RFID Implementation of CSU ID Cards

Taylor Nicholas Perry, BA Computer Science

Charleston Southern University, Department of Computer Science

Senior Project Design Proposal

Contents

[Purpose Statement 4](#_Toc39488104)

[Schedule 5](#_Toc39488105)

[Design: Component 1: RFID Tag/student ID 7](#_Toc39488106)

[Design Component 2: RFID Reader 9](#_Toc39488107)

[Design component 3: Compiler code 12](#_Toc39488108)

[Design Component 4: tagToID.exe 15](#_Toc39488109)

[Future Research Efforts 16](#_Toc39488110)



[This Photo](https://www.electronics-lab.com/project/rfid-rc522-arduino-uno-oled-display/) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

# Purpose Statement

RFID Implementation of CSU ID Cards.

The Idea of this project is to build and introduce RFID readers to be able to read student ID’s at Charleston Southern to make certain processes easier as a student. The primary problem I wish to simplify with this project is to make the hassle of leaving Chapel easier. The current process is everyone leaves at once and at each door there is a scanner that reads the bar code on a students ID. It works well but unfortunately with hundreds of students leaving at once and different lighting conditions this could make scanning the bar code difficult and could with hundreds of cards to scan the wasted seconds add up. With RFID Student ID cards all students would have to do is tap their cards on a RFID reader (a scanner) and they would be on their way. This gives the student more time to get lunch (which everyone is already going that way) or more time before their next class to get ready.

If I finish this part early there are more opportunities that can be implemented. For example there already exists RFID readers and cards at CSU. If you are a Nursing or Graphics design major you are allowed into certain rooms only for those majors. The problem here is that the student has to not only keep up with their ID but a RFIC Key when those could just be one item. The problem that surrounds the Graphics Design students is that there is only enough RFID keys for the majors but not the minors, which means the minor students cant access the software they need after class hours to complete their work. If their Student ID’s were also RFID’able then this would no longer be a problem since the teacher could just request the minor students access to Graphic Design RFID reader(s). The RFID system would allow students into certain room showing when and where they scanned adding to security for the school.

Things I will need to do (as far as I know right now)

* Figure out how the current scanning of student ID’s is done and the process surrounding that. (What happens to the information once its scanned, is there a database?, can RFID scanners replace the barcode scanners? )
* Make a RFID reader and database to read students information create a practice simulation.
* Make a web page for when you: lose your ID and want to revoke its accesses, make a new card (new student).

If more time was allowed I would like to make a case for the scanner and battery for to make it look more professional (rather than cables and a weird looking science experiment). I would also like to see where I can expand on the previous ideas with outlook from teachers and students

# Schedule

Schedule for Senior Project

**During School:**

Week 1 (1/13-1/17)

* Class Sign up and setting up adviser

Week 2 (1/20- 1/24)

* Talk with Advisor and make sure that everything is all good and if need to make changes do so this week.
* Make up for Project implementation (497), redo that.

Week 3(1/27 – 1/31)

* Talk to the networking people at csu to see how it would work to implement a RFID reader onto campus
* Look at what data they are looking to read (names, times, dates, what other things do I need for a test environment)
* Start making prototype RFID reader to get a working model and start messing around.
* Think if you want a throwaway design or spiral prototype design

Week 4(2/3- 2/7)

* Make sure that you have everything you need for the back end of things from the networking people. (database information, API’S)
* Continue to test and learn how RFID and how it read/writes
* Begin design documentation

Week 5 (2/10-2/14)

* Acquire rfid tags as test to see how the prototype program does with any kind of RFID tags
* Start looking into how to make a web page to test services.
* continue design documentation

Week 6 (2/17- 2/21)

* Continue to tinker with RFID cards and building tests (Not close to the final product)
* Think of how to make the RFID system compact and into a handheld system.
* Design complete ()
* Design review.

Week 7 (2/24-2/28)

* If design is done: make a schedule for coding (**milestone**)
* Design review

**Over the summer**

* Milestone 1:

Buy RFID tags (1k classic tags) and also SD/usb port for arduino.

* Milestone 2:

Successfully read in multiple RFID cards to be accessed by a computer or

* Milestone 3:

Create the software that compares the list of RFID tags to were all the student information and then yield a CSV file that office staff can use for their records.

* Milestone 4:

Polish all paperwork and make sure the documentation is presentable

# Design: Component 1: RFID Tag/student ID

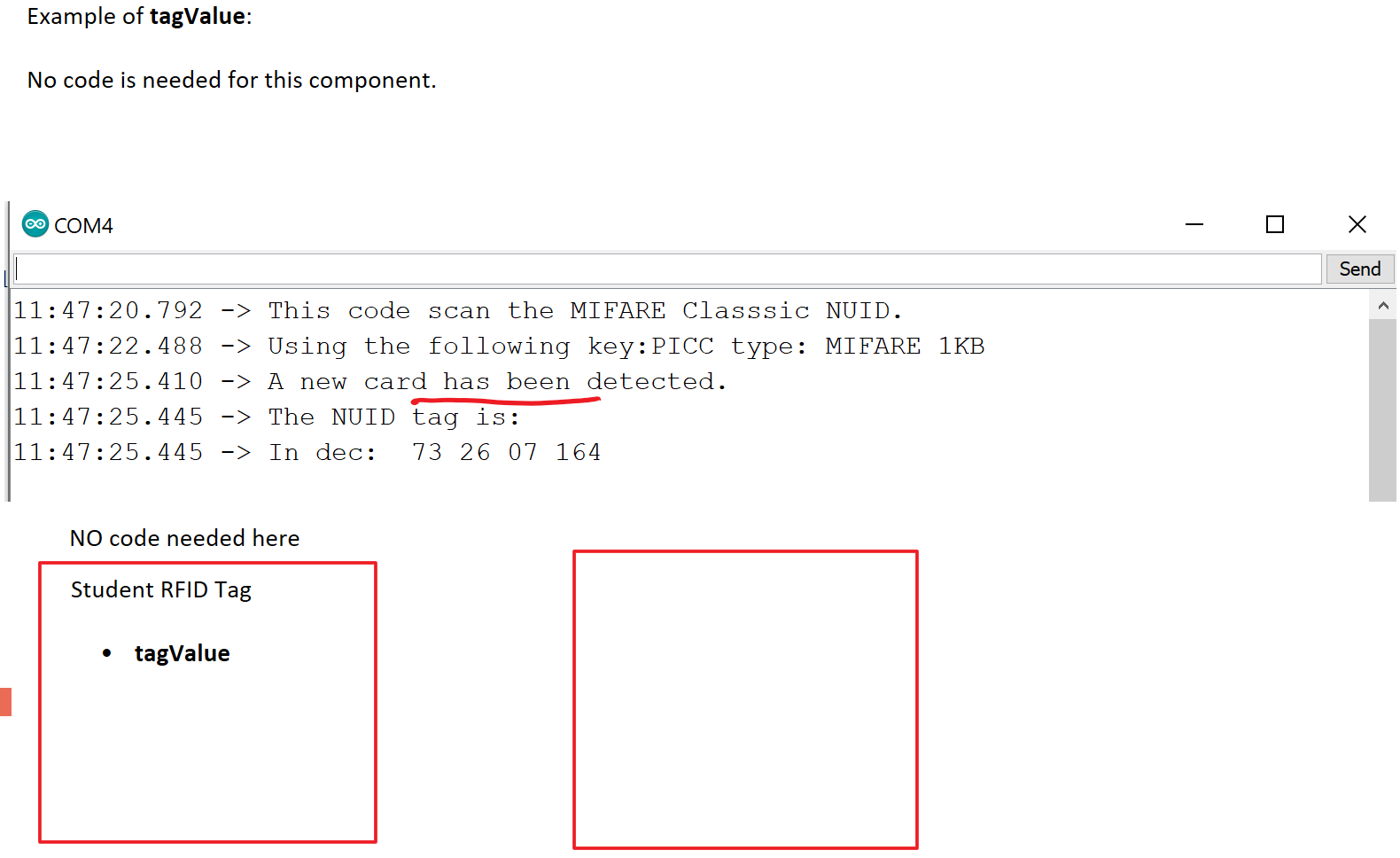
Purpose: This works as identification for the student. This should hold the same info that a normal ID has currently, but the card would have a RFID tag in the student ID holding the **tagValue** of each student. The **tagValue** will be a 9 digit number value that cannot be changed on a Student ID.

If a student were to lose their Student ID then how would security deactivate their card? Since the value on the card is a permanent value it cannot be changed, the **tagValue** can be re-assigned if the student has lost their card. For example is John Doe had the **tagValue** of 123456789, but lost hid card, the lost card will always have he value of 123456789. The security/IT would have to go to his name in the database/spread sheet and change his **tagValue** of 123456789 to the new RFID **tagvalue** of a new card. This way old **tagValue** cannot be associated with him.

Why do you need this component:

Every student has a Student ID, it is swiped for meal points and a barcode is scanned to access the students ID number. The Student RFID will still have those options to swipe and scan but it will have the RFID to allow for a better interaction of getting students information with or without annoyance of duplicate scans or trouble scanning.

Interaction:

The Student RFID tag will only communicate with the RFID Reader. Only the **tagValue** will be interacting with RFID reader. 

# Design Component 2: RFID Reader

Purpose: This component is responsible for the following :

* Reading Student RFID Values**: tagValues**
* Storing all **tagValues** in a value that will be called **tagList**
* **tagList**(list of all scanned **tagValues**) will be sent to **tagToID.exe** that will take **tagList** and match the **tagValues** to student's names

The **tagValue** is the Student RFID's value being read from the RFID tag from component 1. The purpose of the RFID Reader is to read in the RFID values, make a list of all the values, then in that new list of values will interact with the search program. The code used for the Reader will be C++ based.

How: The RFID Reader will use sample libraries provided by Arduino to access and use the RFID-RC522 reader. Those libraries are #include <SPI.h>, <MFRC522.h>. These libraries allow you to use keyword functions to use certain pin numbers on the RFID-RC522 module and to read/write on RFID tags. The library of "MFRC522" is an open source created by Miguelbalboa. <<https://github.com/miguelbalboa/rfid>>

Interaction:

A Student RFID **tagValue**  will tap on the RFID-RC522 module on the Reader and it will record the **tagValue**  and store it in a list to be given to the **tagToID.exe** to then be used in that component. The Code to tell what the RFID Reader what to do will be on Component 3: computer.

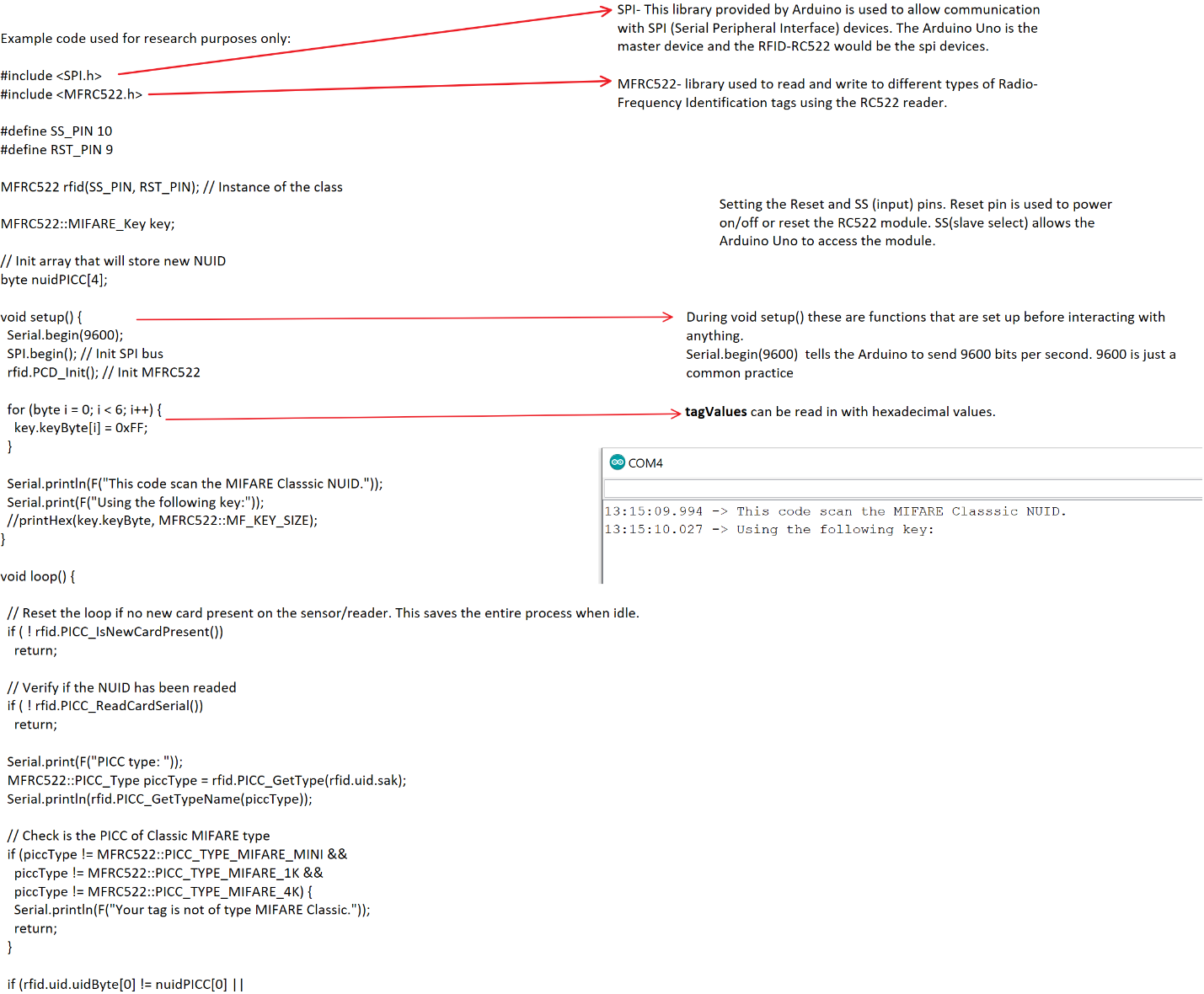
Ink Drawings
Ink Drawings
Student RFID Tag
tagValue

Ink Drawings
Ink Drawings
Ink Drawings
RFID READER
tagValue

tagList


The **tagValue** read by the RFID Reader then stored in a list called **tagList.**

The readers sole purpose is just getting values and storing them to be used by the **tagToID.exe**

  
    rfid.uid.uidByte[1] != nuidPICC[1] || 
    rfid.uid.uidByte[2] != nuidPICC[2] || 
    rfid.uid.uidByte[3] != nuidPICC[3] ) {
    Serial.println(F("A new card has been detected."));

    // Store NUID into nuidPICC array
    for (byte i = 0; i < 4; i++) {
      nuidPICC[i] = rfid.uid.uidByte[i];
    }
   
    Serial.println(F("The NUID tag is:"));
    Serial.print(F("In dec: "));
    printDec(rfid.uid.uidByte, rfid.uid.size);
    Serial.println();
  }
  else Serial.println(F("Card read previously."));

  // Halt PICC
  rfid.PICC_HaltA();

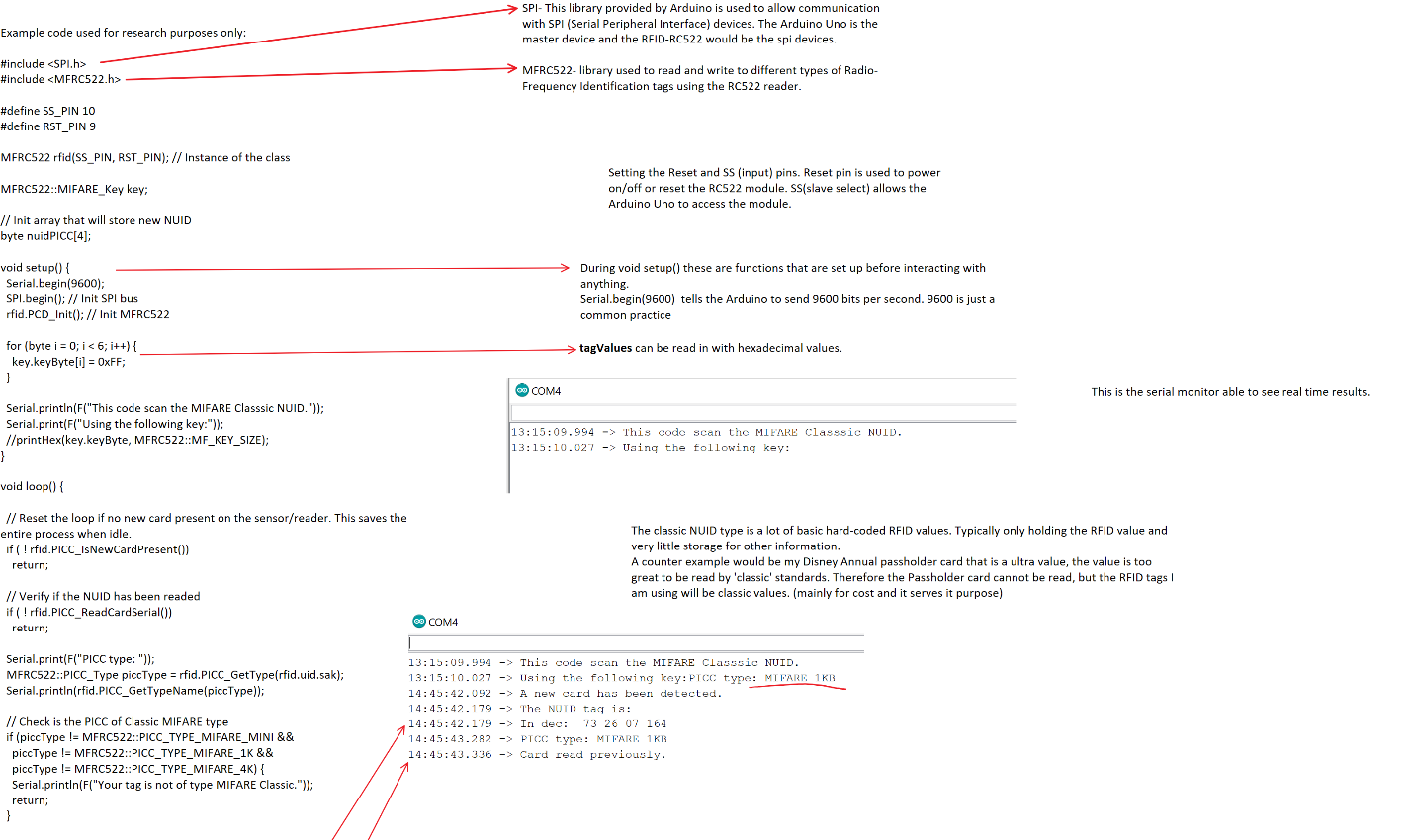
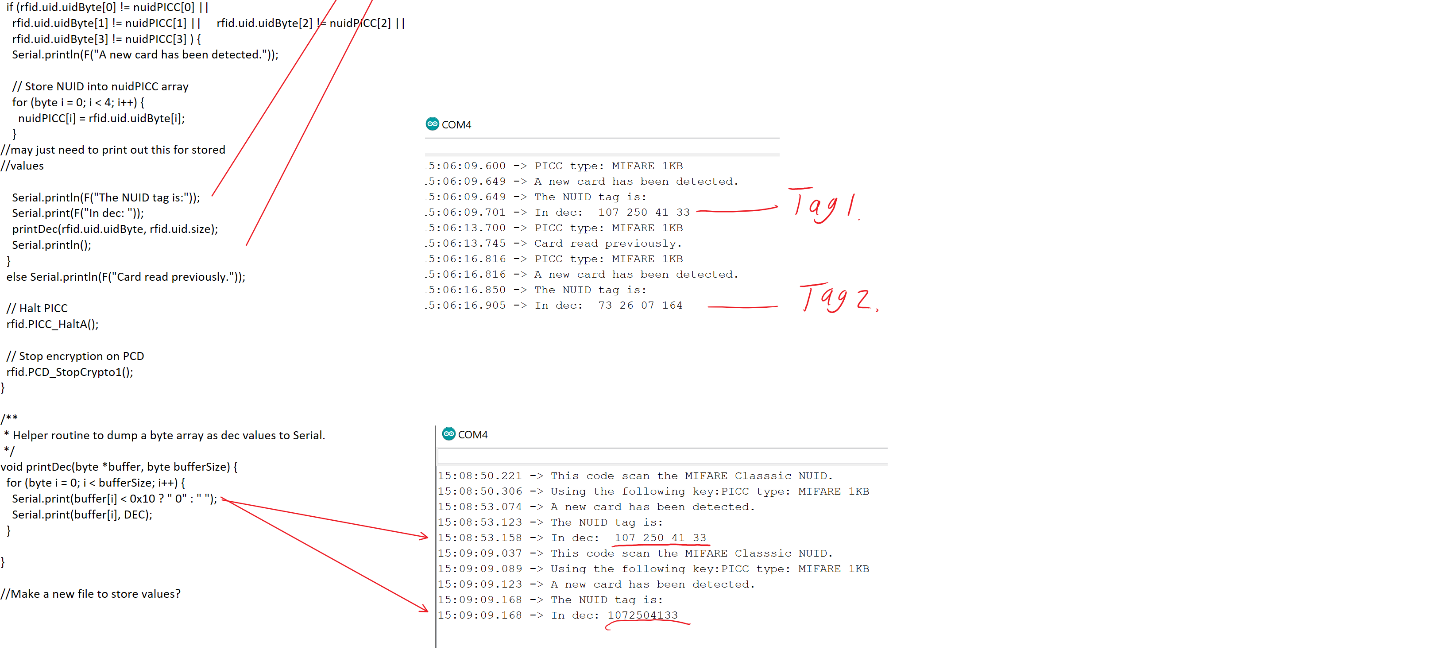
  // Stop encryption on PCD
  rfid.PCD_StopCrypto1();
}

/**
 * Helper routine to dump a byte array as dec values to Serial.
 */
void printDec(byte *buffer, byte bufferSize) {
  for (byte i = 0; i < bufferSize; i++) {
    Serial.print(buffer[i] < 0x10 ? " 0" : " ");
    Serial.print(buffer[i], DEC);
  }
  
}

//Make a new file to store values? 


# Design component 3: Compiler code

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  | Arduino IDE |  |
|  | The computer has 2 roles:   * Role 1:   + Compile code to operate the RFID * Role2:   + Make **tagToID.exe** that takes **tagList** values from RFID and ouputs a list of Student IDs to a csv file.         Role 1: compiler    Purpose:  This is code used to tell what the RFID what it needs to do. (code to operate the hardware). This will use Arduino IDE, given by the Arduino as a free download to run c++ code. The test code will be posted below showing how each function works and can be used for this project.    How:  Going to Arduino.cc you are able to download the Arudino IDE downloader and from there you can write programs to interact with your arduino uno board. Using the library's <SPI.h> and <MFRC522.h> will allow us to work directly with the RFID-RC522 module.  Machine generated alternative text: Show_to_Henderson_on_RFlD I Arduino 1.8.12 (Windows Store 1.8.33.0)  Eile Edit Sketch Tools Help  eo  Show to Henderson on RFID  r fid. u i d. uidByte [2]  r fid. u i d. uidByte [3]  Serial . print In (F ("A  nuidPICC [2]  nuidPICC [3] )  new card has been detected. )  // Store NUID into nuidPICC array  i  for (byte  O,  nuidPICC [i]  r fid. u i d. uidByte [i] ;  Serial . print In (F ("The NUID tag is: ,  Serial . print (F ("In dec  print Dec (r fid. u i d. uidByte, r fid. u i d. size)  Serial . print In ( ) ,  else Serial . print In (F ("Card read previously  // Halt PICC  • PICC HaltA() ;  r fid.  // Stop encryption on PCD  r fid. PCD StopCrypt01 ( ) ;  * Helper routine to dump a byte array as dec values to Serial .  void  print Dec (byte *buffer, byte  bufferSize) {  i  i < bufferSize; i++) {  for (byte  O,  // erial.print (buffer [i] < Ox 10 ? "  Serial . print (buffer  [i],  DEC) ;  // Make a new file to store values?  Done uploading.  Sketch uses 5608 bytes (17%) of program storage space. Maximum is 32256 bytes.  Global variables use 283 bytes (13%) of dynamic memory, leaving 1765 bytes for local variables .  81  ype here to search  Maximum is 2048 bytes.  O  x  Arduino Uno on COM4  3:18 PM  4/8/2020 |  |  |  |
|  |  |  |  |  |
|  |  | Machine generated alternative text: Show_to_Henderson_on_RFlD I Arduino 1.8.12 (Windows Store 1.8.33.0)  Eile Edit Sketch Tools Help  eo  Show to Henderson on RFID  r fid. u i d. uidByte [2]  r fid. u i d. uidByte [3]  Serial . print In (F ("A  nuidPICC [2]  nuidPICC [3] )  new card has been detected. )  // Store NUID into nuidPICC array  i  for (byte  O,  nuidPICC [i]  r fid. u i d. uidByte [i] ;  Serial . print In (F ("The NUID tag is: ,  Serial . print (F ("In dec  print Dec (r fid. u i d. uidByte, r fid. u i d. size)  Serial . print In ( ) ,  else Serial . print In (F ("Card read previously  // Halt PICC  • PICC HaltA() ;  r fid.  // Stop encryption on PCD  r fid. PCD StopCrypt01 ( ) ;  * Helper routine to dump a byte array as dec values to Serial .  void  print Dec (byte *buffer, byte  bufferSize) {  i  i < bufferSize; i++) {  for (byte  O,  // erial.print (buffer [i] < Ox 10 ? "  Serial . print (buffer  [i],  DEC) ;  // Make a new file to store values?  Done uploading.  Sketch uses 5608 bytes (17%) of program storage space. Maximum is 32256 bytes.  Global variables use 283 bytes (13%) of dynamic memory, leaving 1765 bytes for local variables .  81  ype here to search  Maximum is 2048 bytes.  O  x  Arduino Uno on COM4  3:18 PM  4/8/2020 | | |
|  |  |  |

Ink Drawings
Ink Drawings
Student RFID Tag
tagValue

Ink Drawings
Ink Drawings
Ink Drawings
RFID READER
tagValue

tagList
Ink Drawings
Ink Drawings
RFID C++

Read RFID tags(tagValue)
Should be able to: 
Store RFID tags in a list (tagList)

Red light is defult(nothing is being read)

Green light means card is found

Ink Drawings
Ink Drawings
C++ Code


# Design Component 4: tagToID.exe

Purpose: This is to take the list of tags from the RFID **(tagList)**and use those values. This will then sort through the list values and compare then to a excel spreadsheet/database that is already holding: Student\_Name, Student\_ID, Student\_Tag. The values from **tagList** should be able to sort through and once it finds the Student\_Tag that matches with the **tagValue** it should output the Student\_name and Student\_ID onto a CSV file to be saved in the event records.

How:

The Code for this **tagToID.exe** is going to be written in c++ using Visual Studio Code. It will have functions to do the following:

* getTaglist- take the **tagList** values and stores them in a new list to be compared.

* findStudent- this will sort through each tag and find the student associated with it.
  + If found it will take the Student\_Name & Student\_ID and put it into a new document called Student.csv

* Student.csv- this will open a new file to put the student information in. This will be given to the person in charge of recording the event details. (events name, date… ect)

\*\*\*If a student database is being used instead of a excel spreadsheet then additional research will be needed to update any process after this point.

# Future Research Efforts

Upon presenting my project I plan on having my project be able to read RFID cards that represent any student, faculty, and staff member at Charleston Southern University while providing a faster, more reliable way of scanning ID cards for CSU events (Chapel, student events, sports games). This project will yield out a document that can be used by office staff to show attendance at any event that requires a ID scan or counts towards chapel credit. This will not contain information, or be able to replace how to pay for things at Charleston Southern University (Dinning dollars, java city, ect…)

If I were to continue to add on to this project I would make RFID sensors that could allow certain students into designated areas. This already exists but the difference would be that the RFID tag would also be the students ID. The current situation is that the RFID cards are separate and students would have to carry multiple cards rather than just one that has permissions to multiple locks.

Another idea is that there is usually 1kb storage on the most simplest of cards so putting secret information on there for use of teachers or advisers would be interesting.

Could also extend this to be a attendance system for teacher that require attendance but