced many combinations of capacitance measurements (as well as conductance measurements), with each measurement containing several frequency measurements. Meaningful data interpretation was simply overwhelming. This problem was compounded by initial prototypes’ high susceptibility to noise. The electrodes had to be manually placed and held against the stalk. Leaves, ears, and other stalk features were difficult to navigate with the electrode array and any movements during measurements would introduce noise. Additionally, capacitance measurements were influenced by the presence of a human hand in close proximity to electrodes.
Looking back, these initial arrays and measurements were extremely ambitious, but they yielded important information about the exquisite sensitivity that would be needed for these measurements, as well as the necessity of producing reliable mechanical placement of the electrodes relative to the stalk to be able to achieve consistent measurements. Stray fields and proper shielding would be necessary to reduce interference that would be associated with any measurement taken in the field. It may be noted that going back to the multichannel measurements may be advantageous, given the knowledge that was gained during subsequent development.

Figure 2-2: A measurement being taken by one of the initial multichannel prototypes.
2.3 **Greenhouse Prototype and Testing**

The issues encountered with the initial prototype led to important revisions as part of the design process. Simplification to obtain reliable electromagnetic impedance data in controlled, repeatable mechanical arrangements became the new goal. The electrode configuration was
simplified to use a single pair of electrodes to take a single, frequency-swept impedance measurement (Figure 2-4). This change greatly simplified data analysis to focus on larger-scale impedance variations that were believed to exist in the stalk. Another improvement was implementing a guarded electrode configuration. The guarded electrode setup helped to confine the probing electric fields to a known volume and mitigated the negative impact of interference by both users and the plant material. Last, the electrodes were attached to a mechanical clamp mechanism. This clamp allowed the electrodes to make good contact with the stalk, stay steady during a measurement, and remove the presence of a human hand during measurements.

Figure 2-4: Single pair of electrodes which shows the guard ring on one electrode to reduce the influence of stray fields on the measurement.

Figure 2-5 shows the experimental setup used for the greenhouse testing. The guarded electrodes were glued to a pair of clamps. The electrodes were connected to the AD2 with a shielded cable. Data was collected from the AD2 and saved on the data collection computer. Due to travel restrictions imposed in response to the COVID-19 pandemic, this setup was shipped to our sponsor. This introduced an additional challenge in that the entire data collection
process needed to be streamlined and convenient for a user without significant training. Sponsor personnel used this simplified test setup to measure the impedance on greenhouse-grown maize plants.

For the greenhouse testing, greenhouse grown plants (Corteva PHR03) were artificially infested with ECB neonates to simulate a 2nd generation ECB infestation. Of the 108 plants grown, a total of 59 plants were electrically measured during this test. Impedance measurements on plants were taken at 10 internode locations, with locations being located 40mm above or

Figure 2-5: Prototype setup used in greenhouse testing (first milestone).
below the first five lower nodes of the plant (nodes are where leaves and ears connect to the plant and are characterized by a distinguishable bump; internodal stalk sections are located between nodes). Each measurement consisted of conductance and capacitance values for 41 frequencies that were logarithmically spaced between 1kHz and 100kHz. A total of 590 stalk locations were measured. Subsequent destructive analysis revealed that 84 of these 590 locations had ECB damage. Because of the nature of the testing arrangement, it truly was a blind test in that no modifications to the hardware or data collection software could be made as the test was executed by personnel who were not involved in the creation of the test apparatus.

Figure 2-6 shows the comparison of mean values for conductance and capacitance between intact locations and damaged locations at each of the interrogated frequencies. Also shown in the plot is the 95% confidence interval. As seen, the results of the greenhouse testing data do not show any statistically significant difference between intact and damaged stalk sections; however, an important trend is apparent in this data: intact sections have a higher capacitance on average than stalk sections damaged by ECB larvae. These findings hinted that the original hypothesis, which stated that ECB damage leads to a drop in measurable capacitance, is correct, but ultimately did not validate the hypothesis.

Because of the consistency of the measurements across the entire frequency range, this initial data also hinted that a parametrization of the impedance data might be possible to extract better estimates of permittivity and conductivity parameters that could be estimated with higher accuracy and reduce the noise associated with the raw measurements. While this type of parameterization was not pursued extensively at this stage because of the desire to improve the measurement capability of the device itself, some fruitful ideas for processing the data were also initially explored at this stage.
Figure 2-6: Greenhouse results comparing conductance (top) and capacitance (bottom) between intact and damaged measurement locations. The difference of means at each frequency index is shown along with the 95% confidence interval.

Nonetheless, these findings provided valuable feedback about how to improve the device. First, the high variance in the results indicated that measurement noise needed to be reduced if actual differences between intact and damaged stalk sections were to be seen. High, variable noise components like the long, shielded cables would need to be eliminated from the design. Additionally, the clamping mechanism that interfaced the electrodes to the stalk needed to be replaced with a mechanism that applied consistent, repeatable contact pressure. Stalk placement and orientation within the apparatus needed to be more tightly controlled. These changes culminated in the design and creation of the subsequent, improved device.
2.4 Initial Maize Stalk Permittivity Estimation

The issues discovered with the initial prototype made clear that numerical simulations would support exploration of the design space parameters that would influence capacitance measurements. Unfortunately, due to time constraints imposed by the greenhouse tests, the bulk of these simulations were not conducted until after the first test milestone. However, the results from the greenhouse tests reiterated the importance of these numerical simulations.

Since permittivity and capacitance are linked, an understanding of maize stalk permittivity was needed to accurately simulate changes in capacitance due to ECB damage. To this end, the permittivity of maize pith was estimated in the laboratory. For this experiment, pith samples were dried out and then rehydrated before taking permittivity measurements to control water content. The pith samples were obtained from one maize stalk that had been harvested from the field at the end of a growing season.

Three 1cm wide slices were cut from the central internode sections of the maize stalk. These three slices were set out to at room temperature for 96 hours before being cut into 10x10x10mm cubes (Figure 2-7). The cubes were then soaked in tap water for 72 hours. These cubes were weighed before and after soaking and removal and surface drying with a paper towel.

Capacitance of these pith samples was measured with an Agilent 4294A from 1kHz to 1MHz. Full details on this measurement setup are given in Chapter 3. Measurements were taken along two different orientations: electrodes parallel to grain and electrodes perpendicular to
grain. An open-air measurement was taken after each orientation measurement, maintaining the same electrode spacing, to estimate parasitic capacitance.

Figure 2-7: Dry pith cubes used to estimate stalk permittivity.

To estimate the relative permittivity, the measured capacitance, $C_{\text{measured}}$, was assumed to have three capacitance contributions: the capacitance of the pith cube, the capacitance of air for any volume between the GE and CE not occupied by the pith cube ($C_{\text{air}}$), and parasitic capacitance ($C_{\text{parasitic}}$). The $C_{\text{parasitic}}$ was estimated by subtracting the theoretical open-air capacitance from the measured open-air capacitance at each cubes’ height. The relative permittivity was estimated with the following equation:

$$\varepsilon_r = \left( C_{\text{measured}} - C_{\text{parasitic}} - C_{\text{air}} \right) \frac{H}{\varepsilon_0 A}$$

(1)

where $H$ is the height of the pith cube, $A$ is the surface area of the pith cube in contact with the electrode, and $\varepsilon_0$ is vacuum permittivity.

Measurements indicate that the parallel orientation had a lower average permittivity than the vertical orientation. At 10.2kHz, the average horizontal permittivity was 15.8, the average
vertical permittivity was 17.0, and the overall average permittivity was 16.4. Ulaby and Jedlicka found the permittivity of maize stalks to be between the values of 2 to 40 between 1.5GHz to 8.0GHz with permittivity increasing with higher water content [22]. Given the possible range of permittivity that stalks might exhibit (and structure-based capacitance changes would be exaggerated at higher permittivity values), these experiments motivated an estimate of the pith permittivity to be about 20 for numerical simulations of capacitance changes. The results of these simulations are detailed in Chapter 3.

This chapter presented many of the initial designs that were created at the beginning of this project. Based on the results of experimentation with those designs, a simplified design was created and was used in Iowa to take initial greenhouse measurements of maize stalks with ECB damage. While not statistically significant at individual frequency points, those results hinted that a change in capacitance might be expected due to ECB damage and that the underlying hypothesis might be correct. Finally, some initial permittivity estimates were made to guide further analytical and numerical simulation work. These steps laid the foundation for creation of an improved device that was used in the field experiments that are documented in the next chapter.
CHAPTER 3. CAPACITANCE MEASUREMENT DEVICE TO ASSESS ECB DAMAGE IN MAIZE

This chapter will be published in a peer-reviewed paper that was submitted for review before the thesis was defended. As such, this chapter contains small differences between the to-be-published work and what is presented here.

3.1 Introduction

Maize, *Zea mays*, is an important global cereal crop used in food products, animal feed, and fuel production. In 2019, about 197 million hectares of maize was grown globally, of which, about 52 million hectares were grown in Europe and North America [1]. One challenge to growing maize in Europe and North America is the European Corn Borer, *Ostrinia nubilalis* (ECB). Native to Europe, the ECB was introduced to North America in the early twentieth century and has since spread Midwest [2, 3]. In the United States alone, the ECB is estimated to cause $1-2 billion dollars of loss to maize crops [4, 5].

A few strategies have been employed to mitigate the impact of the ECB. Traditional methods of pest management have utilized strategic crop planting and harvesting schedules [7, 8], insecticide application [7, 8], and the use of pest-resistant varieties such as Bt maize [5, 8-11]. One challenge to all management strategies is tracking ECB infestation throughout the growing season [5]. Current methods rely on estimating infestation based on ECB larva and moth sample counts [12, 13]. Furthermore, evaluating the effectiveness of pest-resistant varieties has required
quantification of ECB damage [10, 14, 15]. Since ECB damage primarily occurs inside of corn stalks and is rarely visible from the outside, current evaluation methods are destructive in nature and require significant manual labor [14, 15]. An objective, rapid, nondestructive evaluation method for detecting and quantifying ECB damage would be a valuable new tool that would aid in the development of maize varieties that are resistant to ECB infestation.

Little work has been done in developing nondestructive methods for evaluating ECB damage. One study used X-ray imaging to map ECB cavities within maize plants, but cost and field deployment concerns are limiting factors for this method [18]. Another study looked at detecting larvae in the roots of sugarcane [19]. Electrical impedance measurements have been used to nondestructively estimate the root mass size of maize plants [20, 21]. These studies exploited known physiological changes and dimensions in plants to develop nondestructive test methods. Since ECB damage induces physical changes to maize plant structure, there is a possibility that electrical impedance measurements can be used to evaluate ECB damage without damaging the plants.

The purpose of this study was to assess the feasibility of using electrical impedance measurements via external probes to quantify ECB damage. First, numerical simulations and laboratory experiments were performed to explore theoretical impedance changes that would result from ECB damage. The findings from these simulations and experiments were used to aid the development of a measurement device to take impedance measurements on maize stalks. The study culminated in a field experiment that deployed this measurement device to collect impedance data on maize stalks with ECB damage.
3.2 Background

The relative permittivity of a material reflects its ability to polarize when an electric field is applied. Essentially, polarization is charge displacement due to the applied electric field and different materials will polarize to varying degrees. The ratio of electric charge accumulation due to an applied electric potential between two or more conductors is termed capacitance and is dependent on the relative permittivity of the material between the conductors. Since capacitance depends mainly on physical parameters, such as conductor separation and surface area, as well as the electrical permittivity of the medium between the conductors, capacitance measurements can be used to quantify material changes that may be intrinsic in a homogeneous medium as well as rearrangements of heterogeneous material. These principles are exploited in several common nondestructive measurement tools such as stud finders. In each of these cases, changes in physical features and permittivity result in a measurable change in capacitance.

Figure 3-1: Cross-section view of a maize stalk. ECB damage is characterized by replacement of water-rich pith material with air and frass, altering the permittivity and internal material arrangement. Capacitance measured across the stalk decreases.

ECB damage is characterized by a change in stalk internal material structure due to larval tunneling. Water-rich, pith material is replaced with air and frass (larval excrement). Since water has a much higher relative permittivity than air, the working hypothesis is that this damage
would result in a decrease in material permittivity and consequently lead to a measurable
decrease in capacitance between probe conductors placed diametrically across an ECB damaged
stalk portion as compared to an intact portion (Figure 3-1).

3.3 Laboratory Experiments and Numerical Simulations

To determine whether ECB damage would lead to significant changes in capacitance, we
conducted a series of lab experiments and computer simulations to explore capacitance changes
for models comparable to maize stalks. We first present laboratory and simulation results
exploring capacitance change in polytetrafluoroethylene (PTFE) rods with varying degrees of
hollowness. We then measured the permittivity of maize pith and used those results to simulate
potential sources of capacitance variation for maize stalks.

3.4 Physical 3D Cylinder Model – Hollow PTFE Rod Measurements

To explore the relationship between ECB damage and capacitance measurements we first
modeled the maize stalk as a 3D cylinder of homogenous material. As a first-order
approximation, ECB damage was modeled to be a concentric hollow section within the cylinder.
We chose to test this model using PTFE rods. PTFE is a stable, non-hygroscopic polymer with
well-documented electromagnetic properties.

Six PTFE rods were procured, each with a diameter of 12mm and a length of 25mm.
Holes were drilled through the length of five rods to create central hollow sections. One rod was
left intact (Figure 3-2). A range of hole diameters was used: 1.6mm, 3.2mm, 4.8mm, 6.4mm,
and 8.0mm. Each rod’s capacitance was measured at least ten times with an Agilent 4294A
Impedance Analyzer at 5.03kHz using a parallel plate, guarded electrode stage (Figure 3-2) [23].
The guarded electrode setup consisted of three electrodes: the guarded electrode (GE), the guard ring (GR), and the counter electrode (CE) [24]. Placed on the same printed circuit board (PCB), the GE was electrically connected to the low potential (LPOT) and low current (LCUR) terminals and the GR connected to the guard terminal. The CE was on a second PCB and was electrically connected to the high potential (HPOT) and high current (HCUR) terminals. The stage consisted of these two PCBs mounted to two vertically opposed, 3D printed platforms with the CE placed on the top platform and the GE and GR placed on the bottom platform. The top platform was elevated with a screw actuator. For a measurement, the sample rod was placed on the bottom platform centered on the GE. The top platform was then lowered to contact the rod and measurements were taken.

To account for the parasitic system capacitance of the measurement system, an open-air measurement was taken at the height of the rods. Parasitic capacitance is assumed to be additive to the true capacitance of the PTFE rods and was estimated by subtracting the theoretical open-air capacitance of a parallel plate guarded electrode system from the open-air capacitance measurement. Measurements were corrected by subtracting the estimated parasitic capacitance from the measured capacitance for each of the six rods [25].

In addition to the lab measurements on the PTFE rods, the capacitance model was numerically simulated (Ansys® Electronics Desktop 2020 R1 using the Maxwell 3D Design module) to solve for electrostatic capacitance. The simulation model also assumed a PTFE rod with a diameter of 12 mm and a relative permittivity of 2.10 [26]. The hollow ECB damage was modeled as an air cylinder concentric to the rod and was assigned a relative permittivity of 1.00. A guarded electrode setup with the same GE, GR, and CE dimensions as the lab stage were used in this simulation (Figure 3-2). The capacitance of the rod was simulated for the same hole.
diameters as the physical experiment (hole diameters of 1.6mm, 3.2mm, 4.6mm, 7.9mm, 10mm, and 11.9mm).

Figure 3-2: Laboratory and simulation setup for PTFE capacitance measurements. (a) CAD model showing the locations of the guarded electrode (GE), guard ring (GR), and counter electrode (CE) in relation to the rod. (b) The measurement stage with cutouts showing the dimensions of the GE, GR and CE.
The results from the lab test and the simulation are compared against each other in Figure 3-3. As seen, the lab experiment and simulation results exhibit decreasing capacitance as hole size increases in relation to stalk size, with the average measured capacitance being lower than the simulated capacitance. For a hole diameter of 3.18mm, it appears that the off-center cut may have influenced the measured value. The measured open-air measurement is 50.8% of the intact rod capacitance for measured values and 53.1% for simulated values. One key threshold is that measured values and simulated values fall to 75.9% and 77.3% of the intact value respectively at a hole-stalk diameter ratio of 0.66. This suggests that the greatest differences in capacitances between intact and ECB damage stalk sections will be when there is significant internal hollowing due ECB damage. Furthermore, capacitance differences may only be measurable under conditions of extensive ECB damage.

Figure 3-3: Simulated and measured capacitance values at 5.02 kHz for PTFE rods with increasing hole size, represented as a ratio to stalk diameter. Capacitance values are presented as a ratio to intact rod capacitance. Error bars for measured values are 95% confidence intervals.
3.5 Maize Stalk Permittivity Measurements

Since permittivity and capacitance are linked, a reasonable estimate of the relative permittivity of maize stalk materials was needed before additional numerical simulations using a maize stalk capacitance model could be performed. A study conducted by Ulaby and Jedlicka found the permittivity of maize stalks to be between the values of about 2 to 40 between 1.5GHz to 8.0GHz with permittivity increasing with higher water content [22]. Similar results were produced by another study that measured the permittivity of internodal stalk maize sections to be between about 10 to 40 at 1.25GHz [27].

To confirm whether these permittivity values were similar at lower frequencies, we measured the relative permittivity of three moist stalk pith samples. We found the average relative permittivity to be 16.4 at 10.2kHz, which is consistent with the findings of the other studies. Based on these studies and experiments, we decided to use a conservative value of 20 for modeling relative permittivity in our simulations while being conscious of the fact that real-world deviations in stalk permittivity would, to first order, scale the capacitance trends found in our simulations.

3.6 Parameter Sensitivity Analysis by Numerical Simulation of Hollow Maize Stalks

Measurement sensitivity to physical maize parameters is key to a successful measurement. Maize stalks will have varying diameters between plants as well as within a single plant. ECB damage will also vary in shape and diameter. Additionally, maize possess irregular geometry in which the stalk and attached leaves depart from an ideal cylindrical shape. Irregular shapes present a challenge in electrode placement, leading to offsets and contact gaps between the stalk and electrodes. Capacitance sensitivity to critical parameters of maize stalk diameter,
ECB damage diameter, and electrode placement relative to the ECB damage need to be quantitatively explored. To evaluate these factors, maize stalk capacitance was simulated for the following four parameters: stalk diameter (Stalk), ECB damage diameter (Hole), electrode offset (Offset), and contact gap (Gap) (Figure 3-5).

![Simulated maize stalk model. The dimensions for the guard ring (GR), guarded electrode (GE), counter electrode (CE), and the guarded back.](image)

In these simulations, the maize stalk was modeled as a cylinder composed of homogenous material (Figure 3-4). The cylinder was 50mm in length and 20mm in diameter, except for simulations exploring the effects of varying stalk diameter. The maize stalk material was assumed to have a relative permittivity of 20 regardless of stalk size or ECB damage diameter (larger permittivity assumptions increase simulated effects). ECB damage was modeled as a cylinder of air concentric to the maize stalk with a diameter that was bound by the stalk diameter. ECB hole size (Hole) was increased in 0.5mm increments from 0mm to the size of the
maize stalk (*Stalk*). Guarded electrodes were used in the simulation as with the PTFE simulations; however, a few changes were made to reflect a smaller, field deployable design. A second guard area, termed the guarded back, was placed behind the GE and GR to minimize fringing electrical fields. The space between the guarded back and the guarded back was 1.6mm wide and was filled with a dielectric material ($\varepsilon_r = 4.5$) representative of a FR-4 PCB. The dimensions of the GE, GR, and CE were reduced to ensure that the electrodes would approximate the diameter of a typical maize stalk (Figure 3-4).

![Cross section of maize stalk with parallel electrodes](image)

**Figure 3-5:** Cross section of maize stalk with parallel electrodes. Simulated parameters are: (a) ECB size, (b) electrode offset, and (c) contact gap.

Three sets of simulations are conducted. In each set the ECB damage diameter (*Hole*) varies from 0mm (intact) to the diameter of the stalk; capacitance results are displayed as a ratio of *Hole* to *Stalk* ($Hole / Stalk$), where a $Hole / Stalk$ value of 1.0 represents an intact stalk and a $Hole / Stalk$ value of 0.0 is a completely damaged stalk.
The first set of simulations compares stalk diameters of 12mm, 16mm, 20mm, and 24mm. Results are shown in Figure 3-6a. The greatest difference in simulated capacitance between Stalk values 12mm and 24mm is 64.1fF, which is a 11.6% drop in capacitance at 12mm, and occurred when the stalk is intact. The capacitance difference between stalk sizes decreases as Hole increases until the difference effectively disappears when Hole / Stalk is 0.6. This suggests that the stalk diameter Stalk will have minimal impact on capacitance changes if there is extensive ECB damage.

The second set of simulations compared capacitance for Offset values of 0mm, 1mm, 3mm, and 5mm. Results are shown in Figure 3-6b. As seen, as the GE was offset further from the stalk center, measured capacitance decreased significantly. For intact stalks, capacitance differences compared to no electrode offset were 0.019pF at 1mm, 0.192pF at 3mm, and 0.405pF at 5mm, or 3.8%, 37.9%, and 80.0% drops in capacitance, respectively (Figure 3-6b). Furthermore, the capacitance difference between an intact stalk and a completely damaged stalk (Hole / Stalk of 1.0) when the Offset is 0mm is 0.323pF, or a 63.8% drop from the intact capacitance value. This suggests that electrode offset errors could cause estimations of damage from capacitance to overestimate the true damage. Accurate ECB damage detection and quantization based on capacitance requires electrodes to be centered accurately and repeatedly on the stalk.
Figure 3-6: Simulated capacitance values for maize stalks with ECB damage from three simulation sets. (a) Capacitance for maize stalks with diameters ranging from 12mm to 24mm electrode offset from stalk. (b) Capacitance for increasing electrode offset from the maize stalk. (c) Capacitance for increasing electrode contact gap from stalk.
The third simulation set compared contact gaps between the guarded electrode the maize stalk at *Gap* values of 0mm, 1mm, 2mm, and 3mm. Only the GE, GR, and guarded back were moved relative to the stalk; the CE remained in contact with the stalk. The gap between the electrode and stalk was assumed to be air with a relative permittivity of 1. Results are shown in Figure 3-6c. At *Hole / Stalk* of 1.0, capacitance differences compared to no contact gap were 0.341pF at 1mm, 0.402pF at 2mm, and 0.423pF at 3mm, or 67.4%, 79.4%, and 83.6% drops in capacitance, respectively. At 0mm, the difference between capacitance at *Hole / Stalk* of 1.0 and capacitance at *Hole / Stalk* of 0.8 and is 0.323pF, or 63.8% drop from the intact capacitance (Figure 3-6c). These results show that air gaps between the electrode and the stalk will drastically reduce measured capacitance. This suggests that electrodes need to maintain contact with the maize stalk for maximum sensitivity to ECB damage. Furthermore, features that may introduce effective air gaps, such as leaves and stalk sheathing, may pose problems when taking measurements.

These simulations provide sensitivity insight to variations of stalk diameter, ECB damage, and electrode placement. Variation in stalk diameter had small impact on capacitance change. Electrode offset and separation from the stalk appeared to significantly impact capacitance measurements. To detect ECB damage, measurement apparatus designs should avoid poor electrode alignment and tightly control contact with the stalk to produce reliable stalk measurements.
3.7 Prototype Measurement Device

3.7.1 Measurement Apparatus

An apparatus to measure impedance on maize plants was designed and constructed to test whether ECB damage could be detected and measured in maize plants. The measurement apparatus we constructed consisted of these main parts: Handheld gripper with guarded electrodes, interface circuitry for measurement, and an attached computer for control and logging. Figure 3-7 shows the finished device that would be attached to a computer via a USB cable.

![Diagram of measurement apparatus](image)

**Figure 3-7: Stalk measurement apparatus mounted on a 25.4mm diameter wood dowel.** The trigger actuates the spring-loaded plunger to open a space between the electrode guides. When the trigger is released, the electrode guides close and clamp onto the dowel and center the electrodes in the process. The electrodes are pressed against the dowel with small springs to ensure good contact with the dowel.
3.7.2 Analog Discovery & Interface Circuitry

An Analog Discovery 2 (AD2) (Digilent) was configured to measure 51 capacitance and conductance values logarithmically spaced from 500Hz to 100kHz using a 1V peak-to-peak sinusoidal signal. The measurement signal was applied to a 1 MΩ reference resistor in series with the material under test (MUT). The AD2 scope channels 1 and 2 (SC1 and SC2) were connected before and after, respectively, the reference resistor. An op-amp in a voltage follower configuration was used to drive the GR and the op-amp input was connected to the node after the reference resistor and before the MUT. An LF347N (Texas Instruments) powered at ±3.3V was used for the op-amp. Although the GE and GR both interface with the MUT, the GE and GR are not electrically connected to each other. A laptop with a custom Python script (Digilent’s® WaveForms™ SDK) was used to control the AD2, trigger measurements, and save data. Figure 3-8 shows the circuit diagram of the measurement system.

![Circuit diagram of measurement apparatus.](image)

**Figure 3-8: Circuit diagram of measurement apparatus.**
3.7.3 Electrodes

A guarded electrode arrangement was designed with the same dimensions used in the numerical simulations (Figure 3-9). One PCB contained the GE and GR, and the other PCB contained the CE. These electrodes were fabricated on two-layer, FR-4 PCBs using copper pads with a conformal coating. GE diameter was 7mm. A 0.5mm gap separated the GE from the GR. The GR was circular with a width of 5mm. A guarded back with a diameter of 18mm was centered behind the GE and GR. The CE had a diameter of 18mm and was placed on both sides of the PCB. The CE was connected to ground. Both electrode PCBs were 50mm by 20mm in size. A 50Ω coaxial shielded cable connected the GE and GR to the interface board, with the shield layer being connected to the GR. A 26 AWG stranded wire connected the CE to the interface board.

![Figure 3-9: Stalk measurement PCBs. Top PCB with CE. Bottom PCB with GE and GR.](image)
3.7.4 Physical Apparatus

Figure 3-10 shows the handheld apparatus that positions the electrodes next to the maize stalk. The electrodes are connected to 3D-printed, PLA holders with a cyanoacrylate glue (a). These holders share a sliding connection along a guide rail system. The simulation results in section Chapter 3.6 showed that gaps between the electrodes and stalk lead to measurement sensitivity degradation. Because of this, a spring is placed between each PCB and holder to ensure firm contact between electrodes and the stalk by applying pressure to clamp the electrodes to the stalk. As also seen in section Chapter 3.6, electrode offset from the stalk negatively impacts measurement sensitivity to ECB damage. To facilitate proper placement, wedge guides are placed next to the electrode PCBs to center the stalk between the electrodes. The guides also firmly clamp the entire handheld device onto the stalk. The guide for the CE has two wedge grips and the guide for the GE has four wedge grips; the wedge grips alternate between the two guides to ensure an even grip on the maize stalk (b). Compliant hot glue was applied to the edge of the wedge grip to improve friction between the PLA and the stalk.

The CE holder was also connected to a spring-loaded plunger. The plunger is actuated by a trigger to open and close the holders around a maize stalk. The plunger spring is contained within a 3D printed, PLA body that is connected to a handle. On top of this body is a platform holding the AD2 and interface PCB.
Figure 3-10: Clamping mechanism. (a) Electrode PCBs connected to holders with springs. (b) Electrode guides with glue on edges. (c) Bottom view of apparatus showing the spring-loaded plunger.
3.7.5 Device Calibration & Measurement Model

The apparatus was calibrated to estimate impedance parameters. To characterize our
device measurement performance, a series of measurements were performed on known electrical
loads and a measurement model was developed to extract estimates of the true electrical load
values. This model was necessary because the Analog Discovery 2 was used as an inexpensive
impedance analyzer.

A no-load measurement was used to characterize parasitic and system noise. The resistor
\( R_{ref} \) was disconnected from the GE and GR driver by removing the jumper at Node A; cables
connecting the interface board to the electrodes were disconnected (Figure 3-8). 15
measurement sets were recorded. Capacitance and conductance values were averaged from the
15 measurements at each of the 51 measurement frequencies from 100 Hz to 100 kHz. These
averaged measurements became our baseline noise impedance, \( N_{bl} \).

Known-load device performance was determined by measuring 9 ceramic capacitors and
10 1/4W resistors. For these measurements, the jumper at Node A was replaced and the shielded
cable for the GE and GR was reconnected to the interface board. This shielded cable, along with
a ground wire, were connected from the interface board to a breadboard. Loads were placed on
the breadboard to connect one terminal to ground and the other terminal to the GE. The GR was
left electrically unconnected to the load. Each load was measured 5 times. To determine ground
truth values of the resistors and capacitors, each resistor and capacitor was measured with an
Agilent 4294A at 5kHz 15 times, and an average value was calculated. Average measured
values for the capacitors with the Agilent 4294A at 5kHz were 1.08pF, 1.61pF, 3.59pF, 7.96pF,
10.05pF, 20.62pF, 29.58pF, 47.74pF, and 59.36pF. Values for the resistors were 999.8Ω,
21.6kΩ, 46.9kΩ, 99.5kΩ, 223.4kΩ, 565.5kΩ, 831.6kΩ, 998.4kΩ, 10.7MΩ, and 18.3MΩ.
Figure 3-11: Comparison of $N_{bl}(\omega)$ to $Y_{\text{measured}}(\omega)$ for various capacitor (left) and resistor (right) loads. The resistors values from 223kΩ to 18.2MΩ are shown, and all capacitor values from 1pF to 59.3pF are shown.

The impedance $N_{bl}$ was compared to measurements of these known loads. Figure 3-11 shows $N_{bl}$ compared to these measurements on plots of frequency, conductance, and capacitance. These plots show that $N_{bl}$ is not insignificant for these measurements. Furthermore, $N_{bl}$ appeared to be additive. To extract out the true measured values across all frequencies we developed a model that would allow us to remove $N_{bl}$ from measurements and correctly estimate the true load values.

A visual representation of this measurement model is shown in Figure 3-12. This model assumes that the baseline $N_{bl}$ is linearly added to the load. The load is represented as an inductor ($L_s$) in series with a parallel resistor ($G_p$) and capacitor ($C_p$). $C_p$ is the main parameter of interest—the parameter that we expect to correspond to ECB damage. $L_s$ and $G_p$ are expected to contain additional parasitic noise beyond $N_{bl}$ and may contain additional measurement information about the maize stalk. In this model, the measured capacitance and conductance values across all frequencies were converted into an admittance value $Y$ such that

$$Y(\omega) = G(\omega) + j\omega C(\omega) + N_{bl}(\omega) \quad (2)$$
where $G(\omega)$ and $C(\omega)$ are the measured conductance and capacitance respectively at frequency $\omega = 2\pi f$. $N_{bl}(\omega)$ is the baseline noise admittance across all frequencies. For simplification and noise reduction, $N_{bl}(\omega)$ was estimated using a second order parametric estimate $N_{est}(\omega)$ such that

$$N_{est}(\omega) = \omega (G_{n1} - jL_{n1}) + \omega^2 (G_{n2} - jL_{n2})$$

(3)

where $G_{n1}$, $G_{n2}$, $L_{n1}$, and $L_{n2}$ fitting parameters. We chose this estimate because we noticed that the baseline noise was inherently inductive. $N_{est}(\omega)$ was fitted to $N_{bl}(\omega)$ noise minimizing the error function $E_{bl}(\omega)$:

$$E_{bl}(\omega) = |log_{10}N_{bl}(\omega) - log_{10}N_{est}(\omega)|^2$$

(4)

where log_{10} is a complex logarithm defined as

$log_{10}A = log_{10}|A| + j\theta_A, \theta_A = atan2(I\{A\}, R\{A\})$.

The complex logarithm was used to aid the computer in minimizing the error function because the order of magnitude of the raw capacitance and conductance values often dropped below $10^{-7}$. $G_{n1}$, $G_{n2}$, $L_{n1}$, and $L_{n2}$ were given initial values of $10^{-11}$, $10^{-11}$, $10^{-11}$, and $10^{-11}$ respectively. Minimization was conducted using Scipy’s fmin function (Nelder-Mead simplex algorithm).

Having estimated $N_{est}(\omega)$, values for the parameters $L_s$, $G_p$, and $C_p$ were then estimated using the following model:

$$Y_{est}(\omega) = N_{est}(\omega) + \left( j\omega L_s + \frac{1}{G_p + j\omega C_p} \right)^{-1}$$

(5)

$Y_{est}(\omega)$ was fitted to measured admittance data $Y_{measured}(\omega)$ by minimizing the error function $E_Y(\omega)$:
Minimizing this error function produces estimates for $L_s$, $G_p$, and $C_p$. We used the initial values of 1pH, 10pS, and 1pF for $L_s$, $G_p$, and $C_p$ respectively.

This model was used to estimate $G_p$ and $C_p$ values for known loads and these estimates were compared to measured values. This comparison is shown in Figure 3-13. The measurement procedure accurately estimates value of R ($R = 1/G$) from 999.7 to 998kΩ. Measurement estimates begin to diverge from true values for higher resistor values. For capacitors, the model accurately estimates values from 20.6pF to 59.3pF. The model estimates diverge from true capacitor values below 20.6pF, with large divergence below 3.6pF. This may also be due to some trace capacitance found in the system. Fortunately, departure trends for both resistors and capacitors appear to be monatomic in nature. Because of this result, the measurement procedure is predicted to be sufficiently sensitive, but estimates of capacitance changes due to ECB damage if the true maize stalk values are below 20.6pF may be slightly compressed. As simulation
results placed capacitance changes due to ECB in this low range, sensitivity to ECB damage may be slightly reduced in this measurement apparatus.

Figure 3-13: Model estimated conductance (left) and capacitance (right) compared to known values of resistors and capacitors, respectively. The error bars show the 95% confidence intervals for all measurements (most are so small they are not visible).

3.8 Field Methods

A field experiment was conducted to observe the correlation between ECB damage and measured impedance using the new apparatus. This experiment consisted of two parts: electrical measurements on intact plants and subsequent destructive measurements to obtain ground truth data. Due to COVID travel restrictions, the entirety of this experiment was carried out by sponsor personnel in Iowa, USA: planting and caring for the experimental crop, ECB infestation, data gathering using the new measurement apparatus, and performing destructive tests on the maize stalks.
The maize plants for this experiment were planted at two different sites in Iowa, USA. One site was planted in Johnston on April 30, 2020. The second site was planted near Reasnor on May 1, 2020. Two Pioneer® Brand commercial hybrids of similar genetic background were selected for planting: an ECB-resistant hybrid (Bt, P1151AMX), and an ECB susceptible hybrid (nBt, P1151R). The experiment utilized a randomized complete block design with 3 replications per site. Treatments included ECB larvae infested at 3 rates to each hybrid: 300 neonates per plant, 150 neonates per plant, and no infestation. Maize plots consisted of one row, each 3.96 m in length, and contained twenty-two plants. Nine plots of each variety were planted at each site for 18 plots of maize at each location, or a total of 36 plots for the entire experiment. Both sites were managed with standard agronomic practices typical for commercial maize production in the region.

The maize plants at each site were artificially infested with ECB larvae to mimic second generation ECB infestation. The source of ECB was from a colony maintained at the Corteva Agriscience Insectary in Johnston, IA. Artificial ECB infestation at a site began at anthesis when approximately 50% of plants in the experiment were shedding pollen. Each plot was infested with ECB neonates 3 times within a 7 to 10 day period. Within each plot, the first 3 plants were skipped and then the next consecutive 7 plants were infested. At Johnston, the first infestation occurred on July 13, 2020, but the site was lost to a severe windstorm on July 14 that caused extensive plant lodging. Reasnor was successfully infested on July 13, 15, and 17, 2020. Each plant variety received three different treatments: 100 neonates per infestation application for 3 plots per site, 50 neonates per infestation application for 3 plots per site, and no neonate application for 3 plots per site.
For the 100-neonate treatment, the first application of neonates applied 50 neonates to the leaf collar of the primary ear and 50 to the leaf collar just above the primary ear. The second application applied 50 neonates to the leaf collar of the primary ear and 50 to the leaf collar just below the primary ear. The third application of neonates applied 50 neonates to the leaf collar of the primary ear and 50 to the leaf collar just above the primary ear. For the 50-neonate treatment, neonates were applied to the leaf collar of the primary ear for all three applications.

The original experiment plan called for electrical measurements to occur 50 days after the last successful infestation date with a destructive scoring evaluation to occur immediately after the plants were electrically measured. On August 10-11, 2020, a severe derecho caused significant damage to plants at Reasnor. As result, the decision was made to collect samples from the field on September 14, 2020 and bring those samples into a controlled indoor space where electrical and destructive testing would take place. 44 plants were ultimately harvested from the fields for testing.

3.8.2 Measurements

Each of the 44 plants brought in from the field was stripped of leaves and cobs during the harvest. Each plant was tagged with a unique plant ID barcode that indicated its treatment type. 10 measurement locations were selected on the internodal regions of each plant. The locations were located 40mm above and below the first 5 nodes starting from the bottom of the plant. These locations were marked and labeled from 1 to 10 from bottom to top. The diameter of each measurement location on the stalk was measured and recorded.

The first round of testing consisted of taking electrical measurements on each plant. The measurement procedure began by scanning the bar code of the plant to record which plant was
being tested. The measurement apparatus was then clamped onto the first measurement location and an electrical measurement was taken. Subsequent measurements were then taken at each of the other locations. When all the locations had been measured one time, this process was repeated two more times for a total of three measurements at each measurement location. The purpose of measuring three times was to mitigate any effects of electrode misplacement. Once all measurements were completed on a plant, the next plant was scanned, and the process was repeated in its entirety.

After all electrical measurements were complete, the destructive testing began. Each plant was sliced in half longitudinally to reveal the inside of the maize stalk. A photograph of each plant was taken after it was split and measurement locations with visible ECB damage were noted. Other conditions, such as plant rot, were also noted.

3.9 Field Results

A primary objective of these measurements is to determine whether a statistically significant difference existed in the capacitance between measurement locations with ECB damage and intact locations. Table 3-1 shows summary statistics for the plants that were collected from the field and measured in the lab. A total of 44 plants were evaluated. 15 plants were of the Bt variety and received the 0 neonate treatment, and 29 plants were the nBt variety and received the 50 and 100 neonate treatments. Destructive testing revealed that of the 440 measurement locations, 89 had ECB damage and 351 were intact. Figure 3-14 shows an example of what the ECB damaged looked like for some of the plants measured.
Table 3-1: Summary statistics of maize plants in field experiment. Details about ECB damage are given.

<table>
<thead>
<tr>
<th>Plants per Treatment Type (Variety)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 neonates (Bt)</td>
<td>15</td>
</tr>
<tr>
<td>50 neonates / infestation (nBt)</td>
<td>15</td>
</tr>
<tr>
<td>100 neonates / infestation (nBt)</td>
<td>14</td>
</tr>
<tr>
<td>Total Plants</td>
<td>44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurements</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement locations per plant</td>
<td>10</td>
</tr>
<tr>
<td>Total locations measured</td>
<td>440</td>
</tr>
<tr>
<td>Measurements per location</td>
<td>3</td>
</tr>
<tr>
<td>Total measurements</td>
<td>1320</td>
</tr>
<tr>
<td>Frequency range of measurement (kHz)</td>
<td>0.5 - 100</td>
</tr>
<tr>
<td>Frequency bins</td>
<td>51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ECB Damage / Intact Locations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Locations with ECB damage (total)</td>
<td>89 / 351</td>
</tr>
<tr>
<td>0 neonate treatment with ECB damage</td>
<td>0 / 150</td>
</tr>
<tr>
<td>50 neonate treatment</td>
<td>45 / 105</td>
</tr>
<tr>
<td>100 neonate treatment</td>
<td>44 / 96</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diameter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Diameter (mm)</td>
<td>19.57</td>
</tr>
<tr>
<td>Standard Deviation (mm)</td>
<td>2.65</td>
</tr>
<tr>
<td>Min – Max (mm)</td>
<td>12.06 – 27.38</td>
</tr>
</tbody>
</table>

Each plant had 10 locations that were measured 3 times for a total of 30 measurements per plant. Each measurement consisted of capacitance and conductance values across 51 logarithmically spaced frequencies from 500Hz to 100kHz. In processing the data, the capacitance and conductance values from the 3 measurements at each location were averaged at each frequency. No adjustments were made for diameter. These average measurements were then separated into two population groups based on whether the measurement site had ECB damage.

The measurement model was used to calculate $L_s$, $G_p$, and $C_p$ values for each measurement location based upon the frequency averaged capacitance and conductance for that location. These $L_s$, $G_p$, and $C_p$ parameter values were then averaged across the two population groups.
groups. Figure 3-15 shows the non-parameterized frequency-averaged data for each measurement location. The location data is segregated based on the presence of ECB damage. The average model fit for each population is also shown. As seen in Figure 3-15, there is a visual difference between the average fit of the two populations. Table 3-2 shows the difference of means between the two populations for each of the parameter values. As seen in Table 3-2, only the $C_p$ parameter was statistically significant between the two populations, indicating that ECB damage produces a statistically significant, measurable change in the capacitance of the maize stalk where the damage is located.

![Figure 3-14](image)

**Figure 3-14:** Photograph of representative set of split maize stalks, with some stalks showing ECB damage (dark areas).

### 3.10 Discussion

Field experiment results, combined with the lab experiments and simulations, paint a complete picture of how ECB damage may impact the capacitance of maize stalks. As seen in Figure 3-15 and Table 3-2, measurement sites with ECB damage, on average, exhibited lower
capacitance than their intact counterparts. This trend was seen in the raw data, as well as in the extracted \( C_p \) values from the measurement model. Furthermore, the difference of means was statistically significant for the average \( C_p \) value between intact stalk locations and ECB damaged locations. These findings confirm our hypothesis that ECB does change the physical structure of maize stalks in ways that are measurable with capacitance. Strikingly, the confidence interval for \( C_p \) agrees in magnitude with the predicted difference from the simulations as seen in Figure 3-6. The predicted difference in capacitance was about 0.3 to 0.5pF depending on the extent of damage and stalk diameter. The estimated measured difference was about 0.546pF, with a confidence range of 0.346pF.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Diff Mean ± Std Error</th>
<th>Statistically Significant at P=0.05 level</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_s )</td>
<td>2.22pH ± 11.6pH</td>
<td>No</td>
</tr>
<tr>
<td>( G_p )</td>
<td>27.2 pS ± 751 pS</td>
<td>No</td>
</tr>
<tr>
<td>( C_p )</td>
<td>0.546 pF ± 0.346pF</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Additionally, the results show that ECB does not significantly change the conductance of the maize stalk as estimated by the model parameter \( G_p \), nor do the results suggest any significant inductive change as given by the parameter \( L_s \). However, it is important to note that the parameter values produced by the measurement model are estimates of a simplified circuit model of the maize stalk.
The fundamental electrical property that we expected to change in response to ECB damage was the relative permittivity of the maize stalk at the site of damage. The statistical change in the average capacitance parameter between intact and damage maize stalks sections strongly suggest a change in permittivity; however, a more detailed study would need to be conducted to explore how ECB damage changes permittivity. Additionally, we acknowledge the limitations of our measurement apparatus. As seen in Figure 3-13, the measurements collected...
with our apparatus have a monatomic nonlinear behavior below 20pF, which is where we would expect the true values of stalk capacitance to be. Although we could still see a difference in capacitance, the true change in capacitance, and thus the true change in permittivity, needs further investigation. Additionally, while the statistical averages clearly show changes in capacitance, because of large variability of fundamental electromagnetic parameters – both intrinsic (perhaps water content and structure) and extrinsic (how electrodes interfaced with the complex geometry of the stalk) – there is also clearly a great deal of variation in the measurements. Visual indication through destructive measurements is quite clear, in comparison.

Unfortunately, due to the 2020 Iowa derecho, our planned field test had to be altered. As a consequence, we did not take electrical measurements in the field. Furthermore, the sample size of the plants was severely reduced, prohibiting the possibility of comparing measurements between and within plant varieties, as well as between and within treatment types. Although these factors do not detract from the findings, they do obscure some potential nuances of plant variety and treatment type that may exist. More testing would need to be done to explore these influences on capacitance and permittivity.

Regardless, the results of this study validate the possibility of using capacitance measurements to nondestructively evaluate ECB damage in maize stalks. Certainly, many additional improvements could be made to the measurement device to increase the resolution and reduce the variability of the measurements. Overall, because ECB damage appears to change the electromagnetic properties of the stalk, capacitance measurements show promise as a field-deployable, nondestructive method of quantifying ECB damage within maize plants.
3.11 Conclusion

This study tested the hypothesis that the capacitance of maize stalks would change due to ECB damage. Laboratory experiments and numerical simulations were performed to study the potential order of magnitude of this effect and indicated that a hollow core would result in reduced capacitance. A new device was constructed to take guarded electrical impedance measurements in the laboratory and field. An experiment was conducted on field-grown stalks. ECB damage within maize stalks was statistically correlated with measurable changes in capacitance. These capacitance changes were apparent in both simulations and in measurements taken on maize plants. The results of this study validate capacitance measurements as a potential nondestructive tool to evaluate and quantify ECB damage.

This study also indicates that nondestructive electrical impedance measurements may be useful in additional agricultural applications which result in structural or material properties changes that result in different electrical impedance signatures. The creation of this device and the data gathering strategy also point to smart measurement and management techniques that could be deployed to quantitatively measure and enhance agricultural productivity.
CHAPTER 4. CONCLUSION AND FUTURE WORK

The work presented in this thesis demonstrated that electrical impedance measurements can be used effectively to evaluate maize stalks for non-visible ECB damage. Specifically, lab tests, numerical simulations, and field testing showed a statistically significant capacitance difference between intact maize stalk sections and damaged sections. Additionally, a portable, rapid measurement impedance analyzer was developed and deployed to demonstrate that impedance measurements can be taken in the field. Importantly, this device was deployed by personnel who were not involved in its creation, demonstrating the transferability and viability of the technology. Remarkably, this research demonstrated the entire product development cycle from a hypothesis about a proposed material change leading to a measurable difference to deployment of a new device harnessing that principle.

This work lays the groundwork for continued technological development for nondestructive electrical testing and evaluation of pests and diseases in the agricultural industry. The following suggestions may be considered in the advancement of this work:

- An in-depth study of the electrical permittivity of maize plant material. This study would be invaluable to understand how water content affects the impedance of maize plants. Additionally, this study would shed light upon frequency ranges with enhanced sensitivity in which to take measurements and greatly improve numerical simulations of stalk responses.
• A field test of the measurement device. Unfortunately, the planned field test for this study was altered due to natural disasters. Such a test would provide valuable feedback in developing the device for field use.

• Improved measurement device circuitry. The measurement circuitry was based upon the Analog Discovery 2 (Digilent) and was rudimentary in nature. Moving to more capable circuitry would likely improve measurement quality and expand the frequency range at which measurements could be taken. Going back to multichannel measurements could also improve measurement speed and reliability as well as noise-rejection abilities of algorithms to estimate damage.

• Improved handheld apparatus. The handheld apparatus used in the measurement device could be improved to negotiate plants with leaves and ears left in place. The speed and ease of data acquisition could also be improved with more human factors engineering.

• Development of quantitative metrics for evaluating ECB damage. This study showed that there are statistically significant capacitive differences between damaged and intact maize stalk. With higher quality measurements and a better understanding of how factors such as water content affect maize stalk capacitance, it would be helpful to have objective metrics with which to estimate and grade ECB damage.
REFERENCES


[16] ed: University of Nebraska--Lincoln Department of Entomology


APPENDIX A. MEASUREMENT STAGE SCHEMATICS

The schematics and assembly diagrams are presented for the guarded electrode stage used for the permittivity estimates in Chapter 2 and the PTFE rod measurements detailed in Chapter 3. The stage uses a guarded electrode configuration and was initially designed to be used with an Agilent 4294A.

A.1 Fabrication and Assembly

This measurement stage was 3D printed using PLA filament. When 3D printed, the CE and the GE-GR PCBs can be screwed directly into the stage’s platforms using M3.5 screws. The test setup had the CE PCB on top, and the GE-GR PCB on the bottom, but these can easily be reversed.

The three ¼”-28 x 2.5” screws can be directly screwed into the bottom stage platform. Rubber feet can be placed on the bottom stage platform to provide grip and elevation to bottom platform. Two ¼” nuts are placed above and below the top platform on the center actuator screw control the top platform’s elevation.

Please note that the schematics have been scaled to fit in this document and that the labeled scales are only valid for the original documents.

The schematics and CAD files for the stage platforms were designed in SpaceClaim (version 2020.1.0.12031). The electrode CAD files were designed in Altium Designer (version 20.1.14).
Figure A-1: Assembly diagram for measurement stage.

Assembly note:
When stages are 3-D printed with PLA, use M3.5 screws to fasten electrode PCBs to stages.
Figure A-2: Diagram showing multiple views of the bottom platform.
Figure A-3: Dimensions for the bottom platform.

<table>
<thead>
<tr>
<th>HOLE</th>
<th>X</th>
<th>Y</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>33.68mm</td>
<td>25mm</td>
<td>Ø6.4mm</td>
</tr>
<tr>
<td>A2</td>
<td>68.32mm</td>
<td>25mm</td>
<td>Ø6.4mm</td>
</tr>
<tr>
<td>A3</td>
<td>51mm</td>
<td>15mm</td>
<td>Ø6.4mm</td>
</tr>
<tr>
<td>B1</td>
<td>38mm</td>
<td>93mm</td>
<td>Ø3.5mm</td>
</tr>
<tr>
<td>B2</td>
<td>64mm</td>
<td>93mm</td>
<td>Ø3.5mm</td>
</tr>
<tr>
<td>B3</td>
<td>38mm</td>
<td>46mm</td>
<td>Ø3.5mm</td>
</tr>
<tr>
<td>B4</td>
<td>64mm</td>
<td>46mm</td>
<td>Ø3.5mm</td>
</tr>
</tbody>
</table>
Figure A-4: Diagram showing multiple views of the top platform.
Figure A-5: Dimension for the top platform.
Figure A-6: Dimensions for the measurement stage CE PCB.
Figure A-7: Manufacturing schematic for the measurement stage CE PCB.
Figure A-8: Dimensions for the measurement stage GE and GR PCB.
Figure A-9: Manufacturing schematic for the measurement stage CE PCB.
APPENDIX B. CODE FOR ANALOG DISCOVERY MEASUREMENTS

This section contains code to run the Analog Discovery 2 and collect impedance measurements. All code listed was written in Python 3.6. Some of the code uses code found in Digilent’s WaveForms SDK and API and is subject to Digilent’s software licensing terms and conditions.

B.1 File Structure

The code presented in this appendix should be organized into the following file structure:

- Main directory
  - GUI
    - MainWindow.py
  - Tools
    - AnalogDiscovery.py
    - BitTools.py
    - dwfconstants.py
  - Data
    - <filled with folders and files with measurement data>
  - MeasurementDriver.py
**B.2 MeasurementDriver.py**

The following code is the primary driver of the measurement program. Running this script will start the data collection software. The software is operated from a graphical user interface (GUI).

```python
import tkinter as tk
import time
from Tools.AnalogDiscovery import AnalogDiscovery
from Tools.dwfconstants import *
from GUI.MainWindow import MainWindow
import os

class GreenHouseApplication:
    def __init__(self):
        """This class contains the main program loop. It holds the objects used to interact with the GUI and Analog Discovery. Note that the GUI object will make calls to this class.

        Setup includes initializing the Analog Discovery and GUI.
        """
        # Version Number
        self.version = 1.2

        # Interface with Analog Discovery
        self.discovery_is_connected = False
        self.discovery = None  # Placeholder for interface object

        # Analog Discovery Settings
        self.ref_resistor_value = 1e6
        self.start_freq = 5e2  # OG value: 1e3
        self.end_freq = 1e5    # OG value: 1e5
        self.steps = 51        # OG value: 41
        self.data_1_id = DwfAnalogImpedanceConductance
        self.data_2_id = DwfAnalogImpedanceParallelCapacitance

        # Create Tkinter main root and main GUI
        self.tk_root = tk.Tk()
        self.tk_root.protocol('WM_DELETE_WINDOW', self.quit_program)
        self.gui_window = MainWindow(self.tk_root, self)

        # Current test directory: used as reference for saving data files.
```

This is the main driver for the measurement code. It is designed to take and save impedance measurements from an Analog Discovery. It supports the GUI, facilitates the interactions with the Analog Discovery, and manages data organization and saving.

It is written in Python 3.6

version 1.2
about this version:
- Changed frequency range to 500Hz to 100kHz
- Increased samples from 41 to 51

Mavrik Thomas
BYU
July 7, 2020
self.corn_dir = ''

# Flags
self.running = True

def run(self):
    ''' Calling this function initiates the main program loop.
    The mainloop consists of updating the GUI. The loop runs as long as the self.running flag is
    true.
    :return:
    '''
    while self.running:
        # Update GUI
        try:
            self.tk_root.update_idletasks()
            self.tk_root.update()
            self.gui_window.update_status_indicators()
        except Exception as error:
            print('Update GUI error: ', error)
            quit()

def quit_program(self):
    ''' Quit the program. Called when main GUI window is closed.
    This will terminate the mainloop in the 'run' function and wait a bit for things to wrap up.
    :return: None
    '''
    self.running = False
    time.sleep(0.2)

def connect_discovery(self):
    ''' Connect to Analog Discovery. This is create the class instance for the Analog Discovery.
    The connection flag is set in this function based on connection outcome.
    :return: nothing
    '''
    if not self.discovery_is_connected:
        print('Connecting to Analog Discovery...')
        self.discovery = AnalogDiscovery()
        self.discovery_is_connected = self.__configure_analog_discovery()
        if self.discovery_is_connected:
            print('Analog Discovery Connected')
        else:
            print('Failed to connect to Analog Discovery')

def disconnect_discovery(self):
    ''' Disconnects from the Analog Discovery. The class instance for the Analog Discovery is
    closed.
    :return: nothing
    '''
    if self.discovery_is_connected:
        print('Disconnecting from Analog Discovery')
        self.discovery.close()
        self.discovery = None
        self.discovery_is_connected = False

def __configure_analog_discovery(self):
    ''' Configures the Analog Discovery after connection.
    The following settings are configured:
    > Power supply voltage
    > Reference resistor value
    > GPIO default output
# Set up power rails

try:
    self.discovery.voltage_set_positive(3.3)
    self.discovery.voltage_set_negative(-3.3)
    self.discovery.set_reference_resistor(self.ref_resistor_value)

# Configure digital IO
enable_mask = 0x80FF
self.discovery.digital_out_reset()
self.discovery.digital_out_configure(enable_mask)
    return True
except Exception as error:
    return False

@staticmethod
def create_data_directory():
    
    
    os.mkdir('Data')

def create_corn_stalk_directory(self, new_stalk_name):
    """ Creates a subdirectory for a new corn stalk ID under the 'Data' directory. This subdirectory
will hold the
measurement data files for this corn stalk.
    """
    old_corn_dir = self.corn_dir
    self.corn_dir = 'Data/' + new_stalk_name
    try:
        os.mkdir(self.corn_dir)
        return True
    except FileNotFoundError:
        # Usually due to the 'Data' directory not existing: create it on move on
        self.create_data_directory()
        os.mkdir(self.corn_dir)
        return True
    except FileExistsError:
        print('Note: this corn stalk ID already has a directory.
        'This may mean that you have already measured this particular corn stalk.')
        return True
    except OSError:
        print('Failed to create new directory: check for illegal characters \ / * ? : " < > | '
in corn stalk name')
        self.corn_dir = old_corn_dir
        return False

def take_measurement(self):
    """ Commands the Analog Discovery to take a measurement. Collects and saves the measurement.
This function will create a new data file (text format) each time it is called. This file has the
format:
<time stamp>_<stalk name>_<point count>_<pass count>.txt

Also, the point and pass count, as well as stalk name, are retrieved from the MainWindow GUI class
instance.

The point and pass count are reported to the user. The elapsed time is also reported to the user.
    """
    if not self.discovery_is_connected:
        print('Warning: No Analog Discovery connected')
    elif self.corn_dir == '':
        print('Specify stalk name before taking measurement')
else:
    pass_num = self.gui_window.pass_count
    point_num = self.gui_window.point_count
    time_stamp = time.strftime("%Y%m%d_%H_%M_%S", time.localtime())
    file_name = time_stamp + '_' + self.gui_window.stalk_name.get() + \\
                '_point' + str(point_num) + '_pass' + str(pass_num) + '.txt'
    start_time = time.time()
    data_file = open((self.corn_dir + '/' + file_name), 'w')
    try:
        measurement = self.discovery.measurement_run_sweep(self.start_freq, self.end_freq,
               self.steps, self.data_1_id, self.data_2_id)
        # Write data_1, data_2, Hz
        data_file.write(str(measurement[0]) + '\t' + 
                        str(measurement[1]) + '\t' + 
                        str(measurement[2]) + '\n')
        data_file.close()
        print('Measured point ', str(self.gui_window.point_count) + 
              ' - pass ', str(self.gui_window.pass_count), 
              ' - Elapsed time: ', str(time.time() - start_time))
        self.gui_window.update_point_count(increment=True)
    except Exception as error:
        print(error)

if __name__ == '__main__':
    print('Starting Greenhouse Measurement Program')
    app = GreenHouseApplication()
    app.run()

B.3 MainWindow.py

The following code is the backend code for the GUI.

"""
Mavrik Thomas
BYU
July 7, 2020

This is GUI for the greenhouse measurement code. The GUI allows the user to issue commands regarding the following:
- Connecting to the Analog Discovery
- Updating the corn stalk name/ID
- Taking measurements
- Controlling the point and pass counts

The GUI also provides visual feedback about the following:
- Analog Discovery connection status
- Current corn stalk name/ID
- Pass and point count

This code is written in Python 3.6

version 1.1
about this version:
- Initial release send to Iowa for greenhouse testing
- Added detailed comments; cleaned up code
- Added functionality to check for valid stalk ID’s to ensure directories are created

"""

import tkinter as tk
class MainWindow:
    def __init__(self, master, application):
        """ Variables, frames, buttons, and so on are set up
        """
        # Tk root and Greenhouse application
        self.master = master
        self.app = application

        # String variables
        self.stalk_name = tk.StringVar()
        self.stalk_name.set('')
        self.connection_status_text = tk.StringVar()

        # Pass and Point counts
        self.pass_count = 1
        self.pass_count_tk = tk.StringVar()
        self.pass_count_tk.set(str(self.pass_count))
        self.point_count = 1
        self.point_count_tk = tk.StringVar()
        self.point_count_tk.set(str(self.point_count))

        # Create main widgets and stuff
        self.create_main_window()
        self.frames_init()
        self.create_widgets()
        self.master.bind('<Return>', self._update_stalk_name)

    def create_main_window(self):
        """ Names main window
        """
        :return: Nothing
        """
        self.master.title('Greenhouse Measurement Program - v' + str(self.app.version))

    def frames_init(self):
        """ Creates and packs the primary frames for the GUI window. The frames are mainly organized by function.
        """
        :return: nothing
        """
        # Primary frames
        frame_program_title = tk.Frame(self.master)
        frame_settings = tk.Frame(self.master)
        self.frame_display_stalk_name = tk.Frame(self.master, relief='sunken', borderwidth=1)
        frame_pass_point = tk.Frame(self.master)
        self.frame_run_measurement = tk.Frame(self.master, relief='sunken', borderwidth=1)

        frame_program_title.pack(side='top')
        frame_settings.pack(side='top')
        self.frame_display_stalk_name.pack(side='top', fill='x', expand=True, padx=10, pady=5)
        frame_pass_point.pack(side='top', fill='x', expand=True, padx=10, pady=5)
        self.frame_run_measurement.pack(side='top', fill='x', expand=True, padx=10, pady=5)

        # Secondary Frames
        self.frame_inner_title = tk.Frame(frame_program_title)
        self.frame_discovery_connection = tk.Frame(frame_settings, relief='sunken', borderwidth=1)
        self.frame_corn_stalk_entry = tk.Frame(frame_settings, relief='sunken', borderwidth=1)
        self.frame_pass = tk.Frame(frame_pass_point, relief='sunken', borderwidth=1)
        self.frame_point = tk.Frame(frame_pass_point, relief='sunken', borderwidth=1)

        self.frame_inner_title.pack(fill='x', expand=True, padx=10, pady=5)
        self.frame_discovery_connection.pack(side='left', fill='x', expand=True, padx=10, pady=5)
        self.frame_corn_stalk_entry.pack(side='left', fill='x', expand=True, padx=10, pady=5)
def create_widgets(self):
    """ Populates the features of the primary GUI frames.
    """
    :return: nothing
    
    self.pack_inner_title()
    self.pack_discovery_connection()
    self.pack_cornstalk_entry()
    self.pack_cornstalk_id()
    self.pack_pass_point()
    self.pack_run_measurement()
    # self.pack_quit()

def pack_inner_title(self):
    """ Creates the main window title
    """
    :return: nothing
    
    self.inner_title = tk.Label(self.frame_inner_title, font=('Rockwell', 20), borderwidth=2)
    self.inner_title['text'] = 'Greenhouse Testing'
    self.inner_title.pack(side='top')

def pack_quit(self):
    """ (Defunct) Creates the quit button
    """
    :return: nothing
    
    self.quit_button = tk.Button(self.frame_quit, text='QUIT', font=('Arial', 16), fg='red',
                                command=self.close_window, border=5)
    self.quit_button.pack(side='top', pady=2, padx=5, expand=True)

def pack_discovery_connection(self):
    """ Creates the features for the Analog Discovery connection. Includes buttons and a status label.
    """
    :return: nothing
    
    # Create two main frames for buttons and label
    title_frame = tk.Frame(self.frame_discovery_connection)
    button_frame = tk.Frame(self.frame_discovery_connection)
    title_frame.pack(side='top')
    button_frame.pack(side='top')
    # Create frames for connect and disconnect button, as well connection status
    connect_frame = tk.Frame(button_frame)
    disconnect_frame = tk.Frame(button_frame)
    status_frame = tk.Frame(button_frame)
    # Pack them up
    connect_frame.pack(side='left')
    status_frame.pack(side='left')
    disconnect_frame.pack(side='left')
    # Title Label
    connection_title = tk.Label(title_frame, text='Analog Discovery Status:', font=('Arial', 14))
    connection_title.pack(pady=5)
    # Create Button
    connect_button = tk.Button(connect_frame, text='Connect', command=self.app.connect_discovery,
                                font=('Arial', 12), fg='blue')
    disconnect_button = tk.Button(disconnect_frame, text='Disconnect',
                                   command=self.app.disconnect_discovery,
                                   font=('Arial', 12), fg='blue')
    # Status Label
    self.connection_status_text.set('Disconnected')
    self.connection_status_label = tk.Label(status_frame, textvariable=self.connection_status_text)
    self.connection_status_label.config(font=('Arial', 12), fg='red')
# Pack it up
connect_button.pack(padx=10, pady=5)
disconnect_button.pack(padx=10, pady=5)
self.connection_status_label.pack(padx=10, pady=5)

# Create Variable

def close_window(self):
    ""
    (Defunct) Called by the quit button to close the window.
    ""
    :return: nothing
    :""
    self.master.destroy()
    self.app.quit_program()

def pack_corn_stalk_entry(self):
    ""
    Creates the entry field for the corn stalk ID/name. The entry field interacts with the corn stalk ID string variable and indirectly interacts with the GreenHouse app in creating the directory for this stalk.
    ""
    :return: nothing
    :""
    # Create title, function, and status frames
    title_frame = tk.Frame(self.frame_corn_stalk_entry)
    function_frame = tk.Frame(self.frame_corn_stalk_entry)

    title_frame.pack(side='top')
    function_frame.pack(side='top')

    # Title Frame
    corn_stalk_title = tk.Label(title_frame, text='Corn Stalk Name', font=('Arial', 14))
    corn_stalk_title.pack(pady=5)

    # Create subframes & pack them
    entry_frame = tk.Frame(self.frame_corn_stalk_entry)
    set_button_frame = tk.Frame(self.frame_corn_stalk_entry)
    entry_label_frame = tk.Frame(self.frame_corn_stalk_entry)

    entry_frame.pack(padx=10, pady=5, side='left')
    set_button_frame.pack(padx=10, pady=5, side='left')
    entry_label_frame.pack(padx=10, pady=5, side='bottom')

    # Create entry
    self.corn_name_entry = tk.Entry(entry_frame)

    # Create configure button
    set_button = tk.Button(set_button_frame, text='Update', font=('Arial', 12), fg='black',
                           border=2, command=self._update_stalk_name)

    # Create current test label
    current_test_label = tk.Label(entry_frame, text='Stalk Name: ')

    # Pack them up
    current_test_label.pack(side='left')
    self.corn_name_entry.pack(side='left')
    set_button.pack(side='left')

    # Pack the current name in use
    current_name_label = tk.Label(status_frame, text='Current Stalk: ', font=('Arial', 14))
    name_status_label = tk.Label(status_frame, textvariable=self.stalk_name, font=('Arial', 14),
                                 justify='left', wraplength=200)
def pack_pass_point(self):
    """ Creates the features for the point-pass counts. Includes buttons to increment, decrement, and reset the
count. Also includes a status label for the current count of each.

:return: nothing
""
# Variables
button_height = 3
button_width = 10
button_font_size = 12
# Main frames
pass_frame = tk.Frame(self.frame_pass)
point_frame = tk.Frame(self.frame_point)

point_frame.pack(side='top', pady=5)
pass_frame.pack(side='top', pady=5)

###  Pass Items  ###
# Pass subframes
pass_label_frame = tk.Frame(pass_frame)
pass_functions_frame = tk.Frame(pass_frame)
pass_label_frame.pack(side='top', pady=2)
pass_functions_frame.pack(side='top', pady=2)

# Pass tertiary frames
pass_buttons_frame = tk.Frame(pass_functions_frame)
pass_count_label_frame = tk.Frame(pass_functions_frame)
pass_count_label_frame.pack(side='top')
pass_buttons_frame.pack(side='top')

# Pass Label
pass_label = tk.Label(pass_label_frame, text='Pass', font=('Arial', 14))
pass_label.pack(side='top', pady=10)

# Pass buttons
pass_dec_button = tk.Button(pass_buttons_frame, text='-', font=('Arial', button_font_size),
fg='blue',
command=lambda: self.update_pass_count(False, False),
width=button_width, height=button_height)
pass_inc_button = tk.Button(pass_buttons_frame, text='+', font=('Arial', button_font_size),
fg='blue',
command=lambda: self.update_pass_count(True, False),
width=button_width, height=button_height)
pass_reset_button = tk.Button(pass_buttons_frame, text='Reset', font=('Arial', button_font_size),
fg='red',
command=lambda: self.update_pass_count(False, True),
width=button_width, height=button_height)

pass_dec_button.pack(side='left', padx=5, pady=2)
pass_reset_button.pack(side='left', padx=5, pady=2)
pass_inc_button.pack(side='left', padx=5, pady=2)

# Pass count label
pass_count_label = tk.Label(pass_count_label_frame, textvariable=self.pass_count_tk,
font=('Arial', 32))
pass_count_label.pack(side='top', padx=28, pady=15)

###  Point Items  ###
# Point subframes
point_label_frame = tk.Frame(point_frame)
point_functions_frame = tk.Frame(point_frame)
point_label_frame.pack(side='top', pady=2)
point_functions_frame.pack(side='top', pady=2)
# Point tertiary frames
point_buttons_frame = tk.Frame(point_functions_frame)
point_count_label_frame = tk.Frame(point_functions_frame)
point_count_label_frame.pack(side='top')
point_buttons_frame.pack(side='top')

# Point Label
point_label = tk.Label(point_label_frame, text='Point', font=('Arial', 14))
point_label.pack(side='top', pady=10)

# Point buttons
point_dec_button = tk.Button(point_buttons_frame, text='-', font=('Arial', button_font_size), fg='blue',
                             command=lambda: self.update_point_count(False, False),
                             width=button_width, height=button_height)
point_dec_button.pack(side='left', padx=5, pady=2)

point_inc_button = tk.Button(point_buttons_frame, text='+', font=('Arial', button_font_size), fg='blue',
                            command=lambda: self.update_point_count(True, False),
                            width=button_width, height=button_height)
point_inc_button.pack(side='left', padx=5, pady=2)

point_reset_button = tk.Button(point_buttons_frame, text='Reset', font=('Arial', button_font_size), fg='red',
                                command=lambda: self.update_point_count(False, True),
                                width=button_width, height=button_height)
point_reset_button.pack(side='left', padx=5, pady=2)

# Point count label
point_count_label = tk.Label(point_count_label_frame, textvariable=self.point_count_tk, font=('Arial', 32))
point_count_label.pack(side='top', padx=20, pady=15)

def pack_run_measurement(self):
    """ Creates the 'Measurement' button for taking measurements
    Interacts with GreenHouse app to take measurements.
    :return: nothing
    """
    # Run measurement button
    run_measurement_button = tk.Button(self.frame_run_measurement, text='Measure', font=('Arial', 26),
                                        fg='blue',
                                        command=self.app.take_measurement, activeforeground='red')
    run_measurement_button.pack(pady=10, padx=5)

def update_stalk_name(self, event=None):
    """ Backend function: takes string from entry field. This string is checked for validity and is
    passed on to the Greenhouse app to create the directory. The string variable is updated as well.
    A valid string (i.e. a correct corn stalk name/ID) should be non-blank and contain no illegal
    characters that will prevent a directory from being created: / * ? : " < > |
    :param event: Used to bind the 'Enter' button to this function; technically required to bind, but
    not used.
    :return: nothing
    """
    try:
        if not self.corn_name_entry.index('end') == 0:
            new_stalk_name = self.corn_name_entry.get()
            new_stalk_name = new_stalk_name.strip()  # Remove leading and trailing spaces

            if self.app.create_corn_stalk_directory(new_stalk_name):
                self.stalk_name.set(new_stalk_name)
self.corn_name_entry.delete(0, 'end')
self.update_pass_count(False, True)
self.update_point_count(False, True)
print('Current Stalk: ', new_stalk_name)
else:
    self.corn_name_entry.delete(0, 'end')

except Exception as error:
    print('Update stalk name error: ', error)

def update_point_count(self, increment=True, reset=False):
    """ Updates or resets the point count.
    This function is called automatically when:
    - Measurement is taken (incremented)
    - New corn stalk ID entered (reset)
    - Pass count incremented (reset)
    :param increment: Increment count if true, decrement count if false
    :param reset: If True, set the count to one; overrides increment flag
    :return: nothing
    ""
    try:
        if reset:
            self.point_count = 1
        elif increment:
            self.point_count += 1
        else:
            self.point_count -= 1
            if self.point_count < 1:
                self.point_count = 1
        self.point_count_tk.set(str(self.point_count))
    except Exception as error:
        print('Update Point Count Error: ', error)

def update_pass_count(self, increment=True, reset=False):
    """ Updates or resets the pass count.
    This is called automatically when:
    - New corn stalk ID entered (reset)
    This function resets the point count if the pass count is incremented or reset.
    :param increment: Increment count if true, decrement count if false
    :param reset: If True, set the count to one; overrides increment flag
    :return: nothing
    ""
    try:
        if reset:
            self.pass_count = 1
            self.update_point_count(False, True)
        elif increment:
            self.pass_count += 1
            self.update_point_count(False, True)
        else:
            self.pass_count -= 1
            if self.pass_count < 1:
                self.pass_count = 1
        self.pass_count_tk.set(str(self.pass_count))
    except Exception as error:
        print('Update Point Count Error: ', error)

def update_status_indicators(self):
    """ Updates the connection status and other visual indicators. Called by the mainloop of the
    Greenhouse app.
    :return: nothing
    ""
    try:
if self.app.discovery_is_connected:
    self.connection_status_text.set('Connected')
    # self.connection_status_label['textvariable'] = self.connection_status_text
    self.connection_status_label.config(font=('Arial', 12), fg='green')
else:
    self.connection_status_text.set('Disconnected')
    # self.connection_status_label['textvariable'] = self.connection_status_text
    self.connection_status_label.config(font=('Arial', 12), fg='red')

except Exception as error:
    pass

B.4 AnalogDiscovery.py

The following code is the backend code for the AD2. It handles connection, configuration, and measurements.

---
Mavrik Thomas
March 24, 2020

Analog Discovery Class
This code is based on example code provided by Digilent Inc. and contains all the necessary code to measure impedance with an Analog Discovery 2.

(see https://github.com/amuramatsu/dwf for example code)

---

import sys
from ctypes import *
from Tools.dwfconstants import *
import time
import math

class AnalogDiscovery:
    def __init__(self):
        # Print statement flags
        self.print_connect = False

        # Device setup
        self.dwf, self.device = self.anadisc_connect(self.print_connect)
        self._zero_mask = 0x0000
        self.max_voltage = 5  # Volts
        self.r_ref = 1e4

        # Analog Channel - **WARNING** Reference only, put c_type(##) directly into argument; self.<var> will not work
        self._anchnnl_pos_volt = c_int(0)
        self._anchnnl_neg_volt = c_int(1)
        self._anchnnl_usb = c_int(2)
        self._anchnnl_aux = c_int(3)

        # Voltage Nodes (channels 0 and 1)
        self._volt_node_enable = c_int(8)  # Enable/disable channel
        self._volt_setting = c_int(1)  # Voltage setting

        # Measurement exception
        class MeasurementError(Exception):
            pass
```python
def __del__(self):
    try:
        self.dwf.FDwfDeviceCloseAll()
    except Exception as error:
        print('Analog Discovery - Hard Close:', error)

def close(self):
    try:
        self.dwf.FDwfDeviceCloseAll()
    except Exception as error:
        print('Analog Discovery - Failed to close:', error)

Connect to Analog Discovery; Set Up
...

@staticmethod
def anadisc_connect(print_statements=False):
    """Connects to an Analog Discovery. Code based on example code from SDK."
    :return: Analog Discovery object
    """
    # Determine operating system
    if sys.platform.startswith("win"):  
        dwf = cdll.LoadLibrary("dwf.dll")
    elif sys.platform.startswith("darwin"):  
        dwf = cdll.LoadLibrary("/Library/Frameworks/dwf.framework/dwf")
    else:
        dwf = cdll.LoadLibrary("libdwf.so")

    # Determine DWF version
    version = create_string_buffer(16)
    dwf.FDwfGetVersion(version)
    if print_statements:
        print("DWF Version: " + str(version.value.decode('utf-8')))

    # Get Analog Discovery object
    hdwf = c_int()
    szerr = create_string_buffer(512)
    if print_statements:
        print("Opening first device")
    dwf.FDwfDeviceOpen(c_int(-1), byref(hdwf))

    # If failed
    if hdwf.value == hdwfNone.value:
        dwf.FDwfGetLastErrorMsg(szerr)
        print("Failed to open device:", str(szerr.value.decode('utf-8')))
        quit()

    # this option will enable dynamic adjustment of analog out settings like: frequency, amplitude...
    dwf.FDwfDeviceAutoConfigureSet(hdwf, c_int(3))

    return dwf, hdwf

def set_reference_resistor(self, value):
    """Sets the reference resistor value
    :param value: int of resistor value in ohms (e.g. 10k = 10000)
    :return: nothing
    """
    value = int(float(value))  # Convert to int if necessary
    if value > 0:
        self.r_ref = value
    else:
        print('Analog Discovery: invalid reference resistor value')

def get_reference_resistor(self):
    """"""
    :return: Value of current reference resistor
```

Voltage Control

```python
def voltage_set_positive(self, volt_value):
    
    # Sets the voltage of the positive rail.
    
    Note: double check that voltage was set correctly with multimeter; some values are finicky.
    
    :param volt_value: Float to set positive voltage rail.
    :return: nothing
    
    if not 0 <= volt_value <= self.max_voltage:
        print('AD error: invalid voltage')
    return

    # enable positive supply
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(0), c_int(0), c_double(True))
    # set voltage to 5 V
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(0), c_int(1), c_double(volt_value))
    # master enable
    self.dwf.FDwfAnalogIOEnableSet(self.device, c_int(True))

def voltage_get_positive(self):
    
    # Gets value of positive voltage rail.
    
    :return: Float of positive voltage rail value
    
    read_voltage = c_double()
    self.dwf.FDwfAnalogIOChannelNodeStatus(self.device, self._anchnnl_pos_volt,
                                            self._volt_setting, byref(read_voltage))
    return read_voltage.value

def voltage_set_negative(self, volt_value):
    
    # Sets the voltage of the negative rail.
    
    Note: double check that voltage was set correctly with multimeter; some values are finicky.
    Note: has not been verified to work with a value different than the negative of the positive voltage rail (e.g. -3.3V and +5.0V). Proceed at own risk.
    
    :param volt_value: Float to set negative voltage rail.
    :return: nothing
    
    if not -self.max_voltage <= volt_value <= 0:
        print('AD error: invalid voltage')
    return

    # enable negative supply
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(1), c_int(0), c_double(True))
    # set voltage to -5 V
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(1), c_int(1), c_double(volt_value))
    # master enable
    self.dwf.FDwfAnalogIOEnableSet(self.device, c_int(True))

def voltage_get_negative(self):
    
    # Gets value of negative voltage rail
    
    :return: Float value of negative voltage rail
    
    read_voltage = c_double()
    self.dwf.FDwfAnalogIOChannelNodeStatus(self.device, self._anchnnl_neg_volt,
                                            self._volt_setting, byref(read_voltage))
    return read_voltage.value

def voltage_enable_master(self):
```

def voltage_reset_all(self):
    """ Turns off power supply
    :return: nothing
    """
    # master disable
    self.dwf.FDwfAnalogIOEnableSet(self.device, c_int(False))

def voltage_test(self):
    """ Basic test for setting voltages. Sets voltage to 3.3V.
    Check with a multimeter
    :return: nothing
    """
    # set up analog IO channel nodes
    # enable positive supply
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(0), c_int(0), c_double(True))
    # set voltage to 3.3 V
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(0), c_int(1), c_double(3.3))
    # enable negative supply
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(1), c_int(0), c_double(True))
    # set voltage to -3.3 V
    self.dwf.FDwfAnalogIOChannelNodeSet(self.device, c_int(1), c_int(1), c_double(-3.3))
    # master enable
    self.dwf.FDwfAnalogIOEnableSet(self.device, c_int(True))

... Digital IO Tools ...

def digital_out_reset(self):
    """ Turns off all digital outputs
    :return: nothing
    """
    # digital IO output enable
    self.dwf.FDwfDigitalIOOutputEnableSet(self.device, c_int(self._zero_mask))

def digital_out_configure(self, channel_mask):
    """ Specifies which digital IO pins can be used. Note that it does not
    activate any of them; self.digital_out_set() actives the IO pins
    See example code and SDK manual.
    :param channel_mask: hexadecimal integer mask
    :return: nothing
    """
    self.dwf.FDwfDigitalIOOutputEnableSet(self.device, c_int(channel_mask))

def digital_out_set(self, channel_mask):
    """ Actives digital IO pins. Note that self.digital_out_configure()
    must be called first before using this function.
    See example code and SDK manual.
    :param channel_mask: hexadecimal integer mask
    :return: nothing
    """
    self.dwf.FDwfDigitalIOOutputSet(self.device, c_int(channel_mask))

def digital_out_get_status(self):
    """ Gets status of IO pins
    :return: value of which IO pins are active
    """
read = c_uint32()
# fetch digital IO information from the device
self.dwf.FDwfDigitalIOModeSet(self.device)
self.dwf.FDwfDigitalIOModeSelect(self.device, byref(read))
return bin(read.value)[2:].zfill(16)

... Take Measurement Tools ...

```
def measurement_take_single(self, Hz, step_count, data_1_id, data_2_id, voltage=1):
    """ Takes measurements at a single frequency. """

    :param Hz: frequency to measure at
    :param step_count: how many measurements to take (elements in output arrays)
    :param data_1_id: DWF constant value for measurement type (see SDK manual)
    :param data_2_id: DWF constant value for measurement type (see SDK manual)
    :param voltage: Voltage to take measurement at
    :return: two arrays: data 1 and data 2

    sts = c_byte()
szerr = create_string_buffer(512)
steps = step_count

    reference = self.r_ref

    if voltage > 5:
        print('AD Measurement error: Invalid voltage - using 1V')
        voltage = 1

    self.dwf.FDwfAnalogImpedanceReset(self.device)
    self.dwf.FDwfAnalogImpedanceModeSet(self.device, c_int(1))
    self.dwf.FDwfAnalogImpedanceReferenceSet(self.device, c_double(reference))
        # reference resistor value in Ohms
    self.dwf.FDwfAnalogImpedanceFrequencySet(self.device, c_double(Hz))
        # frequency in Hertz
    self.dwf.FDwfAnalogImpedanceAmplitudeSet(self.device, c_double(voltage))
        # 1V amplitude = 2V peak2peak signal
    self.dwf.FDwfAnalogImpedanceConfigure(self.device, c_int(1))

    time.sleep(0.001)

    # Set up variables to hold data
    rgHz = [0.0] * steps
    rgDt1 = [0.0] * steps
    rgDt2 = [0.0] * steps

    # Run frequency sweep
    start_time = time.time()
    for i in range(steps):
        while True:
            if self.dwf.FDwfAnalogImpedanceStatus(self.device, byref(sts)) == 0:
                self.dwf.FDwfGetLastErrorMsg(szerr)
                quit()
            if sts.value == 2:
                break

            # Collect and save data after one frequency step iteration
            raw_data_1 = c_double()
            raw_data_2 = c_double()
            my_hz = c_double()
            self.dwf.FDwfAnalogImpedanceStatusMeasure(self.device, data_1_id, byref(raw_data_1))
            self.dwf.FDwfAnalogImpedanceStatusMeasure(self.device, data_2_id, byref(raw_data_2))
            self.dwf.FDwfAnalogImpedanceFrequencyGet(self.device, byref(my_hz))

            rgDt1[i] = raw_data_1.value  # absolute value for logarithmic plot
            rgDt2[i] = raw_data_2.value

        # Stop measurements
        end_time = time.time()

        self.dwf.FDwfAnalogImpedanceConfigure(self.device, c_int(0))
```
def measurement_run_sweep(self, start_freq, end_freq, step_count, data_1_id, data_2_id):
    """ Takes frequency swept measurements
    :param start_freq: starting frequency (Hz)
    :param end_freq: end frequency (Hz)
    :param step_count: how many frequency bins
    :param data_1_id: DWF constant value for measurement type (see SDK manual)
    :param data_2_id: DWF constant value for measurement type (see SDK manual)
    :param voltage: Voltage to take measurement at
    :return: three arrays: data 1, data 2, and frequency array
    """
    sts = c_byte()
    szerr = create_string_buffer(512)
    steps = step_count
    start = start_freq
    stop = end_freq
    reference = self.r_ref
    # print("Reference: " + str(reference) + " Ohm  Frequency: " + str(start) + " Hz ... " + str(
    #    stop / 1e3) + " kHz for nanofarad capacitors")
    self.dwf.FDwfAnalogImpedanceReset(self.device)
    self.dwf.FDwfAnalogImpedanceModeSet(self.device, c_int(1))
    self.dwf.FDwfAnalogImpedanceReferenceSet(self.device, c_double(reference))
    # reference resistor
    self.dwf.FDwfAnalogImpedanceFrequencySet(self.device, c_double(start))
    self.dwf.FDwfAnalogImpedanceAmplitudeSet(self.device, c_double(0.5))
    self.dwf.FDwfAnalogImpedanceConfigure(self.device, c_int(1))
    time.sleep(0.001)
    # Set up variables to hold data
    rgHz = [0.0] * steps
    rgDt1 = [0.0] * steps
    rgDt2 = [0.0] * steps
    # Run frequency sweep
    start_time = time.time()
    for i in range(steps):
        hz = stop * pow(10.0, 1.0 * i / (steps - 1) - 1) * math.log10(stop / start))  # exponential frequency steps
        rgHz[i] = hz
        self.dwf.FDwfAnalogImpedanceFrequencySet(self.device, c_double(hz))  # frequency in Hertz
        time.sleep(0.001)
        self.dwf.FDwfAnalogImpedanceStatus(self.device, None)  # ignore last capture since we changed
        the frequency
        while True:
            if self.dwf.FDwfAnalogImpedanceStatus(self.device, byref(sts)) == 0:
                self.dwf.FDwfGetLastErrorMsg(szerr)
                raise self.MeasurementError('Measurement Error: Check Analog Discovery Connection')
            if sts.value == 2:
                break
        # Collect and save data after one frequency step iteration
        raw_data_1 = c_double()
        raw_data_2 = c_double()
        my_hz = c_double()
        self.dwf.FDwfAnalogImpedanceStatusMeasure(self.device, data_1_id, byref(raw_data_1))
        self.dwf.FDwfAnalogImpedanceStatusMeasure(self.device, data_2_id, byref(raw_data_2))
        self.dwf.FDwfAnalogImpedanceFrequencyGet(self.device, byref(my_hz))
        rgDt1[i] = raw_data_1.value  # absolute value for logarithmic plot
        rgDt2[i] = raw_data_2.value
    # Stop measurements
    end_time = time.time()
# print('Elapsed measurement time: ', end_time - start_time)
self.dwf.FDwfAnalogImpedanceConfigure(self.device, c_int(0))
# Return phase
return rgDt1, rgDt2, rgHz

# Misc functions

def misc_produce_sinewave(self):
    channel = c_int(0)
    self.dwf.FDwfAnalogOutNodeEnableSet(self.device, channel, AnalogOutNodeCarrier, c_bool(True))
    self.dwf.FDwfAnalogOutNodeFunctionSet(self.device, channel, AnalogOutNodeCarrier, funcSine)
    self.dwf.FDwfAnalogOutNodeFrequencySet(self.device, channel, AnalogOutNodeCarrier, c_double(1000))
    self.dwf.FDwfAnalogOutNodeAmplitudeSet(self.device, channel, AnalogOutNodeCarrier, c_double(1.41))
    self.dwf.FDwfAnalogOutNodeOffsetSet(self.device, channel, AnalogOutNodeCarrier, c_double(1.41))
    print("Generating sine wave...")
    self.dwf.FDwfAnalogOutConfigure(self.device, channel, c Bool(True))

B.5 BitTools.py

The following code contains a function used to create hexadecimal bitmasks for configuring the
digital IO ports of the AD2.

```python
""
Mavrik Thomas
April 10, 2020
This file contains function for generating bit masks for specific tasks
""

def bit_mask_high_to_low(high, low):
    """Generates a hexadecimal bit mask to measure from the high side electrode number to the low side
electrode number. Takes the form 0x00<low><high>
:param high: integer representing the high side electrode
:param low: integer representing the low side electrode
:return: integer representing the hexadecial bit mask
""
    if not (0 <= high <= 7 or 0 <= low <= 7):
        print('Bit Mask Gen: invalid values')
        return

    bit_mask = '0x00' + str(low) + str(high)
    return int(bit_mask, 16)

B.6 dwfconstants.py

The following code is provided in the Digilent WaveForms SDK and is necessary for interfacing
with the AD2.

""
DWFConstants (definitions file for DWF library)
Author: Digilent, Inc.
Revision: 2019-10-15
""
from ctypes import *

# device handle

# HDWF
dwfNone = c_int(0)

# device enumeration filters
enumfilterAll = c_int(0)
enumfilterEExplorer = c_int(1)
enumfilterDiscovery = c_int(2)
enumfilterDiscovery2 = c_int(3)
enumfilterDDiscovery = c_int(4)

# device ID
devidEExplorer = c_int(1)
devidDiscovery = c_int(2)
devidDiscovery2 = c_int(3)
devidDDiscovery = c_int(4)

# device version
devverEExplorerC = c_int(2)
devverEExplorerE = c_int(4)
devverEExplorerF = c_int(5)
devverDiscoveryA = c_int(1)
devverDiscoveryB = c_int(2)
devverDiscoveryC = c_int(3)

# trigger source
trigsrcNone = c_ubyte(0)
trigsrcPC = c_ubyte(1)
trigsrcDetectorAnalogIn = c_ubyte(2)
trigsrcDetectorDigitalIn = c_ubyte(3)
trigsrcAnalogIn = c_ubyte(4)
trigsrcDigitalIn = c_ubyte(5)
trigsrcDigitalOut = c_ubyte(6)
trigsrcAnalogOut1 = c_ubyte(7)
trigsrcAnalogOut2 = c_ubyte(8)
trigsrcAnalogOut3 = c_ubyte(9)
trigsrcAnalogOut4 = c_ubyte(10)
trigsrcExternal1 = c_ubyte(11)
trigsrcExternal2 = c_ubyte(12)
trigsrcExternal3 = c_ubyte(13)
trigsrcExternal4 = c_ubyte(14)
trigsrcHigh = c_ubyte(15)
trigsrcLow = c_ubyte(16)

# instrument states
DwfStateReady = c_ubyte(0)
DwfStateConfig = c_ubyte(4)
DwfStatePrefill = c_ubyte(5)
DwfStateArmed = c_ubyte(1)
DwfStateWait = c_ubyte(7)
DwfStateTriggered = c_ubyte(3)
DwfStateRunning = c_ubyte(3)
DwfStateDone = c_ubyte(2)

# DwfEnumConfigInfo
DECIAnalogInChannelCount = c_int(1)
DECIAnalogOutChannelCount = c_int(2)
DECIAnalogIOChannelCount = c_int(3)
DECDigitalInChannelCount = c_int(4)
DECDigitalOutChannelCount = c_int(5)
DECDigitalIOChannelCount = c_int(6)
DECIAnalogInBufferSize = c_int(7)
DECIAnalogOutBufferSize = c_int(8)
DECIDigitalInBufferSize = c_int(9)
DECIDigitalOutBufferSize = c_int(10)

# acquisition modes:
acqmodeSingle = c_int(0)
acqmodeScanShift = c_int(1)
acqmodeScanScreen = c_int(2)
acqmodeRecord = c_int(3)
acqmodeOvers = c_int(4)
acqmodeSingle1 = c_int(5)

# analog acquisition filter:
filterDecimate = c_int(0)
filterAverage = c_int(1)
filterMinMax = c_int(2)

# analog in trigger mode:
trigtypeEdge = c_int(0)
trigtypePulse = c_int(1)
trigtypeTransition = c_int(2)

# trigger slope:
DwfTriggerSlopeRise = c_int(0)
DwfTriggerSlopeFall = c_int(1)
DwfTriggerSlopeEither = c_int(2)

# trigger length condition
triglenLess = c_int(0)
triglenTimeout = c_int(1)
triglenMore = c_int(2)

# error codes for the functions:
dwfercNoErc = c_int(0) # No error occurred
dwfercUnknownError = c_int(1) # API waiting on pending API timed out
dwfercApiLockTimeout = c_int(2) # API waiting on pending API timed out
dwfercAlreadyOpened = c_int(3) # Device already opened
dwfercNotSupported = c_int(4) # Device not supported
dwfercInvalidParameter0 = c_int(16) # Invalid parameter sent in API call
dwfercInvalidParameter1 = c_int(17) # Invalid parameter sent in API call
dwfercInvalidParameter2 = c_int(18) # Invalid parameter sent in API call
dwfercInvalidParameter3 = c_int(19) # Invalid parameter sent in API call
dwfercInvalidParameter4 = c_int(20) # Invalid parameter sent in API call

# analog out signal types
funcDC = c_ubyte(0)
funcSine = c_ubyte(1)
funcSquare = c_ubyte(2)
funcTriangle = c_ubyte(3)
funcRampUp = c_ubyte(4)
funcRampDown = c_ubyte(5)
funcNoise = c_ubyte(6)
funcPulse = c_ubyte(7)
funcTrapezium = c_ubyte(8)
funcSinePower = c_ubyte(9)
funcCustom = c_ubyte(30)
funcPlay = c_ubyte(31)

# analog io channel node types
analogioEnable = c_ubyte(1)
analogioVoltage = c_ubyte(2)
analogioCurrent = c_ubyte(3)
analogioPower = c_ubyte(4)
analogioTemperature = c_ubyte(5)

AnalogOutNodeCarrier = c_int(0)
AnalogOutNodeFM = c_int(1)
AnalogOutNodeAM = c_int(2)
DwfAnalogOutIdleDisable = c_int(0)
DwfAnalogOutIdleOffset = c_int(1)
DwfAnalogOutIdleInitial = c_int(2)

DwfDigitalInClockSourceInternal = c_int(0)
DwfDigitalInClockSourceExternal = c_int(1)

DwfDigitalInSampleModeSimple = c_int(0)
DwfDigitalInSampleModeNoise = c_int(1)
# alternate samples: noise|sample|noise|sample|...
# where noise is more than 1 transition between 2 samples

DwfDigitalOutOutputPushPull = c_int(0)
DwfDigitalOutOutputOpenDrain = c_int(1)
DwfDigitalOutOutputOpenSource = c_int(2)
DwfDigitalOutOutputThreeState = c_int(3)

DwfDigitalOutTypePulse = c_int(0)
DwfDigitalOutTypeCustom = c_int(1)
DwfDigitalOutTypeRandom = c_int(2)
DwfDigitalOutTypeROM = c_int(3)

DwfDigitalOutIdleInit = c_int(0)
DwfDigitalOutIdleLow = c_int(1)
DwfDigitalOutIdleHigh = c_int(2)
DwfDigitalOutIdleZet = c_int(3)

DwfAnalogImpedanceImpedance = c_int(0)
DwfAnalogImpedanceImpedancePhase = c_int(1)
DwfAnalogImpedanceResistance = c_int(2)
DwfAnalogImpedanceReactance = c_int(3)
DwfAnalogImpedanceAdmittance = c_int(4)
DwfAnalogImpedanceAdmittancePhase = c_int(5)
DwfAnalogImpedanceSusceptance = c_int(6)
DwfAnalogImpedanceSeriesCapactance = c_int(8)
DwfAnalogImpedanceParallelCapacitance = c_int(9)
DwfAnalogImpedanceSeriesInductance = c_int(10)
DwfAnalogImpedanceParallelInductance = c_int(11)
DwfAnalogImpedanceDissipation = c_int(12)
DwfAnalogImpedanceQuality = c_int(13)

DwfParamUsbPower = c_int(2) # 1 keep the USB power enabled even when AUX is connected, Analog Discovery 2
DwfParamLedBrightness = c_int(3) # LED brightness 0 ... 100%, Digital Discovery
DwfParamOnClose = c_int(4) # 0 continue, 1 stop, 2 shutdown
DwfParamAudioOut = c_int(5) # 0 disable / 1 enable audio output, Analog Discovery 1, 2
DwfParamUsbLimit = c_int(6) # 0..1000 mA USB power limit, -1 no limit, Analog Discovery 1, 2

# obsolete
#STS
stsRdy = c_ubyte(0)
stsArm = c_ubyte(1)
stsDone = c_ubyte(2)
stsTrig = c_ubyte(3)
stsCfg = c_ubyte(4)
stsPrefill = c_ubyte(5)
stsNotDone = c_ubyte(6)
stsTrigDly = c_ubyte(7)
stsError = c_ubyte(8)
stsBusy = c_ubyte(9)
stsStop = c_ubyte(10)

#TRIGCOND
trigcondRisingPositive = c_int(0)
trigcondFallingNegative = c_int(1)