SURVEY PAPER ON PC LOCK/UNLOCK USING RFID & OBJECT DETECTION CONTROLLING USING ARDUINO

Prof. S.R. Madkar
Department Of Electronics Engineering
Bharati Vidyapeeth
(Deemed to be University)
College of Engineering Pune

Neelesh Rawat
Department Of Electronics Engineering
Student Bharati Vidyapeeth
(Deemed to be University)
College of Engineering Pune

Vaishali Yadav
Department Of Electronics Engineering
Student Bharati Vidyapeeth
(Deemed to be University)
College of Engineering Pune

Vaishnav Yeole
Department Of Electronics Engineering
Student Bharati Vidyapeeth
(Deemed to be University)
College of Engineering Pune

ABSTRACT

The design and implementation of security system that's unlocking a PC is finished by using often Identification RFID tag. This proposed methodology is employed to activate and authenticate for the PC users. The system is developed with microcontroller (Atmega32U4) and RFID tag. RFID tag which have unique Id is scanned in RFID reader. Atmega32U4 act as keyboard to supply password during the time of scan. Initially Atmega32U4 stored the password for various Id's and it reveals the password to unlock the PC. the most objective of the program is to unlock the PC by using an efficient microcontroller.

Keywords: *Atmega32U4, RFID, Security system*

INTRODUCTION

The security and unlock systems play a serious role to stay out unknown users to access the PC with no permission and to assist the PC users to recollect the password. Embedded systems are employed in various applications in numerous field like digital electronic, communications, computer networks, signals and systems, satellite systems, military base defense, research work, and so on. PC unlocks system using RFID is additionally one among the appliance of embedded technologies. Normal unlock system contains password and keyboard based technology. By using modern technology normal unlock is modified into a RFID based unlock system. RFID: Radio-frequency identification (RFID) is one in all the oldest wireless technologies. RFID chips are accustomed store information digitally, which might then be shared between objects through electromagnetic fields and radio waves. it's going to not be super-advanced, but many manufacturers see real potential within the technology, regardless of how old.[1,2]

COMPONENTS USED

- RC522 RFID module
- Arduino Leonardo
- RFID tag and RFID card
- Ultrasonic Sensors
- PC/Laptop
- Breadboard or PCB
- Resistors
- Connecting wires

RFID MODULE OR READER

RFID technology uses radio transmissions to identify an object. It is widely used to automate the identification of something like a barcode or QR code system but does not replace the barcode. Different types of RFID tags are available on the market for various uses. Index Terms RFID history, RFID tags layout, RFID application, RFID frequency band, RFID privacy concerns. RFID consists of a silicon chip and an antenna. RFID tags can be attached to any object. RFID tags antenna manufacturer name, model number etc. It may also store additional information compared to barcode. widely used in the app as a tracking system as a tracking system.

The advantage over the barcode is that it does not require a line to read the tag. Although RFID reader has a longer reading range compared to barcode. it will also tend to point things at the same time.

Eg. Prof can take the whole class at a time instead of taking the school one by one. The main task is to explore the skills of recording and transferring unique tag ID. [4,5,6]

RFID PIN CONFIGURATION

The RFID pin configuration is Vcc Used for Enable Module, usually 3.3V is used, RST is a reset PIN - used for reset or minimize module, GND Connected Under program, IRQ Interrupt pin - used for wake . module when the device enters the range, MISO is a MISO PIN when used for SPI communication, it acts as SCL for I2c and Tx for UART.MOSI is a Master out slave to SPI PIN Serial Clock - used to provide a clock source and SDA Serves as Serial input (SS) for SPI communication, SDA IIC and Rx during UART.

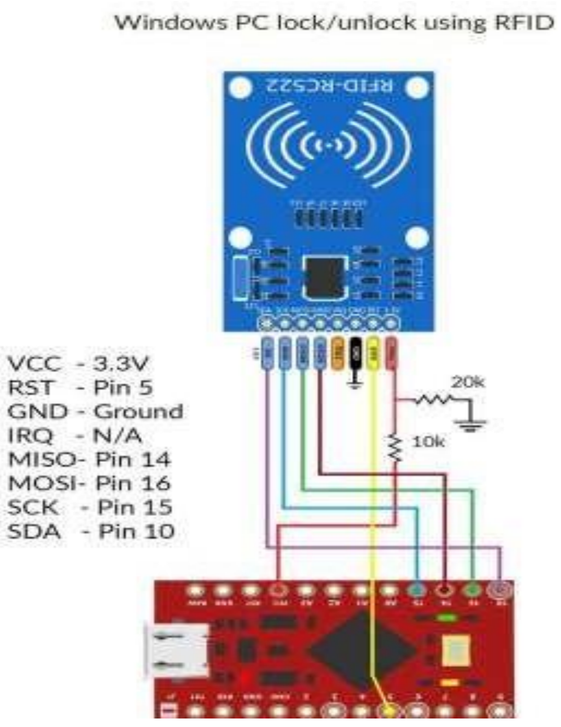
RFID TAG

RFID tags are a type of tracking technology that uses a unique UID such as QR codes to identify objects or objects. Radio Frequency Identification is a short form of RFID and RFID tags use frequency technology. Radio output from an RFID tag to an RFID reader or module is used to transfer data. But these tags will not be used to track transportation vehicles or pets. There is a chip built into this RFID so this RFID tag is also renamed RFID chips. That microchip or stick is used to transfer data to one location or to retrieve data. That chip is also called IC. That chip or microchip records the information or data that the user wants to record. These tags are used for a variety of applications.

ARDUINO LEONARDO

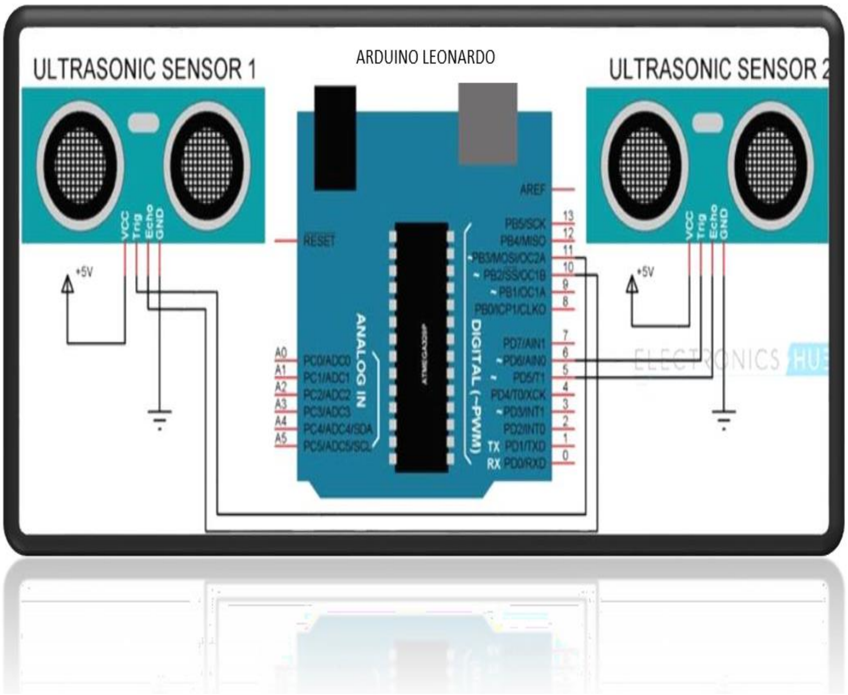
Arduino Leonardo is a microcontroller board based on ATmega32u4. It has 20 digital input / output ports (7 of which can be used as PWM outputs and 12 analog inputs), a 16 MHz crystal oscillator, a small USB connection, a power jack, an ICSP header, and the reset button. It contains everything needed to support a microcontroller; just connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

CIRCUIT CONFIGURATION



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 07 | July - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Fig. 1: Circuit Configuration



ARDUINO LEONARDO PIN CONFIGURATION

ATmega32u4 Microchip: The ATmega32u4 is a high performance, low power AVR 8-bit microchip. It has 32 KB (of which 4 KB is used for boot loader), 2.5 KB for SRAM, and 1 KB for EEPROM.

Crystal Oscillator: The Crystal oscillator inside the board has a frequency of 16MHz, which produces a clock signal on the microcontroller. Its primary function is to provide basic time and control to the board.

Micro USB Port - The Micro USB port is smaller than a standard Mini USB port and is much thinner and slightly taper than a small USB port. Allows us to connect the board between a computer and a computer. It is very important to the plans of the Arduino Leonardo board.

Vin: A voltage input pin connected to an external power supply to power the Arduino board. When a voltage is supplied with a power jacket, this PIN can be used as a power pin.

Analog Input: Leonardo contains 12 analog inputs, labeled from A0 to A11, all of these PINs can be used as digital I / O pins. Each of these analog anchors has an ADC built into 210 bit resolution (so it will deliver 1024 values).

Digital I / O Anchors: There are 20 digital I / O pins in Leonardo that can be used as inputs or outputs. They operate at 5 volts. The Arduino Leonardo digital pins can read only two circuits: when there is a voltage signal present and when not. This type of input is commonly known as digital (or binary) and these regions are called LOW and LOW or 1 and 0.

Arduino Leonardo Pinout-PWM PINs: There are six PINM (Pulse Width Modulation) digital PINM sets with numbers 3, 5, 6, 9, 10, 11, and 13. Provide 8-bit PWM output with analog Write () function

UART anchors: Arduino Leonardo UART anchors These pins are used for serial communication, 0 (RX) data acquisition, and 1 (TX) to transfer TTL serial data (TX) using ATmega32U4 hardware serial capability.

ICSP: Represents In-Circuit Serial Programming. These pins are used to edit Arduino Leonardo board firmware. New firmware updates with new capabilities are sent via microcontroller with the help of this ICSP article. The ICSP header contains 6 anchors.

Two-Wire Interface (TWI) / I2C: Arduino Leonardo I2C Anchors It is a two-wire serial communication protocol. Represents Inter-Integrated Circuits. I2C uses two lines to send and transmit data: product clock PIN (SCL) and serial PIN (SDA) data.

SPI Anchors: Represents Serial Peripheral Interface. These pins are used by microcontrollers to communicate with one or more peripheral devices effectively. Unlike the Arduino UNO SPI pins on Leonardo's board are located at the head of the ICSP, these pins support SPI communication through the SPI library. This means that even if we have a shield that uses the SPI connection, but NOT a 6-pin ICSP connector that can connect to Leonardo's 6-pin ICSP head, the shield will not work.

Other pins: 5V: 5V pin pulls 5v to external parts. The Arduino Nano 5V board power source is a USB and Vin connector. 3.3 V: The 3.3V PIN acts as a 3.3V GND outgoing controlled output. Lower pins. These are used to set the region.

ULTRASONIC SENSORS

An ultrasonic sensor is a device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the resulting sound into an electrical signal. Ultrasonic waves travel faster than the speed of sound (i.e. sound that people can hear). Ultrasonic sensors have two main components: a transmitter (which emits sound using piezoelectric crystals) and a receiver (which meets the sound after movement and departure). This is the HC-SR04 distance sensor. This savings sensor provides 2cm to 400cm of uninterrupted measuring performance with various accuracy up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, receiver and control circuit. There are only four pins you need to worry about in HC-SR04: VCC (Power), Trig (Trigger), Echo (Accept), and GND (Ground). You will find this sensor very easy to set up and use in your next distance acquisition project. This sensor has an additional control cycle that can prevent "bouncy" data being invariant depending on the application.
The Radio Transmission identification (RFID) module or student receives an input voltage value of just 3.3 volts. It is designed as a highly sensitive module or device and therefore a voltage greater than 3.3 volts can damage or overheat the device or module. Arduino Leonardo VCC PIN has a supply voltage of 5 volts. Design 5 volts to 3 volts of low step to create a supply voltage of 3.3 volts. Connect the 3.3 V supply to the VCC of the RFID module. RST to pin 5 Arduino. (You can change this PIN) Connect the GND PIN down. IRQ PIN - No Connections. MISO to pin14 Arduino. MOSI will pin 16 Arduino. SCK to pin 15 Arduino. The SDA will pin 10 of the Arduino. (This is also a user-defined PIN.).

WORKING METHODOLOGY

The operating system is shown in the image above. The operating system provided is unique ID number or UID number and personal computer password or authentication key are stored in the code or UID. If the right card is shown in the RFID module or Atmega or Arduino module will send PC lock keys and a PC unlock password at the same time. [3] If the pc is in lock time, the lock keys will not affect that so the command will turn on the PC. Atmel FLIP software converts Arduino to keyboard. By using the Arduino IDE, the program is written on the Atmega32u4 installed on the Arduino board. The editing language used here is C ++. Two programs are used, one to read the UID on the mark and the other to use to unlock your PC. To do this the Atmega328P is converted into a keyboard using Atmel Flip.

The ultrasonic sensor uses sonar to determine the object range. Here's what happens: An ultrasound transmitter (trig pin) emits high-frequency noise (40 kHz). The sound travels through the air. When it finds an object, it returns to the module. The ultrasound receiver (echo pin) detects the displayed sound (echo).
Two ultrasonic sensors ultrasonic 1 & ultrasonic 2 are connected to an Arduino Leonardo directly connected to a computer. The two ultrasonic sensors are located in a fixed position on the top left and right of the screen / portable computer monitor. The hand gesture is operated and detected by the distance of the hand, the hand-operated hand operates within a width of 7 to 35cm. Hand gestures within range are calculated by Arduino Leonardo and then collected, distance information from Arduino is collected by a C ++ system that converts data into keyboard click actions.

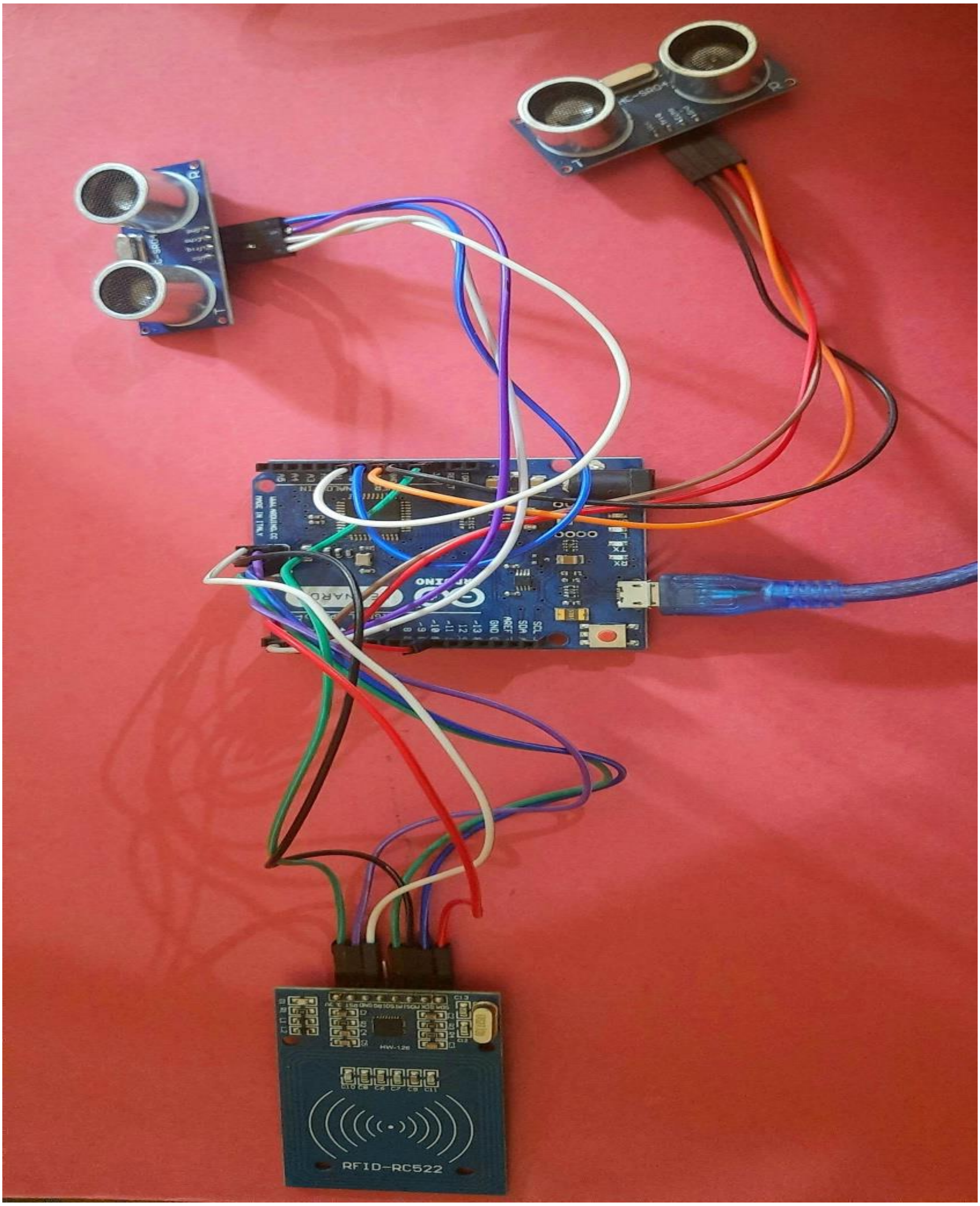


Fig. 2: Security Model

```
const int trigPin1 = 2; // the number of the trigger output pin ( sensor
1 )
const int echoPin1 = 3; // the number of the echo input pin ( sensor 1 )
const int trigPin2 = 4;  // the number of the trigger output pin (
sensor 2 )
const int echoPin2 = 5;  // the number of the echo input pin ( sensor 2
)

////////////////////////////////// variables used for distance
calculation
long duration;
int distance1, distance2;
float r;
int l=0;
////////////////////////////////

void find_distance (void);

// this function returns the value in cm.
/*we should not trigger the both ultrasonic sensor at the same time.
it might cause error result due to the intraction of the both
soundswaves.*/
void find_distance (void)
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
```

```
                                           // it helps this project to
use the gesture control in the defined space.
                                           // so that, it will return
zero if distance greater then 60m. ( it helps usually if we remove our
hands infront of the sensors ).

  r = 3.4 * duration / 2;                  // calculation to get the
measurement in cm using the time returned by the pulsein function.
  distance1 = r / 100.00;
  /////////////////////////////////////////upper part for left sensor
and lower part for right sensor
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);

  duration = pulseIn(echoPin2, HIGH, 2000);
  r = 3.4 * duration / 2;
  distance2 = r / 100.00;
  delay(100);
}


MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance
String readid;
String card1="5ba7bb22";
char Enter = KEY_RETURN;                //Return key is declared as
Enter.
```

```
void setup() {
  Serial.begin(9600);   // Initialize serial communications with the PC
  pinMode(trigPin1, OUTPUT); // initialize the trigger and echo pins of
both the sensor as input and output:
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);

  Keyboard.begin();
  while (!Serial);    // Do nothing if no serial port is opened (added
for Arduinos based on ATMEGA32U4)
  SPI.begin();      // Init SPI bus
  mfrc522.PCD_Init();   // Init MFRC522
  delay(4);       // Optional delay. Some board do need more time after
init to be ready, see Readme
  mfrc522.PCD_DumpVersionToSerial();  // Show details of PCD - MFRC522
Card Reader details
  Serial.println(F("Scan PICC to see UID, SAK, type, and data
blocks..."));
}
///////////////////////////////////////////

void temp(byte *buffer, byte bufferSize)//function to store card uid as
a string datatype.
{
  readid="";
  for(byte i = 0;i<bufferSize; i++)
  {
   readid=readid+String(buffer[i], HEX);
```

```
void control(){
  find_distance(); // this function will stores the current distance
measured by the ultrasonic sensor in the global variable "distance1 and
distance2"
                      // no matter what, the program has to call this
"find_distance" function continuously to get the distance value at all
time.
  Serial.println(distance1);
Serial.println(distance2);
if(lock_status==0){

Serial.println(distance1);
Serial.println(distance2);

  if(distance2<=35 && distance2>=7) // once if we placed our hands in
front of the right sensor in the range between 15 to 35cm this condition
becomes true.
   {

    for(int i=0;i<30;i++){
        find_distance();
        if(distance1<=35 && distance1>=7){break;}
    delay(1);
    }


     if(distance1<=35 && distance1>=7) // once if we placed our hands in
front of the right sensor in the range between 15 to 35cm this condition
becomes true.
   {
```

```
      Keyboard.releaseAll();
      delay(100);
      Keyboard.end();
// SWITCH TEB END
        }

      }else  if(distance1<=35 && distance1>=2) // once if we placed our
hands in front of the right sensor in the range between 15 to 35cm this
condition becomes true.
    {
      Serial.println("right");
      if(distance1<=10 && distance1>=2){
          Keyboard.press(KEY_PAGE_UP);
          delay(100);
          Keyboard.releaseAll();
          delay(100);
          Keyboard.end();
      }if(distance1<=35 && distance1>=10){
          Keyboard.press(KEY_PAGE_DOWN);
          delay(100);
          Keyboard.releaseAll();
          delay(100);
          Keyboard.end();

        }
      }

    }

    }// control
```

```
     {
      Serial.println("tab");
      // SWITCH APP START
      Keyboard.press(KEY_RIGHT_ALT);
      delay(200);
      Keyboard.press(KEY_TAB);
      delay(100);
      Keyboard.press(KEY_TAB);
      delay(100);
 // SWITCH APP END
      Keyboard.releaseAll();
      delay(100);
      Keyboard.end();

      Keyboard.press(0xB0);
      delay(50);
      Keyboard.releaseAll();
      delay(100);
      Keyboard.end();


      }else{
         Serial.println("left");
         // SWITCH TAB START
      Keyboard.press(KEY_RIGHT_CTRL);
      delay(200);
      Keyboard.press(KEY_TAB);
      delay(100);
      Keyboard.releaseAll();
      delay(100);
```

```
void loop() {
    // Reset the loop if no new card present on the sensor/reader. This
saves the entire process when idle.
    if ( ! mfrc522.PICC_IsNewCardPresent()) {
      Serial.println("hi");
      control();

      return;
    }

    // Select one of the cards
    if ( ! mfrc522.PICC_ReadCardSerial()) {
         Serial.println("hello");
      return;
    }

    // Dump debug info about the card; PICC_HaltA() is automatically
called
     mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
     temp(mfrc522.uid.uidByte, mfrc522.uid.size);
     Serial.print(readid);
     if(card1==readid){
      Serial.print("match");
      if(lock_status==0){
        lock_status=1;
        // SCROL lock start
      Keyboard.press(KEY_LEFT_GUI);
      delay(100);
      Keyboard.press('l');
      delay(100);
```

```
    delay(100);
    Keyboard.press('l');
    delay(100);
/ SCROL lock END


    Keyboard.releaseAll();
    delay(100);
    Keyboard.end();

      }else if(lock_status==1){
        lock_status=0;
        //    Keyboard.press(0xB1);
    delay(300);
    Keyboard.print("4455");//write your password in between the qouts
    Keyboard.press(0xB0);
    delay(50);
    Keyboard.releaseAll();
    delay(100);
    Keyboard.end();

        }
/     Keyboard.press(0xB1);
/     delay(300);
/     Keyboard.print("4455");//write your password in between the qouts
/     Keyboard.press(0xB0);
/     delay(50);
/ SWITCH APP START
/     Keyboard.press(KEY_RIGHT_ALT);
/     delay(200);
```

```
//    Keyboard.press(KEY_TAB);
//    delay(100);
//    Keyboard.press(KEY_TAB);
//    delay(100);
// SWITCH APP END
// SWITCH TAB START
//    Keyboard.press(KEY_RIGHT_CTRL);
//    delay(200);
//    Keyboard.press(KEY_TAB);
//    delay(100);
// SWITCH TEB END
// SCROL lock start
//    Keyboard.press(KEY_LEFT_GUI);
//    delay(100);
//    Keyboard.press('l');
//    delay(100);
// SCROL lock END

//
//    Keyboard.releaseAll();
//    delay(100);
//    Keyboard.end();
      }else
      {Serial.print("did not match");}


}// loop
```

Fig. 3: *Program.*

CONCLUSION

This method can be used for security, home use and office use. The whole program is used by PC users. But using the system concept there are other locking/unlocking programs that can be done as a door opening system using RFID and a library system. Man makes a mistake but the machine does not. So, this system is a modern global defense system, the emergence of technology contains a security problem but can be solved by using the same emergence of technologies such as IoT and AI. By using Arduino Leonardo (Atmega32u4) & Ultrasonic Sensors (HC-SR04) we can operate a PC or a laptop, with just a few hand gestures with the help of objection detection controlling, It can switch between tabs & scroll up & down without using the keyboard or the mouse.

REFERENCES

1. http://wiki.answers.com
2. Huang Y. Secure Access Control Scheme of RFID System Application. Fifth International Conference on Information Assurance and Security. 1, 525-528p.
3. https://www.electronicshub.org/arduino-based-hand-gesture-control-computer/#:~:text=The%20principle%20behind%20the%20Arduino,the%20hand%20and%20the%20sensor
4. Pallavi Verma, Namit Gupta: Windows PC Lock/Unlock using RFID, published 2013
5. Thomas S.J. RFID for everyone: Design of an easily-accessible, experimental UHF RFID platform. Apr 2019.
6. http://www.microchip.com/wwwproducts/en/ATmega32U4