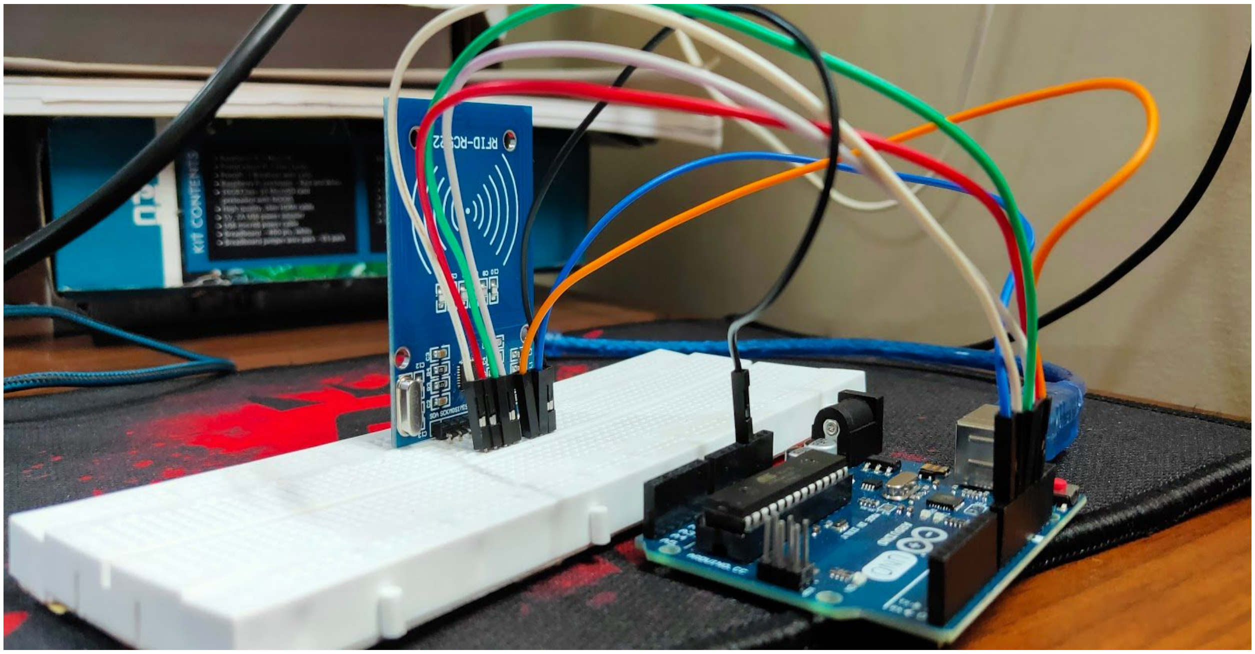
Lock/Unlock Your Computer Using Arduino

Guided By : Prof. Ch. Pradeep Reddy



Team :

1. G V Datta Adithya	- 19BCE7119
2. Abhiram Kalyan	- 19BCD7118
3. S. Vijay Balaji	- 19BCE7571
4. M. Sri Santhosh Reddy	- 19BCN7091
5. K. Kaushik	- 19BCD7137
6. Ch. Sabareesh Reddy	- 19BCE7210

Abstract

All of us use our computers on a daily basis and sleep it as we have a need to resume our work at a later time.

Knowing how hard it is to constantly type in the password without someone peeking in, or making mistakes while typing it in only to get locked out for a while, and leaving your desk and people snooping through your files, doesn't make it better.

Hence, the easiest and safest way to unlock your computer would be using Arduino, and an RFID card.

Tracking your device once the module is integrated within the hardware is also a feature which you can enable using this.

It is not only easy to make but also very helpful in one's day to day life.

A creative device that would help everyone in the long run.

Afterall, it's the small things that matter.

Introduction

Cyber-Safety is vital.

Passwords are a big part in Cyber-Safety and there are so many people who have begun to use computers on a daily basis since the 90's.
People have begun to utilise computers a lot more than the past few years and the time spent on the computer increases exponentially based on recent surveys and studies conducted.

With the rise in usage of computers, there is also a raise in the question of safety in this aspect.
There is a large number of people who suffer from identity theft online, on a daily basis.

And, a large number of these thefts rise from a single change in the equation.
Getting access to one's password.

According to a new report, nearly 3 out of 4 consumers use duplicate passwords, many of which have not been changed in five years or more. Unsurprisingly, about 40 percent of those surveyed say they had "a security incident" in the past year, meaning they had an account hacked, password stolen, or were given notice that their personal information had been compromised.

A large number of us, utilise the same password for every single account and while we all know that this is not a good practice, we can't help ourselves from using the same password everywhere.

We are afraid of losing our password, and losing our computer password is the worst, as there are a number of steps that we have to go through in order to retrieve it.

This is where the RFID Scanner comes into play.
Having the RFID Scanner check your identity using an RFID tag is what makes logging in, extremely easy.

As each RFID tag transmits different signals to the receiver, we do not have to worry about there being an exact duplicate of your RFID tag.

This is an instrument which provides us with a sense of security, and a physical feel to having your password with you, and not drawing a blank in front of the login screen of your computer.

Background

There is a large population of people who suffer from identity theft, and information breaching, and lack network security due to other people snooping around in a person’s private device.

This is why we require tough passwords, like 2#Js(3}2QuaE9, but, logically, it would be a massive task to remember passwords like these without writing them down somewhere.

Problem Definition

Remembering a complex password would take up a lot of time, and most passwords have some or the other crackable pattern and would not depend on randomness.

People could snoop while you type a password, and find out what your credentials are through finger-tracking, which could put a person's personal information out, and give access to files that are not supposed to be privy to other people's eyes.

Having a simpler password, such as a pin, would be easy to crack and something that people could understand after noticing it a few times.

Having a complex password, could avoid this, but would have the user retrying the password almost all the time.

Both of these methods are relatively insecure and would, arguably, frustrate the user on the computer.

The RFID Scanner seeks to rectify this issue and have the users spend a lot less time retrying their password or losing their credentials to someone who was able to access their system.

You can integrate randomness into your passwords and have them be uncrackable and extremely secure.

Utilising our product, one could have extremely complex passwords and still be secure as they don't necessarily have to remember them, but keep it in a place which would be relatively safe.

Problems of Existing Systems

➔ Simple, Easy to crack passcodes.

➔ One single key for all the doors.

➔ It is a slow and taxing process to enter large passcodes, especially for people who don’t use computers on a daily basis.

Objective

The main objective of this project is the utilization of an Arduino Board to lock/unlock your computer.

We have done this project in order to reduce the time it takes to manually type a password, and to allow users to use extremely complex passwords, which accordingly makes the password extremely strong and hard to copy.

This way one can save time, and energy on logging into their account on the computer.

Methodology and Procedure

We have created a device which can unlock/lock your computer when a specific RFID tag is in close proximity to the sensor.

An RFID tags receive energy from the radio waves generated by the reader. When the tag receives the transmission from the reader/antenna, the energy runs through the internal antenna to the tag's chip. The energy activates the chip, which modulates the energy with the desired information, and then transmits a signal back towards the reader, which is the RFID Scanner in this project.

Based on this working principle, we have designed a device that works and can lock/unlock your PC.

Equipment used in this project :

- Arduino Uno
- RFID Scanner
- RFID tags
- Jumper wires
- Breadboard
- Cable from laptop to Arduino

Keyboard Support for the Arduino Board

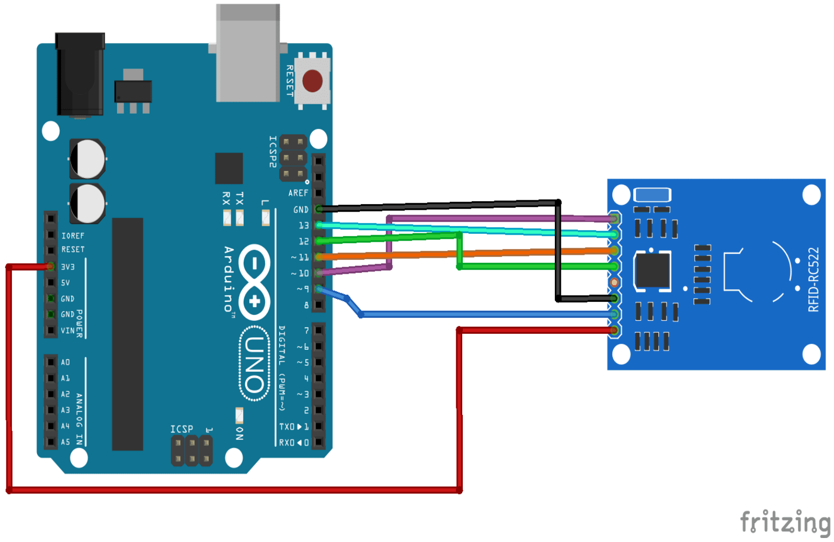
Firstly, we utilised the Arduino application to connect the Arduino board to the computer and upload the code for the execution of the program.
We had to install an Atmel software, in order to convert the arduino board into a keyboard so that we could apply a .hex file to the keyboard to configure keys to the board.
After configuring the arduino board, to accept these .hex codes, we convert the arduino board from a keyboard back into an arduino board.
Now, we can utilise keystrokes in the program, in order to have the arduino board print the symbols for the password.

Locking the computer doesn't require this principle now, as we have already converted the board to accept the keyboard strokes for the password.

The code utilised for this program is given down below in the appendix.

Basic Circuit Diagram

The connections for the circuit are as follows,



Pin	Wiring to Arduino UNO
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	unconnected
GND	GND
RST	Digital 9
3.3 V	3.3V

VCC supplies power for the module. This can be anywhere from 2.5 to 3.3 volts. You can connect it to 3.3V output from your Arduino. Remember connecting it to 5V pin will likely destroy your module!

RST is an input for Reset and power-down. When this pin goes low, hard power-down is enabled. This turns off all internal current sinks including the oscillator and the input pins are disconnected from the outside world. On the rising edge, the module is reset.

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

IRQ is an interrupt pin that can alert the microcontroller when RFID tag comes into its vicinity.

MISO / SCL / Tx pin acts as Master-In-Slave-Out when SPI interface is enabled, acts as serial clock when I2C interface is enabled and acts as serial data output when UART interface is enabled.

MOSI (Master Out Slave In) is SPI input to the RC522 module.

SCK (Serial Clock) accepts clock pulses provided by the SPI bus Master i.e. Arduino.

SS / SDA (Serial Data) / Rx pin acts as Signal input when SPI interface is enabled, acts as serial data when I2C interface is enabled and acts as serial data input when UART interface is enabled. This pin is usually marked by encasing the pin in a square so it can be used as a reference for identifying the other pins.

Results and Discussions

Test successful in unlocking your PC, once the RFID tag is brought into close proximity to the RFID Scanner, the password is entered if the screen is locked.

If the computer is already unlocked, the approach of the RFID tag would result in the locking of the computer.

Conclusion

The entire device can be made extremely compact and had the potential to even be embedded into the armrest of a laptop.

This device has a lot of potential in companies which require employees to be using their ID in order to access the computer without needing any knowledge of the credentials required to access the online workspace of the computer.

On further development, there is potential for the device to feature the start-up of applications, which would enable the users to resume their workflow without any interruption and have a smooth transition into their workstation.

References

- https://www.instructables.com/id/Unlock-PC-With-RFID-and-Arduino-Uno/
- https://www.instructables.com/id/Windows-PC-LockUnlock-Using-RFID/
- http://playground.arduino.cc/
- https://www.instructables.com/id/Unlock-PC-With-RFID-and-Arduino-Uno/
- https://create.arduino.cc/projecthub/kksjunior/windows-pc-lock-unlock-using-rfid-5021a6

Appendix

The code that was utilised in order to make this project feasible is shown below,

```
#include <SPI.h>
#include <MFRC522.h>


#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.


uint8_t buf[8] = {
0 }; /* Keyboard report buffer */


int cardCount = 0;
void setup()
{
  Serial.begin(9600);
  randomSeed(analogRead(0));
  delay(200);
  SPI.begin();      // Initiate  SPI bus

  mfrc522.PCD_Init();   // Initiate MFRC522




}

void loop(){

  // Look for new cards

  if ( ! mfrc522.PICC_IsNewCardPresent())

  {

    return;

  }

  // Select one of the cards

  if ( ! mfrc522.PICC_ReadCardSerial())

  {

    return;

  }

  //Show UID on serial monitor

  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  content.toUpperCase();
  if (content.substring(1) == "E6 F4 94 F4" , "B3 C9 2C 83" , "F3 68 47 0C" , "43 55 51 0C" , "06 7D 4A F6" , "06 E8 5B F6" ) //change here the UID of the card/cards that you want to give access
  {
     delay(50);

buf[0] = 0;
buf[2] = 0xE3; // Win Key
buf[3] = 0x0F; // letter L
Serial.write(buf, 8);
releaseKey();
```

```
delay(200);


buf[0] = 0;

buf[2] = 0x04; // letter A

Serial.write(buf, 8);

releaseKey();


delay(50);


buf[0] = 0;

buf[2] = 0x05; // letter B

Serial.write(buf, 8);

releaseKey();


delay(50);


buf[0] = 0;

buf[2] = 0x06; // letter C
```

```
Serial.write(buf, 8);

releaseKey();


delay(50);


buf[0] = 0;

buf[2] = 0x07; // letter D

Serial.write(buf, 8);

releaseKey();


cardCount++;


 }


else  {

 return;

   }
   if(cardCount= 1  ){
    delay(50);


buf[0] = 0;
buf[2] = 0x28; // letter Enter
Serial.write(buf, 8);
releaseKey();

cardCount--;
    }
  }

void releaseKey()
{
buf[0] = 0;
buf[2] = 0;
buf[3] = 0;
```

```
Serial.write(buf, 8); // Release key

}
```