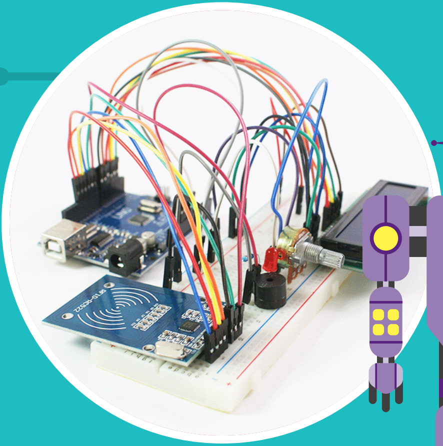
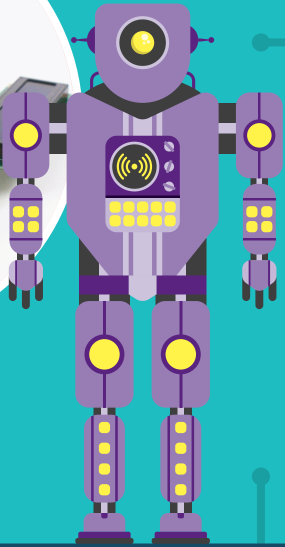
RFID READER

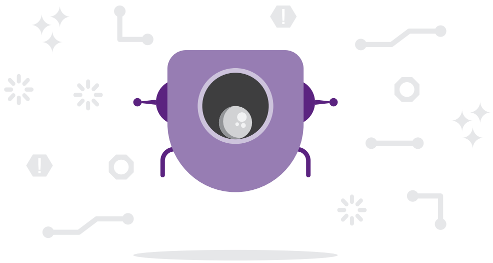




Creation Crate
BUILDING THE MAKERS OF TOMORROW

INDEX

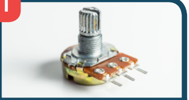
PART LIST	1
WHAT IS A RADIO FREQUENCY ID READER?	3
HARDWARE	5
LIBRARIES	7
PROGRAM	9
WHAT YOU SHOULD SEE	18
EXERCISES	19
MONTHLY CHALLENGE	20
SNEAK PEEK	21

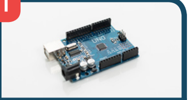


PART LIST

X1

16X2 LCD

X1

10K POTENTIOMETER

X1

UNO R3

X1

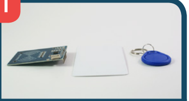
BUZZER

X2

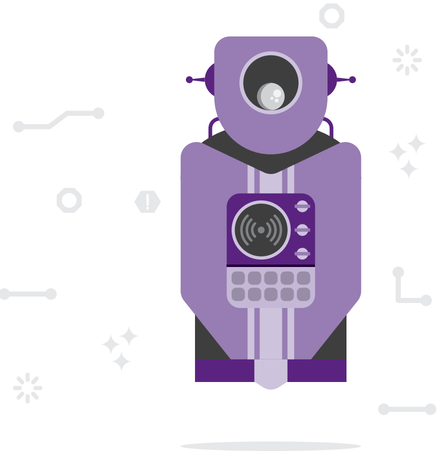
220 OHM RESISTOR

X1

LED

X1

RFID KIT

- RFID READER
- MASTER PROGRAMMING CARD
- TOKEN



INTRODUCTION

WELCOME TO MONTH 17!

WHAT ARE WE CREATING?

Okay, so far you've built a Lock Box, Infrared Security System, and Laser Tripwire. If we didn't know you better we'd think you were a bit paranoid or something. We figure there have to be a few people you would like to see from time-to-time, so this month you will build a Radio Frequency Identification (RFID) reader. That way, the few people you trust can get by that top-notch security system you've rigged up!

PROJECT AND LEARNING OBJECTIVES

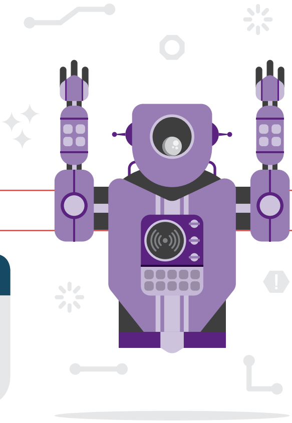
You will learn how to write a program that will allow you to read a master RFID key card and authorize/deactivate a card/token.

ELECTRONICS

1. What is RFID?
2. PICC - Proximity Integrated Circuit Card
3. Integrating multiple Creation Crate projects

PROGRAMMING

1. EEPROM Read/Write - Electrically Erasable Programmable Read-Only Memory
2. SPI - Serial Peripheral Interface



SUPPORT PAGE:

LINK

www.creationcrate.com/month-17

SSKR45

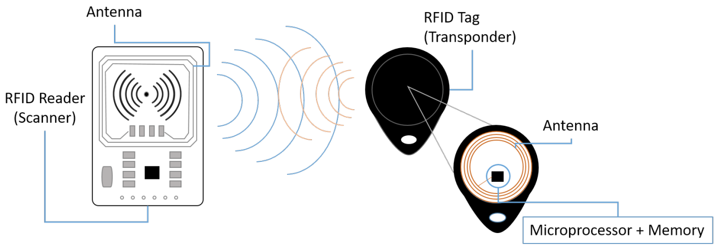
Still Need Help?
Go **www.creationcrate.com** and use our contact page!

WHAT IS A RADIO FREQUENCY ID READER?

Radio-frequency identification (***RFID***) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically stored information. Passive tags collect energy from a nearby RFID reader's "interrogating" radio waves.

RFID tags are used in many industries, for example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line; RFID-tagged pharmaceuticals can be tracked through warehouses; and implanting RFID microchips in livestock and pets allows for positive identification of animals.

RFID tags are small, easy to use and can be attached to cash, clothing, and possessions, or implanted in animals and people.



Radio waves are constantly being sent out from the scanner. When a tag is close enough to the reader, the radio waves will power the tag and activate it, allowing it to send data back to the reader via its own radio waves.

NEW TERMS USED IN THIS PROJECT:

EEPROM – Stand for Electrically Erasable Programmable Read-only Memory and is a type of memory used in computers and other electronic devices to store relatively small amounts of data, but allowing individual bytes to be erased and reprogrammed.

SPI – Stands for Serial Peripheral Interface bus - is a synchronous (meaning the data is kept identical by using a clock signal) serial communication (meaning that the data is delivered and received one byte at a time) interface specification used for short distance communication. This was developed by Motorola in the 80s and remains the standard.

PICC – Pronounced "pick – c" – PIC stands for Peripheral Interface Controller and the C is for the C programming language (in this case, used to program the EEPROM)

UID – Unique Identifier, as in each RFID tag has its own UID so we can tell them apart.

HEX – Short for Hexadecimal, which is a base 16 number system, using 16 symbols (0-9 and A-F) to represent a much larger string of numbers. Each hexadecimal digit represents four **binary digits** (bits), it allows a more human-friendly representation of binary-coded values. One hexadecimal digit represents a **nibble** (4 bits), which is half of an **octet** or **byte** (8 bits).

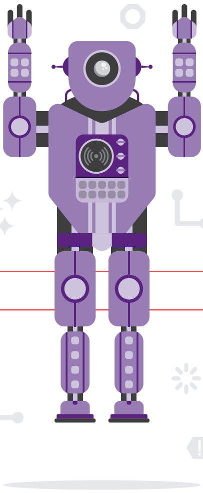
Example:

The number **0** represents **0000**

The number **5** represents **0101**

The letter **F** represents **1111**

Therefore, sending **5F005F** would be **0101 1111 0000 0000 0101 0101** in binary



NEW COMMANDS AND FUNCTIONS FOR THIS PROJECT

Commands defined in the EEPROM library:

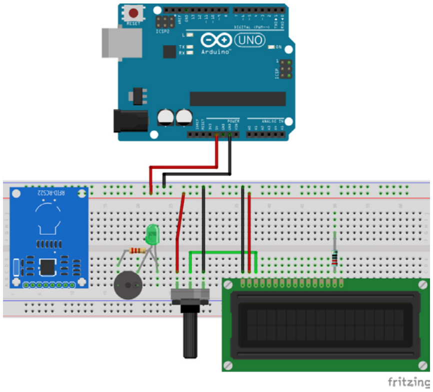
EEPROM.read – gathers the data stored in the EEPROM
EEPROM.write – sends data to the EEPROM

AntennaGain – This command will set the gain (or power intensity) of the antenna embedded into the scanner.

BUILDING THE ELECTRICAL CIRCUIT

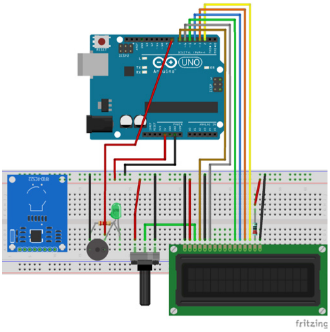
1) Let's begin by wiring the power to the breadboard and setting up the piezo buzzer, LED, and potentiometer. Remember, the potentiometer will be used to adjust the contrast of the LCD screen.

2) There are two 220 Ohm resistors, one for the LED and one for the LCD screen



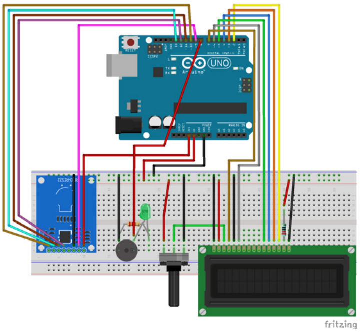
3) Connect input and ground to the buzzer and LED

4) Connect the LC Screen to the Arduino – this should be a familiar setup to what you have done in past projects.



5) Now let's connect the RFID Reader to the Arduino.

6) When connecting the RFID Reader, the jumper wires will actually be behind the module. They are shown on top of the module to make it easier to see where the wires should terminate.



That's it for the hardware!

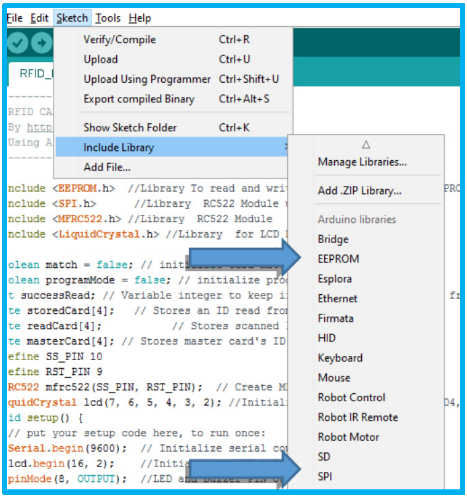
LIBRARIES

BEFORE WE BEGIN PROGRAMMING, WE WILL NEED THE FOLLOWING LIBRARIES:

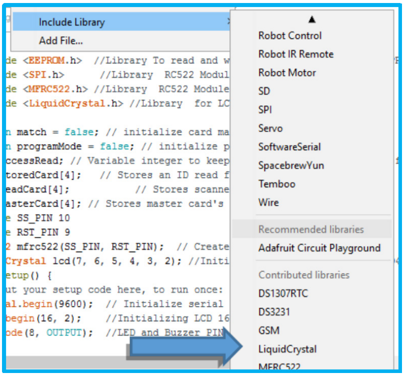
The following 3 libraries are usually loaded with the Arduino IDE software.

EEPROM.h - Read and write PICC's UIDs from/to EEPROM
SPI.h - SPI protocol
LiquidCrystal.h - LCD Display

You should see them here:



```
--------
RFID CA
By http
Using A
--------
nclude <EEPROM.h>  //Library To read and wri
nclude <SPI.h>      //Library  RC522 Module
nclude <MFRC522.h> //Library  RC522 Module
nclude <LiquidCrystal.h> //Library  for LCD
                                                    PRO
olean match = false; // init
olean programMode = false; // initialize pro
t successRead; // Variable integer to keep i                fr
te storedCard[4];   // Stores an ID read fro
te readCard[4];           // Stores scanned
te masterCard[4]; // Stores master card's ID
efine SS_PIN 10
efine RST_PIN 9
RC522 mfrc522(SS_PIN, RST_PIN);  // Create M
quidCrystal lcd(7, 6, 5, 4, 3, 2); //Initial                D4,
id setup() {
// put your setup code here, to run once:
Serial.begin(9600);  // Initialize serial co
lcd.begin(16, 2);    //Initia
pinMode(8, OUTPUT);  //LED a
```



```
de <EEPROM.h>  //Library To read and w
de <SPI.h>      //Library  RC522 Modul
de <MFRC522.h> //Library  RC522 Module
de <LiquidCrystal.h> //Library  for LC

n match = false; // initialize card ma
n programMode = false; // initialize p
ccessRead; // Variable integer to keep
toredCard[4];   // Stores an ID read f
eadCard[4];          // Stores scanne
asterCard[4]; // Stores master card's
e SS_PIN 10
e RST_PIN 9
2 mfrc522(SS_PIN, RST_PIN);  // Create
Crystal lcd(7, 6, 5, 4, 3, 2); //Initi
etup() {
ut your setup code here, to run once:
al.begin(9600);  // Initialize serial
begin(16, 2);    //Initializing LCD 16
ode(8, OUTPUT);  //LED and Buzzer PIN
```

This is a new library and will need to be installed.

MFRC522.h - RFID Reader - RC522 Module

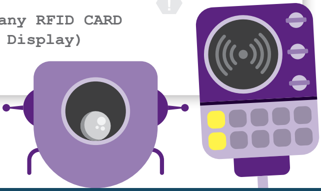
1. Download the Library from **www.mycreationcrate.com/month-17**
2. Open the Arduino IDE
3. Select Sketch → Include Library → Add .ZIP Library...
4. Go to the location you saved the file (usually this will be in your Downloads folder)
5. Select the file and click Open
6. The library is now installed and ready for use. You can confirm this when you begin writing your code. When you enter the command:

#include <MRFC522.h>   the text should change to green and orange

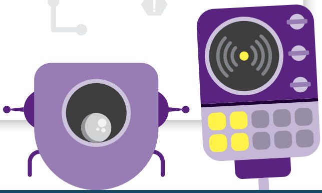
#include <MRFC522.h>

PROGRAM

```
//Month 17: RFID

#include <EEPROM.h>  //Library To read and write PICC's UIDs from/to EEPROM
#include <SPI.h>       //Library  RC522 Module uses SPI protocol
#include <MFRC522.h> //Library  RC522 Module
#include <LiquidCrystal.h> //Library  for LCD Display

boolean match = false; // initialize card match to false
boolean programMode = false; // initialize programming mode to false
int successRead; // Variable integer to keep if we have Successful Read
from Reader
byte storedCard[4];   // Stores an ID read from EEPROM
byte readCard[4];           // Stores scanned ID read from RFID Module
byte masterCard[4]; // Stores master card's ID read from EEPROM
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance.
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); //Initializing LCD PINS as
(RS,EN,D4,D5,D6,D7)
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);  // Initialize serial communications with PC
  lcd.begin(16, 2);    //Initializing LCD 16x2
  pinMode(8, OUTPUT);  //LED and Buzzer PIN OUT
  SPI.begin();           // MFRC522 Hardware uses SPI protocol
  mfrc522.PCD_Init();    // Initialize MFRC522 Hardware
  mfrc522.PCD_SetAntennaGain(mfrc522.RxGain_max);
  if (EEPROM.read(1) != 1) {  // Look EEPROM if Master Card defined,
EEPROM address 1 holds if defined
    Serial.println("No Master Card Defined"); //When no Master Card in
Your EEROM (Serial Display)
    Serial.println("Scan A PICC to Define as Master Card");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.println("SCAN MASTER   "); //When no Master Card in Your EEROM
(LCD Display)
    lcd.setCursor(0, 1);
    lcd.println("SCAN A KEYCARD"); //Scan any RFID CARD
to set Your Master Card in Your EEROM (LCD Display)
    delay(1500);
    do {
```

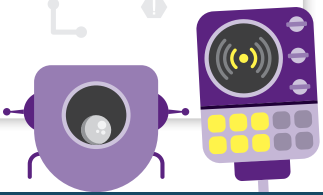


```
      successRead = getID(); // sets successRead to 1 when we get read
from reader otherwise 0
    }
    while (!successRead); //the program will not go further while you
not get a successful read
    for ( int j = 0; j < 4; j++ ) { // Loop 4 times
      EEPROM.write( 2 + j, readCard[j] ); // Write scanned PICC's UID to
EEPROM, start from address 3
    }
    EEPROM.write(1, 1); //Write to EEPROM we defined Master Card.
    Serial.println("Master Card Defined");

  }
  Serial.println("Master Card's UID");
  for ( int i = 0; i < 4; i++ ) {     // Read Master Card's UID from
EEPROM
    masterCard[i] = EEPROM.read(2 + i); // Write it to masterCard
    Serial.print(masterCard[i], HEX); //Master Card only view in serial
     Serial.println("Waiting PICCs to be scanned");
  }
  //WAITING TO SCAN THE RFID CARDS:
  Serial.println("");
  Serial.println("Waiting PICCs to be scanned");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("WAITING:        ");
  lcd.setCursor(0, 1);
  lcd.println("SCAN CARD/TOKEN ");
  delay(1500);
}
void loop() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("      SWIPE");
  lcd.setCursor(0, 1);
  lcd.print("    CARD/TOKEN");

 /*
 if (digitalRead(BUTTON) == HIGH);
//To Delete the EEROM USE the below command just run it
  {
  // for (int i = 0 ; i < EEPROM.length() ; i++) {
  // EEPROM.write(i, 0);
  // }
  // }                                     */
  do {
    successRead = getID(); // sets successRead to 1 when we get read
from reader otherwise 0
    if (programMode) {
      // Program Mode cycles through RGB waiting to read a new card
    }
    else {
   }}
  while (!successRead); //the program will not go further while you not
get a successful read
  if (programMode) {
    if ( isMaster(readCard) ) {  //If master card scanned again exit
program mode
      Serial.println("This is Master Card");
      Serial.println("Exiting Program Mode");
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(" EXITING FROM");
      lcd.setCursor(0, 1);
      lcd.print("PROGRAMING MODE");
      delay(2000);
      programMode = false;
      return;
    }
    else {
      if ( findID(readCard) ) { //If scanned card is known delete it
        Serial.println("I know this PICC, removing from DB");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(" ACTIVE TOKEN:");
        lcd.setCursor(0, 1);
        lcd.print("REVOKING ACCESS");
        delay(5000);
        deleteID(readCard);
        Serial.println("-----------------------------");
      }
      else {                   // If scanned card is not known add it
        Serial.println("I do not know this PICC, adding to DB...");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Card no:");
        lcd.setCursor(0, 1);
        lcd.print(readCard[0], HEX);
        lcd.print(readCard[1], HEX);
        lcd.print(readCard[2], HEX);
        lcd.print(readCard[3], HEX);
        lcd.print(readCard[4], HEX);
        delay(4000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("  NEW TOKEN:");
        lcd.setCursor(0, 1);
        lcd.print("GRANTING ACCESS");
        delay(5000);
        writeID(readCard);
        Serial.println("-----------------------------");
      }} }
  else {
    if ( isMaster(readCard) ) {  // If scanned card's ID matches Master
Card's ID enter program mode
      programMode = true;
      Serial.println("Welcome to Mastercard Mode");
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("ID: MASTER CARD");
      lcd.setCursor(0, 1);
      lcd.print("PROGRAMING MODE");
      delay(3000);
      int count = EEPROM.read(0); // Read the first Byte of EEPROM that
      Serial.print("I have ");    // stores the number of ID's in EEPROM
      Serial.print(count);
      Serial.print(" record(s) on EEPROM");
      Serial.println("");
      Serial.println("Scan a PICC to ADD or REMOVE");
      Serial.println("-----------------------------");
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("SCAN CARD/TOKEN");
      lcd.setCursor(0, 1);
      lcd.print("TO ADD / REMOVE");
      delay(2500);
    }
    else {
```



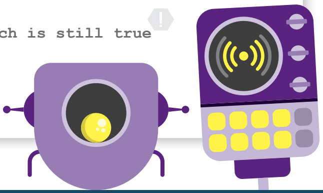
PROGRAM (CONT'D)

```
      if ( findID(readCard) ) {         // If not, see if the card is in
the EEPROM
        Serial.println("ACCESS GRANTED");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(" ID AUTHORIZED");
        lcd.setCursor(0, 1);
        lcd.print(" ACCESS GRANTED");
        for (int abcd = 0; abcd < 2; abcd++)
        {
          digitalWrite(8, HIGH);
          delay(200);
          digitalWrite(8, LOW);
          delay(100);
        }
        delay(900);
        lcd.clear();
      }

      else {         // If not, show that the ID was not valid
        Serial.println("Access Denied");
        for (int abcd = 0; abcd < 4; abcd++)
        {
          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("   UNKNOWN ID");
          lcd.setCursor(0, 1);
          lcd.print("  ACCESS DENIED");
          digitalWrite(8, HIGH);
          delay(600);
          digitalWrite(8, LOW);
          lcd.clear();
          lcd.print("  UNAUTHORIZED");
          lcd.setCursor(0, 1);
          lcd.print("  PERSONNEL");
          delay(900);
        }
        lcd.clear();
      }}}}

int getID() {
  // Getting ready for Reading PICCs
```

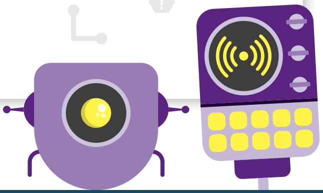


PROGRAM (CONT'D)

```
  if ( ! mfrc522.PICC_IsNewCardPresent()) { //If a new PICC placed to
RFID reader continue
    return 0;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) { //Since a PICC placed get
Serial and continue
    return 0;
  }

  Serial.println("Scanning PICC's UID.........");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("    SCANNING");
  lcd.setCursor(0, 1);
  lcd.print("  CARD / TOKEN");
  delay(2000);
  for (int i = 0; i < 4; i++) {  //
    readCard[i] = mfrc522.uid.uidByte[i];
    Serial.print(readCard[i], HEX);
  }
  Serial.println("");
  mfrc522.PICC_HaltA(); // Stop reading
  return 1;
}
boolean isMaster( byte test[] ) {
  if ( checkTwo( test, masterCard ) )
    return true;    else
    return false;
}

boolean checkTwo ( byte a[], byte b[] ) {
  if ( a[0] != NULL ) // Make sure there is something in the array first
    match = true; // Assume they match at first
  for ( int k = 0; k < 4; k++ ) { // Loop 4 times
    if ( a[k] != b[k] ) // IF a != b then set match = false, one fails,
all fail
      match = false;
  }
  if ( match ) { // Check to see if if match is still true
    return true; // Return true
  }
  else  {
    return false; // Return false
```



```
  }}
boolean findID( byte find[] ) {
  int count = EEPROM.read(0); // Read the first Byte of EEPROM that
  for ( int i = 1; i <= count; i++ ) {  // Loop once for each EEPROM entry
    readID(i); // Read an ID from EEPROM, it is stored in storedCard[4]
    if ( checkTwo( find, storedCard ) ) { // Check to see if the
storedCard read from EEPROM
      return true;
      break; // Stop looking we found it
    }
    else {  // If not, return false
    }}
  return false;
}
void readID( int number ) {
  int start = (number * 4 ) + 2; // Figure out starting position
  for ( int i = 0; i < 4; i++ ) { // Loop 4 times to get the 4 Bytes
    storedCard[i] = EEPROM.read(start + i); // Assign values read from
EEPROM to array
  }
}
void deleteID( byte a[] ) {
  if ( !findID( a ) ) { // Before we delete from the EEPROM, check to see
if we have this card!
    failedWrite(); // If not
  }
  else {
    int num = EEPROM.read(0); // Get the numer of used spaces, position
0 stores the number of ID cards
    int slot; // Figure out the slot number of the card
    int start;// = ( num * 4 ) + 6; // Figure out where the next slot starts
    int looping; // The number of times the loop repeats
    int j;
    int count = EEPROM.read(0); // Read the first Byte of EEPROM that
stores number of cards
    slot = findIDSLOT( a ); //Figure out the slot number of the card to
delete
    start = (slot * 4) + 2;
    looping = ((num - slot) * 4);
    num--; // Decrement the counter by one
    EEPROM.write( 0, num ); // Write the new count to the counter
    for ( j = 0; j < looping; j++ ) { // Loop the card shift times
      EEPROM.write( start + j, EEPROM.read(start + 4 + j)); // Shift the
array values to 4 places earlier in the EEPROM
    }
    for ( int k = 0; k < 4; k++ ) { //Shifting loop
      EEPROM.write( start + j + k, 0);
    }
    successDelete();
  }}
  //For Failed to add the card:
void failedWrite() {

  Serial.println("something wrong with Card");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" PROBLEM ADDING");
  lcd.setCursor(0, 1);
  lcd.print("  TRY AGAIN");
  delay(2000);
}
//For Sucessfully Deleted:
void successDelete() {
  Serial.println("Succesfully removed");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("  SUCCESFULLY");
  lcd.setCursor(0, 1);
  lcd.print("    REMOVED");
  delay(2000);
}
int findIDSLOT( byte find[] ) {
  int count = EEPROM.read(0); // Read the first Byte of EEPROM that
  for ( int i = 1; i <= count; i++ ) { // Loop once for each EEPROM entry
    readID(i); // Read an ID from EEPROM, it is stored in storedCard[4]
    if ( checkTwo( find, storedCard ) ) { // Check to see if the
storedCard read from EEPROM
      // is the same as the find[] ID card passed
      return i; // The slot number of the card
      break; // Stop looking we found it
    }
  }
}
//For Sucessfully Added:
```



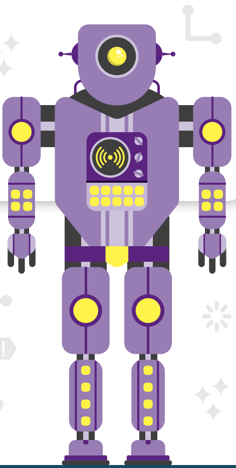
PROGRAM (CONT'D)

```
void successWrite() {

  Serial.println("Succesfully added");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("  SUCCESFULLY");
  lcd.setCursor(0, 1);
  lcd.print("    ADDED");
  delay(2000);
}
//For Adding card to EEROM:
void writeID( byte a[] ) {
  if ( !findID( a ) ) { // Before we write to the EEPROM, check to see if 
we have seen this card before!
    int num = EEPROM.read(0); // Get the numer of used spaces, position 
0 stores the number of ID cards
    int start = ( num * 4 ) + 6; // Figure out where the next slot starts
    num++; // Increment the counter by one
    EEPROM.write( 0, num ); // Write the new count to the counter
    for ( int j = 0; j < 4; j++ ) { // Loop 4 times
      EEPROM.write( start + j, a[j] ); // Write the array values to 
EEPROM in the right position
    }
    successWrite();
  }
  else {
    failedWrite();
  }
}

/ / Attribution: Non Stop Engineering
```

DOWNLOAD THE CODE AT:

www.creationcrate.com/month-17

SSKR45



WHAT YOU SHOULD SEE

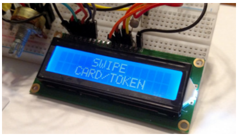
When you power up the Uno R3, the LCD screen should say “SWIPE CARD/TOKEN.”

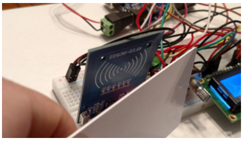
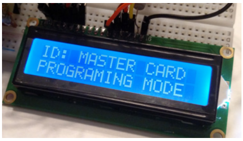
When you hold the white Master Card in front of the reader it will say, “SCANNING CARD/TOKEN” and then “ID: MASTER CARD PROGRAMMING MODE,” then “SCAN CARD/TOKEN TO ADD/REMOVE.”

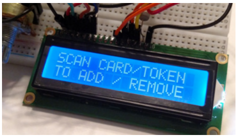
Hold the Key Fob Token in front of the reader and it will display, “Card no: __________” and then “NEW TOKEN: GRANTING ACCESS.”

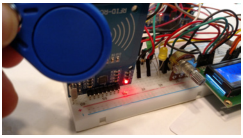
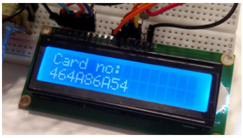
If the card number is already in the database, it will read, “ACTIVE TOKEN: REVOKING ACCESS.”

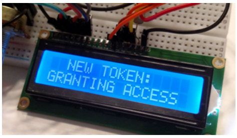
When you are done adding/removing card access, hold the Master Card in front of the reader again and you should see, “EXITING PROGRAMMING MODE.”

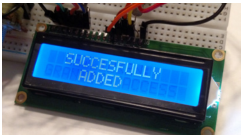


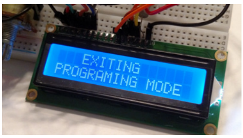





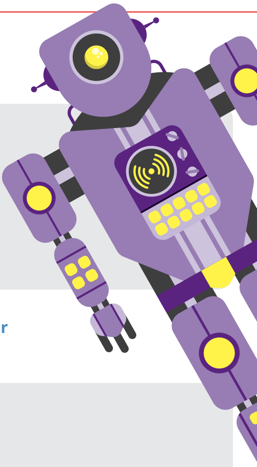






EXERCISES

SOLVE THESE PROBLEMS AND WRITE THE ANSWERS BELOW.



1. How would you change the number of times that the LED and Message flash when access is denied to just 3 times?

Answer:

2. How would you make the "Access Granted" message appear for a longer period of time on the LCD screen?

Answer:

3. How would you increase the number of beeps you hear to 3 when access is granted, without also changing the number of times the message flashes on the screen?

Answer:

MONTHLY CHALLENGE

1. Add a red LED and change the program so that it flashes when access is denied instead of the other LED.

2a.Use parts and integrate code from Month 13, Infrared Security System, or Month 15, Laser Trip Wire, and use the badge to disarm the system.

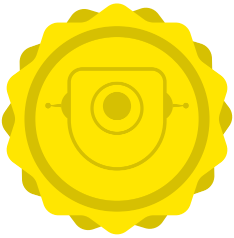
2b.Can you also rearm the system when you leave?

SUPPORT PAGE:

www.creationcrate.com/month-17

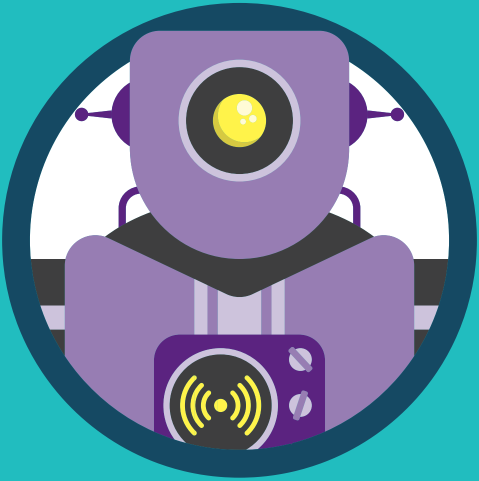
SSKR45

Still need help?
Go to **www.creationcrate.com** and use our contact page!



HERE'S A SNEAK PEAK AT NEXT MONTH'S PROJECT!

Can you guess what next month's project is?



Creation Crate

BUILDING THE MAKERS OF TOMORROW