Laporan

NodeMCU dan RFID-RC522 (RFID Reader dan Writer)

Modul RC522 RFID Reader dirancang untuk membuat bidang elektromagnetik dengan frekuensi 13,56MHz yang digunakannya untuk berkomunikasi dengan tag RFID (tag standar ISO 14443A). Daya digunakan modul ini tergolong kecil yaitu 2,5v sampai 3,3v dengan jarak pembacaan 5cm. Modul ini tidak hanya digunakan untuk pembacaan suatu RFID tag namun juga dapat digunakan untuk melakukan penulisan data pada RFID Tag.

Pin RST pada RFID-RC522 akan disambungkan pada pin D1

Pin SDA pada RFID-RC522 akan disambungkan pada pin D2

Pin SCK pada RFID-RC522 akan disambungkan pada pin D5

Pin MOSI pada RFID-RC522 akan disambungkan pada pin D7

Pin MISO pada RFID-RC522 akan disambungkan pada pin D6

Pin GND pada RFID-RC522 akan disambungkan pada pin GND

Pin 3.3v pada RFID-RC522 akan disambungkan pada pin 3v

|  |
| --- |
| Fungsi untuk melakukan autentikasi kunci pada RFID Tag |
| void authenticate\_key() {  MFRC522::MIFARE\_Key key;  // Kunci yang ada pada RFID Tag  for (byte i = 0; i < 6; i++) {  key.keyByte[i] = 0xFF;  }  // Proses Autentikasinya menggunakan kunci A  status = mfrc522.PCD\_Authenticate(MFRC522::PICC\_CMD\_MF\_AUTH\_KEY\_A, block, &key, &(mfrc522.uid));  } |

|  |
| --- |
| Fungsi untuk melakukan suatu pembacaan pada RFID Tag |
| void read\_card() {    block = 1; //Untuk Memilih block berapa yang akan dibaca  len = 18;    // Autentikasi kunci RFID Tag  authenticate\_key();    // Memunculkan Status RFID Tag jika gagal diautentikasi  if (status != MFRC522::STATUS\_OK) {  Serial.println("Authentication failed!");  }    // Membaca block yang telah dipilih  status = mfrc522.MIFARE\_Read(block, buffer, &len);  if (status != MFRC522::STATUS\_OK) {  Serial.println("Reading failed!");  return;  }    // Menampilkan ke serial monitor data yang ada di block 1  for (uint8\_t i = 0; i < 16; i++) {  if (buffer[i] != 32)  Serial.write(buffer[i]);  }    // Menghentikan komunikasi agar dapat membuat komunikasi yang baru  mfrc522.PICC\_HaltA();  mfrc522.PCD\_StopCrypto1();  mfrc522.PCD\_Reset();  delay(100);  mfrc522.PCD\_Init(); // inisialisasi MFRC522 chip  } |

|  |
| --- |
| Fungsi untuk melakukan suatu penulisan pada RFID Tag |
| Serial.setTimeout(30000L); // Menunggu inputan selama 30 detik  // Membaca masukan dengan diakhiri sebuah karakter bintang  len = Serial.readBytesUntil('\*', (char \*) buffer, 18);    // Jika kurang dari 16 karakter, maka karakter sisanya akan diganti dengan karakter kosong  if (len <= 16) {  for (byte i = len; i < 16; i++)  buffer[i] = ' ';  // akan ditulis di block 1  block = 1;    // Autentikasi kunci RFID Tag  authenticate\_key();  // Memunculkan Status RFID Tag jika gagal diautentikasi  if (status != MFRC522::STATUS\_OK) {  Serial.println("Authentication failed!");  }  // Melakukan Penulisan pada Block 1 dengan karakter yang telah diberikan  status = mfrc522.MIFARE\_Write(block, buffer, 16);  // Menampilkan status penulisan berhasil atau tidak  if (status != MFRC522::STATUS\_OK) {  Serial.print("Write failed!");  return;  }  else {  Serial.print("Write success!");  }  }  // Menghentikan komunikasi agar dapat membuat komunikasi yang baru  mfrc522.PICC\_HaltA();  mfrc522.PCD\_StopCrypto1();  mfrc522.PCD\_Reset();  delay(100);  // inisialisasi MFRC522 chip  mfrc522.PCD\_Init();    } |

|  |
| --- |
| Sourcecode secara keseluruhan yang digunakan pada NodeMCU yang telah tersambung dengan RFID Reader/Writer |
| #include <SPI.h>  #include <MFRC522.h>  #define RST\_PIN D1 // Konfigurasi untuk Pin RST ke NodeMCU  #define SS\_PIN D2 // Konfigurasi untuk Pin SS ke NodeMCU  MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Inisialisasi jika RFID-RC522 tersambung dengan NodeMCU dengan Pin yang telah dipilih  bool rfid\_tag\_present\_prev = false;  bool rfid\_tag\_present = false;  int \_rfid\_error\_counter = 0;  bool \_tag\_found = false;  // Variabel yang mungkin dibutuhkan  byte block, len, buffer[18];  // MFR buffer untuk status code  MFRC522::StatusCode status;  // Buffer untuk reader mode  String reader\_mode;  void authenticate\_key() {  MFRC522::MIFARE\_Key key;  for (byte i = 0; i < 6; i++) {  key.keyByte[i] = 0xFF;  }  status = mfrc522.PCD\_Authenticate(MFRC522::PICC\_CMD\_MF\_AUTH\_KEY\_A, block, &key, &(mfrc522.uid));  }  void read\_card() {  // Mengatur supaya hanya membaca block 1  block = 1;  len = 18;    // autentikasi RFID Tag  authenticate\_key();    // menampilkan pesan autentikasi  if (status != MFRC522::STATUS\_OK) {  Serial.println("Authentication failed!");  }    // Read the block 1  status = mfrc522.MIFARE\_Read(block, buffer, &len);  if (status != MFRC522::STATUS\_OK) {  Serial.println("Reading failed!");  return;  }    // Print what's inside the block 1  for (uint8\_t i = 0; i < 16; i++) {  if (buffer[i] != 32)  Serial.write(buffer[i]);  }  mfrc522.PICC\_HaltA();  mfrc522.PCD\_StopCrypto1();  mfrc522.PCD\_Reset();  delay(100);  mfrc522.PCD\_Init();  }  void write\_card() {  Serial.setTimeout(30000L);  len = Serial.readBytesUntil('\*', (char \*) buffer, 18);  if (len <= 16) {  for (byte i = len; i < 16; i++)  buffer[i] = ' ';  // Select default block to be written  block = 1;    // Authentication  authenticate\_key();  // Authentication status  if (status != MFRC522::STATUS\_OK) {  Serial.println("Authentication failed!");  }  // Write block  status = mfrc522.MIFARE\_Write(block, buffer, 16);  if (status != MFRC522::STATUS\_OK) {  Serial.print("Write failed!");  return;  }  else {  Serial.print("Write success!");  }  }  mfrc522.PICC\_HaltA();  mfrc522.PCD\_StopCrypto1();  mfrc522.PCD\_Reset();  delay(100);  mfrc522.PCD\_Init();    }  void setup() {  Serial.begin(9600); // Initialize serial communications with the PC  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  }  void loop() {  rfid\_tag\_present\_prev = rfid\_tag\_present;  \_rfid\_error\_counter += 1;  if(\_rfid\_error\_counter > 2){  \_tag\_found = false;  }  // Detect Tag without looking for collisions  byte bufferATQA[2];  byte bufferSize = sizeof(bufferATQA);  // Reset baud rates  mfrc522.PCD\_WriteRegister(mfrc522.TxModeReg, 0x00);  mfrc522.PCD\_WriteRegister(mfrc522.RxModeReg, 0x00);  // Reset ModWidthReg  mfrc522.PCD\_WriteRegister(mfrc522.ModWidthReg, 0x26);  MFRC522::StatusCode result = mfrc522.PICC\_RequestA(bufferATQA, &bufferSize);  if(result == mfrc522.STATUS\_OK){  if ( ! mfrc522.PICC\_ReadCardSerial()) { //Since a PICC placed get Serial and continue  return;  }  \_rfid\_error\_counter = 0;  \_tag\_found = true;  }    rfid\_tag\_present = \_tag\_found;    // rising edge  if (rfid\_tag\_present && !rfid\_tag\_present\_prev){  if (Serial.available()) {  reader\_mode = Serial.readStringUntil('#');  if (reader\_mode.equals("write") == true) {  write\_card();  }  }  else {  read\_card();  }  }    // falling edge  if (!rfid\_tag\_present && rfid\_tag\_present\_prev){  Serial.print("Done");  }  } |