Application Note: RFID
Lauren Poole
ECE 480 Design Team 7
Executive Summary:

Radio Frequency Identification (RFID) is an affordable technology that can be used for applications such as security, tracking, and access control [1]. This Application Note will detail the required steps to program a RFID Card Reader for access control in an ArcelorMittal Plant. This Application Note will show and describe the programming needed to successfully identify the unique digital ID of a RFID tag and either grant or deny it access. These steps include the use and programming of a microcontroller.

![RFID card Reader - Serial](image)

Keywords:
- RFID
- Passive RFID tag
- Microcontroller
- Database

Background:
A RFID, radio-frequency identification, uses electromagnetic fields to transfer data. RFID is not a single product but rather a system, which is composed of: a RFID tag (transponder), reader (transceiver) and back-end application system (or database), which require the support of a computer network [2]. For this system the transponder is a passive RFID tag. Passive tags are cheaper, lighter, and smaller than the other tag options. That is because unlike other RFID tag types, passive tags do not require batteries. Passive tags use radio energy transmitted by the reader as a power source. Since the RFID reader powers the RFID tag the tag must be within 2 to 5 inches of the RFID reader in order to be read. It is also because the tag is passive that our RFID reader must be active. The RFID reader not only communicates with the RFID tag, but a microcontroller as well. The microcontroller in this application will serve as the middleman between the RFID reader and the database. The microcontroller notifies the reader if the serial identification code from the RFID tag has the clearance to gain access to the plant or not.

![Diagram of the RFID system](image)

**Figure 2. Illustration of the RFID system**

**Programming of the Microcontroller and RFID Reader:**

Prior to picking and purchasing an RFID a lot of research is needed. This research will assist in picking the best RFID brand for the specific project. Since an Arduino microcontroller had already been selected, a RFID reader that worked well with that particular microcontroller was ideal. Hence, the Parralax RFID card Reader-Serial was chosen with the compatible Parralax RFID Rectangle Tag.
From the research not only did we receive valuable information on the compatibility of the RFID reader and microcontroller, but a code to program the RFID for access control as well.

```c
#include <RFID.h>

int val = 0;
char code[10];
int byteRead = 0;

void setup()
{
  Serial.begin(2400); // RFID reader S0UT pin connected to Serial S0 pin at 2400bps
  pinMode(RFID_PIN, OUTPUT); // Set digital pin 4 as OUTPUT to connect it to the RFID /ENABLE pin
  digitalWrite(2, LOW); // Activate the RFID reader
}

void loop()
{
  if(Serial.available() > 0) // if data available from reader
    byteRead = 0;

  while(byteRead < 10) // read 10 digit code
  {
    if(Serial.available() > 0) // if header or stop bytes before the 10 digit reading
    {
      val = Serial.read(); // stop reading
      if((val == 10)|| (val == 13)) // if header or stop byte
        break;
    }

    code[byteRead] = val; // add the digit
    byteRead++; // ready to read next digit
  }

  if(byteRead == 10) // if 10 digit read is complete
  {
    Serial.print("TAG code is: "); // possibly a good TAG
    Serial.println(code); // print the TAG code
    byteRead = 0;
    digitalWrite(2, HIGH); // deactivate the RFID reader for a moment so it will not flood
    delay(1500); // wait for a bit
    digitalWrite(2, LOW); // Activate the RFID reader
  }
}
```

Figure 3. Code for using the Arduino with the Parallax RFID reader

Although this code alone did not give the desired results, it was a good starting point. This code allowed the RFID reader to recognize the presence of a RFID tag. However, it did not compare it to any database system to figure out if the tag was approved.
In order to add this function first, the code was reviewed and ran in order to understand how the code worked and what it instructed the microcontroller to do. Afterward the code was manipulated in the following ways:

```c
//RFID Settings
#include <SoftwareSerial.h>
//NewSoftwareSerial needs pin in and out, but RFID Reader only has IN
int pinMode = 12;
int pinMode = 8;
int pinMode = 10;
SoftwareSerial RFIDSerial(portRFID, pinRFID); //These are used for serial communication
int va = 0;
char rText[300];
char rText[300];
char rText[300];
char Denied = "00006E2C311";

//Servo settings
#include <Servo.h>
Servo servos;
int pinServoStart = 9;
int pinServoStop = 7;
int servolockRotation = 150;
int servolockRotation = 30;
int unlockTime = 3000;
int pin = 13;
void emptySerialBuffers()
{

//This emplis the serial buffer to prevent repeat reads
while ([0])
{
    if ([0])
    {
        while ([0])
        {
            Serial.read();
        }
        while ([0])
        {
            Serial.read();
        }
    }
    Serial.begin(9600);
//Set up RFID Reader
RFIDSerial.begin(baudRateRFID);
pinMode(portRFID, OUTPU);
digitalWrite(pinRFID, LOW);
//Set up servo
servo.attach(pinServoStart);
//Set up indication LED
pinMode(pin, OUTPUT);
digitalWrite(pin, LOW);

void loop()
{
    //Feed serial from RFID Reader
    val = rFIDSerial.read();
    //Got sign from RFID Reader
    if (val == 10)
    {

bytesfeed = 0;
while (bytesfeed < 10)
{
    if (bytesfeed > 10)
    {
        val = rFIDSerial.read();
        //Line ending mean end of message
        if (val == 10) || (val == 13)
        {
            break;
        }
    }
    bytesfeed = val;
}

if (bytesfeed == 10)
{
    //Send RFID tag to computer
    Serial.print(val);
    digitalWrite(portRFID, HIGH);
    digitalWrite(portRFID, LOW);
    Serial.write(portRFID, HIGH);
    Serial.write(portRFID, LOW);
    if (/*val == 10 || (stricmp("00006E2C311"), 0) */ endnanfoE2C311) == 0)
    {
        //Access Granted
        digitalWrite(portLED, HIGH);
        digitalWrite(portLED, LOW);
        //Set up servo
        servo.attach(portServoStart);
        //Set up indication LED
        pinMode(portLED, OUTPUT);
digitalWrite(portLED, LOW);
        //Feed serial from RFID Reader
    }
}

Figure 4. Modified code for using the Arduino with the Parallax RFID reader
```
Taking the general layout of the original provided loop we were able to incorporate further coding to assist in the transferring of information from the serial code to the microcontroller, to the database, back to the microcontroller.

**Results:**

At this point in our design we are still currently debugging the system.

**Conclusion:**

In conclusion, this application note demonstrates how the RFID is programmed for access control. RFIDs are a cheap and are easy to integrate into a system. That being said, it is important to understanding what exactly you want your RFID system to do and research the various components that need to be integrated in order to form a solution. Through this research you may get lucky and find tutorials on how to accomplish exactly what you are trying to do, or at least find a good place to start.
References: