Arduino-Based Smart Security System using RFID and PIR sensor

Project Overview

This project is a compact Arduino-based security system I designed using an Arduino Mega and a mix of basic but powerful modules. It's fully powered by a 7.4V battery source made from two 3.7V Li-ion cells in series. The power is stepped down to 5V using an LM2596 buck converter, which keeps everything safely powered, including the Arduino and all connected sensors.

Hardware

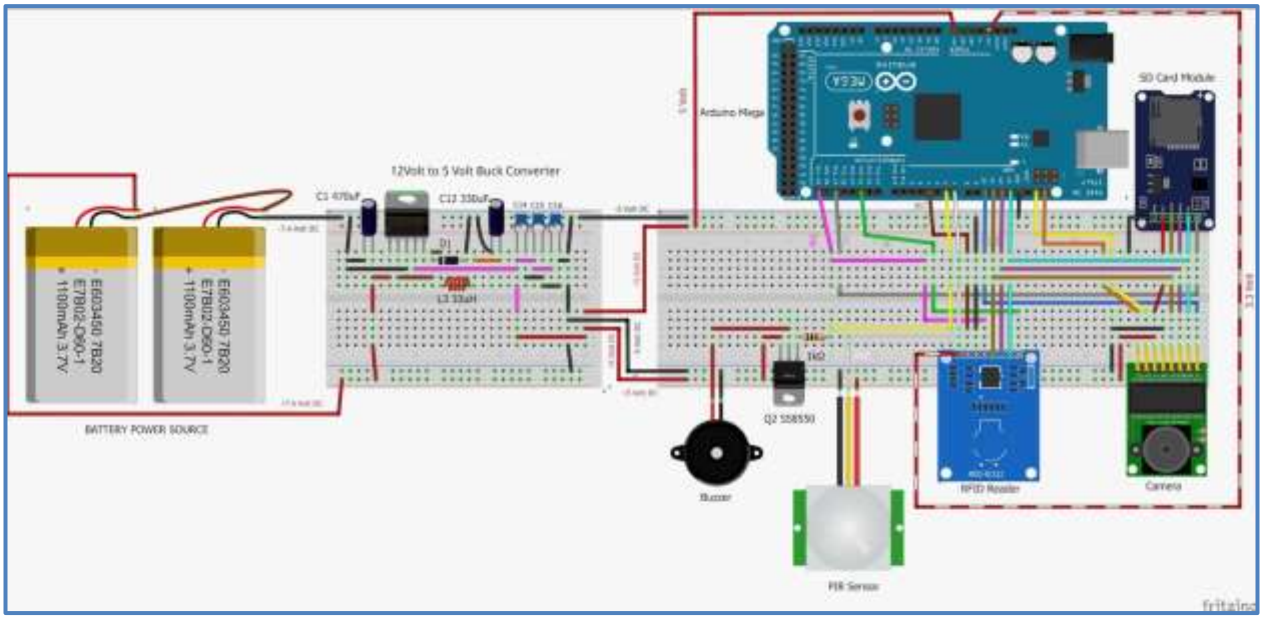
The system includes a PIR sensor for motion detection, an RFID RC522 reader to identify if the person is authorized, a camera module to capture an image, and an SD card module to store images or logs. There's also a buzzer that acts as an alert mechanism when something unusual is detected — like an unauthorized access attempt or motion without an RFID tag.

Working:

Here's how it works: when motion is detected by the PIR sensor, the RFID reader gets activated. If a valid RFID tag is detected, the camera takes a picture and saves it to the SD card. If the tag is invalid or no tag is scanned within a certain time, the buzzer turns on to alert nearby users. Everything runs off the battery, making this system suitable for places where a wired power source isn't available.

I provided the full schematic, code, and breadboard layout as part of the delivery. The whole setup is simple but functional, and can be expanded depending on what kind of logging, alerting, or automation the client needs.

Schematics



Code

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

// RFID Reader pins
#define RST_PIN 4
#define SS_PIN 16  // RFID CS pin

// SD Card pins
#define SD_CS_PIN 53  // SD Card CS pin

// Create MFRC522 instance
MFRC522 mfrc522(SS_PIN, RST_PIN);

// File object for SD card
File dataFile;

// Test function for SD card
bool testSDCard() {
  Serial.println("\n=== Testing SD Card ===");
  Serial.print("Initializing SD card...");

  if (!SD.begin(SD_CS_PIN)) {
    Serial.println("FAILED!");
    Serial.println("Please check:");
    Serial.println("1. SD card is inserted");
    Serial.println("2. SD card is formatted as FAT16 or FAT32");
    Serial.println("3. SD card module connections");
    return false;
  }

  Serial.println("SUCCESS!");

  // Try to create a test file
  Serial.print("Creating test file...");
  dataFile = SD.open("test.txt", FILE_WRITE);
  if (dataFile) {
    dataFile.println("SD Card Test Successful!");
    dataFile.close();
    Serial.println("SUCCESS!");
    return true;
  } else {
    Serial.println("FAILED!");
    return false;
  }
}

// Test function for RFID reader
bool testRFIDReader() {
  Serial.println("\n=== Testing RFID Reader ===");
  Serial.print("Initializing RFID reader...");

  mfrc522.PCD_Init();
  delay(4);

  // Try to read card
  Serial.println("\nPlease place an RFID card near the reader...");
  Serial.println("Waiting for card...");

  unsigned long startTime = millis();
  while (millis() - startTime < 5000) { // Wait for 5 seconds
    if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
      Serial.println("Card detected!");
      Serial.print("UID: ");
      for (byte i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        Serial.print(" ");
      }
      Serial.println();
      return true;
    }
    delay(100);
  }

  Serial.println("No card detected within 5 seconds!");
  return false;
}

void setup() {
  // Initialize Serial communication
  Serial.begin(9600);
  while (!Serial) {
    ; // Wait for serial port to connect
  }
  Serial.println("\n=== RFID Card Reader Test Program ===");

  // Initialize SPI bus
  SPI.begin();

  // Run tests
  bool sdTest = testSDCard();
  bool rfidTest = testRFIDReader();

  if (sdTest && rfidTest) {
    Serial.println("\n=== All tests passed! System is ready ===");
    Serial.println("You can now scan cards and data will be saved to SD card");
  } else {
    Serial.println("\n=== Some tests failed! Please check the errors above ===");
  }

  // Create a new log file for this session
  String fileName = "rfid_log_" + String(millis()) + ".csv";
  dataFile = SD.open(fileName, FILE_WRITE);
  if (dataFile) {
    dataFile.println("Timestamp,Card UID");
    dataFile.close();
    Serial.println("Created new log file: " + fileName);
  }
}

void loop() {
  // Look for new cards
  if (!mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

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

  // Get current timestamp
  unsigned long timestamp = millis();

  // Prepare card UID string
  String cardUID = "";
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    cardUID += (mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
    cardUID += String(mfrc522.uid.uidByte[i], HEX);
    cardUID += (i < mfrc522.uid.size - 1 ? ":" : "");
  }

  // Print to Serial Monitor
  Serial.print("Card detected! UID: ");
  Serial.println(cardUID);

  // Open file in append mode
  dataFile = SD.open("rfid_log.csv", FILE_WRITE);
  if (dataFile) {
    // Write data to file
    dataFile.print(timestamp);
    dataFile.print(",");
    dataFile.println(cardUID);
    dataFile.close();
    Serial.println("Data saved to SD card");
  } else {
    Serial.println("Error opening file for writing");
  }

  // Halt PICC and stop encryption
  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();

  // Small delay before next read
  delay(1000);
}
```