**Difficulty Level**

# Contents

1. **Software ……………………………………………………………..……………………2-6**

1.0 Flow chart ………………………………………………………………………2-3

1.1 function definition…………………………………………………………..4-6

1.2 Release Program ……………………………………………………………6

1. **Debugging Guide………………………………………………………………………….7-8**
2. **Software references………………………………………………………………………9**

**2. Software**

**2.1 Flow Chart**

Include Header files for RFID (i.e. RFID header file and SPI header file)

Declare the pins for RFID pins, LED pins and Switch pins

Enable the UART protocol and initialize SPI communication RFID header file and key value and also turn on Led’s

If New Card Present?

No Yes

If available?

Yes

Return Nothing

Send “0” to Game Pc

Check the PCC key?

No

Yes

Read the card data and store it in buffer and send the data to Game PC

Read the Status of Switches

Send the Switches data to Game Pc

Turn on Particular Switch Connected Led and Turn on timer

If Switch is pressed d?

No yes

Yes No

Is timer is finished?

Remains On all the Led’s

**1.1 Function Definition**

1. Include header files for RFID and SPI Communications

**#include <SPI.h>**

**#include <MFRC522.h>**

1. Define Pins for reset and slave select pins

**#define RST\_PIN 9**

**#define SS\_PIN 10**

1. Define the character key , status and variable to store the string

**String value = " ";**

**MFRC522 mfrc522(SS\_PIN, RST\_PIN);**

**MFRC522::MIFARE\_Key key;**

**MFRC522::Status Code status;**

1. Define the variable for timer and maxtimer

**unsigned long timer = 0;**

**unsigned long maxTimer = 50;**

1. Declare pins for Led pins and Switch Pins

**int ledpin1 = 3,int ledpin2 = 5;int ledpin3 = 6;**

**int pushbutton1 = 7,int pushbutton2 = 4,int pushbutton3 = 2;**

1. Declare bool and normal variables for different functionalities like to know the status of switches

**bool isDiffucltySelected = false;**

**bool canStartTimer = false,int flag = 1;**

**bool b1 = false, b2 = false, b3 = false,int button1 = 0;**

**int button2 = 0,int button3 = 0;**

1. Declare Switch Pins as OUTPUT and Led Pins as INPUT

**pinMode(ledpin1, OUTPUT), pinMode(ledpin2, OUTPUT);**

**pinMode(ledpin3, OUTPUT), pinMode(pushbutton1, INPUT);**

**pinMode(pushbutton2, INPUT), pinMode(pushbutton3, INPUT);**

1. Initially Make the Leds in HIGH state

**digitalWrite(ledpin1, HIGH);**

**digitalWrite(ledpin2, HIGH);**

**digitalWrite(ledpin3, HIGH);**

1. Initialize the SPI and RFID header files to start the communication and also initialize the UART communication with baud rate settings

**Serial.begin(9600),SPI.begin();**

**mfrc522.PCD\_Init();**

1. Initialize the key for RFID

**for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;**

1. Initialize the byte and block variables with zero

**byte block = 0;**

**byte Len = 0;**

1. This function to check the new card is present or not ,if card is present then the compiler will go to else function otherwise it will send zero to the serial monitor

**if ( ! mfrc522.PICC\_IsNewCardPresent()) {**

**Serial.print("0");**

**mfrc522.PCD\_Init(); }**

1. This function will be reading the new card data

**else {**

**if ( ! mfrc522.PICC\_ReadCardSerial()) { return; }**

1. Declare the buffer length and block area

**byte buffer1[18], block = 4, len = 18;**

1. Check the RFID status

**Status=mfrc522.PCD\_Authenticate(MFRC522::PICC\_CMD\_MF\_AUTH\_KEY\_A, 4, &key, &(mfrc522.uid)); //line 834 of MFRC522.cpp file**

1. This function will return the status

**if (status != MFRC522::STATUS\_OK) { return; }**

**status = mfrc522.MIFARE\_Read(block, buffer1, &len);**

1. This loop is used to send the data to game pc and also reset the buffer size

**for (byte i = 0; i < 3; i++){**

**mfrc522.PCD\_Init(),value += (char)buffer1[i];**

**buffer1[i] = 0;} buffer1[18] = 0;**

**Serial.print(value), Serial.print(","), value = "";**

1. Read the button status

**button1 = digitalRead(pushbutton1);**

**button2 = digitalRead(pushbutton2);**

**button3 = digitalRead(pushbutton3);**

1. Send the switch status to game pc

**if (b1 == true){ Serial.print("1");**

**Serial.print(",");}**

**else{Serial.print("0");**

**Serial.print(",");}**

**if (b2 == true){Serial.print("1");**

**Serial.print(",");}**

**else{Serial.print("0");**

**Serial.print(",");}**

**if (b3 == true){Serial.println("1");**

**//Serial.print(",");}**

**else{ Serial.println("0"); }**

1. This function will check and hold the status of switch and enable the particular LED will glow and disable other LEDS and make the canstart timer as true

**if (!isDiffucltySelected) {**

**button1 = digitalRead(pushbutton1);**

**button2 = digitalRead(pushbutton2);**

**button3 = digitalRead(pushbutton3);**

**if (button1 == HIGH){**

**// Serial.println("Button 1 is pressed ");**

**digitalWrite(ledpin3, LOW);**

**digitalWrite(ledpin2, LOW);**

**digitalWrite(ledpin1, HIGH);**

**isDiffucltySelected = true;**

**canStartTimer = true;b1 = true; }**

**if (button2 == HIGH){**

**// Serial.println("Button 1 is pressed ");**

**digitalWrite(ledpin3, LOW);**

**digitalWrite(ledpin1, LOW);**

**digitalWrite(ledpin2, HIGH);**

**isDiffucltySelected = true;**

**canStartTimer = true;**

**b2 = true; }**

**if (button3 == HIGH){**

**// Serial.println("Button 1 is pressed ");**

**digitalWrite(ledpin1, LOW);**

**digitalWrite(ledpin2, LOW);**

**digitalWrite(ledpin3, HIGH);**

**isDiffucltySelected = true;**

**canStartTimer = true;**

**b3 = true; } }**

1. Check the canstart timer status is and if the timer reaches 500 ms then make the canstart timer status is false ,reset the timer and also make the all LEDs to glow

**if (canStartTimer){**

**timer++;**

**//Serial.println(timer);**

**if (timer > maxTimer){**

**// Serial.println("Time Reached turning all on now ");**

**digitalWrite(ledpin1, HIGH);**

**digitalWrite(ledpin2, HIGH);**

**digitalWrite(ledpin3, HIGH);**

**canStartTimer = false;**

**isDiffucltySelected = false; timer = 0;**

**b1 = false;b2 = false; b3 = false;}}**

1. and flush the transmitted data

**Serial.flush();**

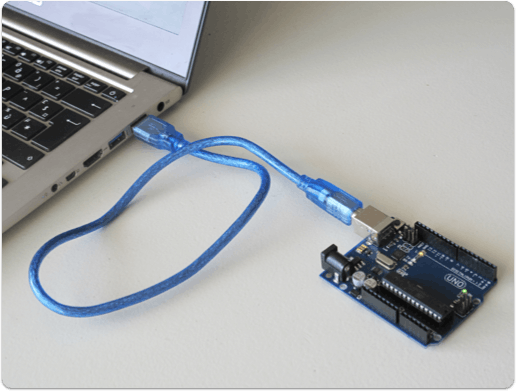
**1.2Release Program**

[**https://github.com/KRUPALATHA/RFID**](https://github.com/KRUPALATHA/RFID)

**2. Debugging Guide**

**Step 1:**

Connect the Arduino Uno Board to the CPU through the USB2.0 cable. You can’t find the Com port when the Arduino board is not working



**Step 2:**

Open the Arduino IDE and select the comport, open the serial monitor, and select the baud rate (9600) and verify the data

**Step3:**

Initially, the microcontroller will be sending values like 0,0,0,0 (RFID value, switch1, switch2and switch3), if you tap the RFID tag on the reader it will give some “3 digit ID” format and if there is no card tapped then it will be sending “0” ,if you press the switch then it will give 1 otherwise it will in 0 state

**Step4:**

After tapping RFID tag on RFID reader, if the “3 digit ID” is not showing on the game check the RFID hardware connections .then open the RFID arduino sketch and verify the data again

**Step5:**

After pressing the switch if the particular should remain stay in glowing state. If it is not in happened then check the switch connections and placement of the switch and verify the data again

**Step6:**

Initially all the led’s should be in Glowing state .if it is not in glowing state check the LED and Switch connections and verify the hardware with code again

**3. Software references**

[**https://circuits4you.com/2018/10/03/interfacing-of-rfid-rc522-with-arduino-uno/**](https://circuits4you.com/2018/10/03/interfacing-of-rfid-rc522-with-arduino-uno/) **//rfid with arduino**

[**https://www.google.com/url?sa=i&url=https%3A%2F%2Fcreate.arduino.cc%2Fprojecthub%2FSBR%2Fworking-with-an-led-and-a-push-button-71d8c1&psig=AOvVaw2lJuRjvENXSBYkMJZjxosj&ust=1630659789197000&source=images&cd=vfe&ved=0CAwQjhxqFwoTCLi91p333\_ICFQAAAAAdAAAAABAD**](https://www.google.com/url?sa=i&url=https%3A%2F%2Fcreate.arduino.cc%2Fprojecthub%2FSBR%2Fworking-with-an-led-and-a-push-button-71d8c1&psig=AOvVaw2lJuRjvENXSBYkMJZjxosj&ust=1630659789197000&source=images&cd=vfe&ved=0CAwQjhxqFwoTCLi91p333_ICFQAAAAAdAAAAABAD) **// switch and led with arduino**