## UNIVERSITÄT “POLITEHNICA” BUKAREST FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN IN FREMDSPRACHEN ANGEWANDTE ELEKTRONIK

BACHELOR THESIS

INTELLIGENTES SCHACHBRETT MIT STELLUNGSANALYSE

Wissenschaftlicher Betreuer:

Conf. Dr. Ing. Andrei DRUMEA

Student: Mihaela STROE

Bukarest 2023

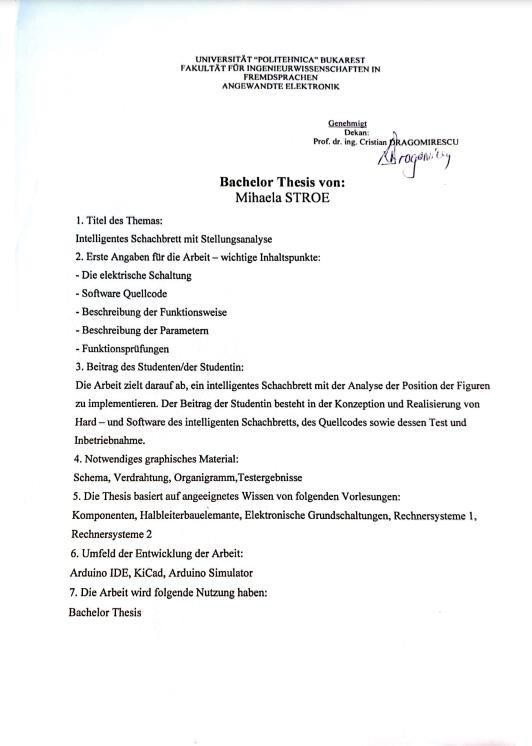
UNIVERSITÄT “POLITEHNICA” BUKAREST FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN IN

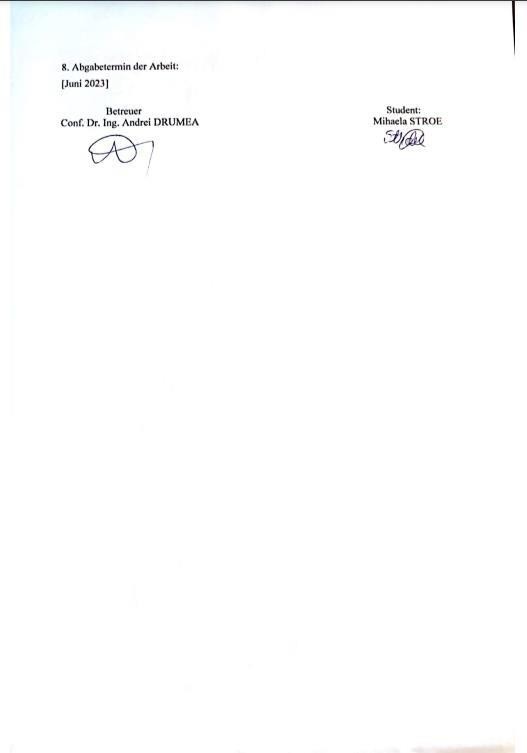
FREMDSPRACHEN ANGEWANDTE ELEKTRONIK

INTELLIGENTES SCHACHBRETT MIT STELLUNGSANALYSE

Wissenschaftlicher Betreuer Conf. Dr. Ing. Andrei DRUMEA Student Mihaela STROE

Bukarest 2023





## Ehrenerklärung

Hiermit erkläre ich, Mihaela STROE, dass die vorliegende Bachelor Thesis mit dem Titel

„INTELLIGENTES SCHACHBRETT MIT STELLUNGSANALYSE“, welche vor dem

Prüfungsausschuss der Fakultät für Ingenieurwissenschaften in Fremdsprachen, an der Universität „Politehnica“ Bukarest, zur Erlangung des Bachelor – Titels verteilgt wird, Ergebnis meiner Arbeit ist und auf meiner eigenständigen Forschung basiert.

Die Bachelor – Thesis ist selbstständig verfasst un die entsprechenden Simulationen, Experimente und Messungen sind eigenständig durchgeführt worden, unter Anleitung des wissenschaftlichen Betreuers und ohne Einbezug von Personen, deren Name und Beitrag nicht in der Danksagung erwähnt worden sind.

Die Bachelor – Thesis hat in gleicher oder ähnlicher Form noch keiner Universität, Hachschuleoder keinem Forschungsausschuss im In – oder Ausland vorgelegen.

Alle benutzten Informationen, einschließlich deren aus dem Internet, entstammen Quellen, welche zitiert und im Literaturverzeichnis gemäß ethischen Standards angeführt worden sind. Die Ergebnisse der Simulationen, Experimente und Messungen sind authentisch.

Ich verstehe, dass Plagiat und Fälschung von Daten und Ergebnissen rechtswidrig und strafbarsind.

Bukarest, den 09.12.2022 Mihaela STROE

A picture containing invertebrate, worm  Description automatically generated

## Inhaltverzeichnis

1. [Einleitung 9](#_bookmark0)
   1. [Relevanz und Aktualität des Themas 9](#_bookmark1)
   2. [Ziele und Wissenschaftliche Instrumente der Arbeit 10](#_bookmark2)
   3. [Struktur der Arbeit 10](#_bookmark3)
2. [Der Stand der Technik in Intelligente Gesellschaftsspiele 12](#_bookmark4)
3. [Anforderungen und Lösungen für die Erstellung eines intelligentes Schachbretts 16](#_bookmark8)
4. [Hardwareaufbau des Schachbretts 22](#_bookmark12)
   1. [Blockschaltbild des Systems 22](#_bookmark13)
   2. [Der Eingangsblock 23](#_bookmark15)
      1. [Kurze Vorstellung des RFID-Readers 23](#_bookmark16)
      2. [Die Rolle von RFID - Lesegeräten und - Etiketten im System 28](#_bookmark20)
   3. [Verarbeitungs- und Kontrollblock 30](#_bookmark23)
      1. [Kurze Vorstellung des Arduino Mega 2560 Mikrocontroller - Entwicklungsboards 30](#_bookmark24)
      2. [Die Rolle des Arduino Mega 2560 - Mikrocontrollers bei der Datenverarbeitungund](#_bookmark26) [Systemsteuerung 32](#_bookmark26)
      3. [SPI-Protokoll 33](#_bookmark27)
      4. [Die Verwendung von LEDs bei der Erkennung von Schachfiguren 36](#_bookmark30)
      5. [Eine kurze Einführung in Hall-Effekt-Sensoren 37](#_bookmark31)
      6. [Verwendung von Hall-Effekt-Sensoren im System 38](#_bookmark32)
5. [Software-Struktur des Schachbretts 40](#_bookmark33)
   1. [Darstellung der Softwarestruktur 40](#_bookmark34)
   2. [Codestruktur für die Modulkonfiguration und -implementierungKommunikationsprotokolle 40](#_bookmark35)
      1. [Erkennung des Bauteiltyps durch Softwareverfahren 40](#_bookmark36)
      2. [RFID-ID-Tracking und Figurenerkennung 42](#_bookmark38)
      3. [Beginn der Ablesung der Anzahl der Figuren und Speicherung der Positionen für jede Figur 44](#_bookmark41)
6. [Physischer Aufbau des Schachbretts 47](#_bookmark44)
7. [Erzielte Ergebnisse 51](#_bookmark50)

[Schlussfolgerungen, persönliche Beiträge und weitereVerbesserungen 53](#_bookmark51)

[Schlussfolgerungen 53](#_bookmark52)

[Persönliche Beiträge 53](#_bookmark53)

[Weitere Verbesserungen 55](#_bookmark54)

[Literaturverzeichnis 56](#_bookmark55)

[Anhänge 57](#_bookmark56)

Abbildungsverzeichnis

[Abbildung 2. 1 Schachbrett, das die Erkennung von Schachfiguren ermöglicht. Quelle: [1] 12](#_bookmark5)

[Abbildung 2. 2 Der Zauberwürfel GoCube. Quelle: [2] 13](#_bookmark6)

[Abbildung 2. 3 Die GoDice Smart Dice von GoCube. Quelle: [3] 14](#_bookmark7)

[Abbildung 3. 1 Intelligent Schachbrett – Spezifikationen aus der Sicht des Nutzers 16](#_bookmark9)

[Abbildung 3. 2 Ermittlung des Verhaltens des Intelligenten Schachbretts 17](#_bookmark10)

[Abbildung 3. 3 Schachbrettfunktionen sind in grundlegende Funktionen unterteilt 19](#_bookmark11)

[Abbildung 4. 1 Blockschaltbild des intelligenten Schachbretts 22](#_bookmark14)

[Abbildung 4. 2 Stiftstruktur des RFID-Lesegeräts. Quelle: [4] 23](#_bookmark17)

[Abbildung 4. 3 RFID Tag. Quelle: [5] 29](#_bookmark21)

[Abbildung 4. 4 RFID-Lesegerät RC522. Quelle: [7] 30](#_bookmark22)

[Abbildung 4. 5 Arduino Mega 2560 Pinout. Quelle: [8] 30](#_bookmark25)

[Abbildung 4. 6 Anschluss von SPI-Protokoll-spezifischen Pins. Quelle: [9] 34](#_bookmark28)

[Abbildung 4. 7 Übertragung eines Datenpakets über das SPI-Protokoll. Quelle: [10] 35](#_bookmark29)

[Abbildung 5. 1 Software-Einführung von Schachfiguren. Quelle: [11] 40](#_bookmark37)

[Abbildung 5. 2 RFID-ID-Tracking und Figurenerkennung. Quelle: [11] 42](#_bookmark39)

[Abbildung 5. 3 RFID-ID-Tracking und Figurenerkennung. Quelle: [11] 43](#_bookmark40)

[Abbildung 5. 4 Beginn der Ablesung der Anzahl der Figuren und Speicherung der Positionen für](#_bookmark42) [jede Figur. Quelle: [11] 44](#_bookmark42)

[Abbildung 5. 5 Beginn der Ablesung der Anzahl der Figuren und Speicherung der Positionen für](#_bookmark43) [jede Figur. Quelle: [11] 45](#_bookmark43)

[Abbildung 6. 1 Plexiglasplatte für Schachbrett 47](#_bookmark45)

[Abbildung 6. 2 USB-Verbindung zwischen PC und Arduino Mega 2560-Board 48](#_bookmark46)

[Abbildung 6. 3 Schachfiguren mit RFID-Tag und Neodym-Magneten montiert 49](#_bookmark47)

[Abbildung 6. 4 Endgültige elektrische Verbindungen 49](#_bookmark48)

[Abbildung 6. 5 Abschlusspräsentation des intelligenten Schachbretts 50](#_bookmark49)

## Tabellenverzeichnis

[Tabelle 4. 1 Stiftstruktur des RFID-Lesegeräts. Quelle: [4] 24](#_bookmark18)

[Tabelle 4. 2 Hauptmerkmale des RFID-Lesegeräts. Quelle: [4] 27](#_bookmark19)

# Einleitung

## Relevanz und Aktualität des Themas

Schach ist ein Brettspiel für zwei Spieler, das vermutlich im 6. Jahrhundert n. Chr. in Indien entstanden ist. Das Spiel hieß „Chaturanga“, was auf Sanskrit „vier Divisionen des Militärs“ bedeutet, und wurde mit Figuren gespielt, die vier verschiedene Arten von Truppenrepräsentierten: Elefanten, Wagen, Reiter und Fußsoldaten.

Schach bietet eine Vielzahl von Vorteilen, sowohl für den Geist als auch für die Persönlichkeitsentwicklung. Hier sind einige der wichtigsten Vorteile von Schach: Verbessert diekognitive Funktion, Fördert die Kreativität, Verbessert die analytischen Fähigkeiten, Fördert diesozialen Fähigkeiten, Stärkt das Selbstbewusstsein, kann als Therapie eingesetzt werden.

Die technische Evolution des Schachs wurde von Fortschritten in Technologie und Computerwissenschaften vorangetrieben. Hier sind einige wichtige Entwicklungen, die die technische Entwicklung des Schachs vorangetrieben haben: Schachprogramme (Ab den 1950er Jahren wurden Schachprogramme entwickelt, um das Spiel auf Computern zu simulieren. In denfolgenden Jahrzehnten wurden diese Programme immer ausgefeilter und leistungsfähiger), elektronische Schachbretter (in den 1990er Jahren wurden elektronische Schachbretter entwickelt, die die Züge automatisch registrieren und speichern konnten. Diese Bretter wurden für Schachturniere und Trainingssitzungen eingesetzt), Online – Schach (mit dem Aufkommen des Internets in den 1990er Jahren wurde Online – Schach immer populärer. Heute gibt es zahlreiche Schach – Plattformen, die es Spielern aus der ganzen Welt ermöglichen, online gegeneinander anzutreten), Künstliche Intelligenz (in den letzten Jahren haben Entwicklungen in der künstlichen Intelligenz das Schachspiel revolutioniert. Durch die Verwendung von MachineLearning und Deep Learning – Technologien haben Schachprogramme enorme Fortschritte gemacht und sind mittlerweile in der Lage, sogar die besten menschlichen Spieler der Welt zu besiegen), Schachvarianten (neben dem klassischen Schachspiel haben sich im Laufe der Jahre zahlreiche Schachvarianten entwickelt, die auf verschiedenen Brettern und mit verscheidenen Regeln geschpielt werden. Einige dieser Varianten werden sogar von Schachprogramm simuliert und können online gespielt werden).

Alle zuvor bereitgestellten Informationen beeinflussten die Auswahl des Themas

in der aktuellen Arbeit, nämlich die Entwicklung eines intelligenten Schachbretts, das den Benutzern die Schachregeln beibringen soll (durch Hinzufügen von Software zur Bestimmung der Positionder Schachfiguren).

## Ziele und Wissenschaftliche Instrumente der Arbeit

Ziel der Arbeit ist es, ein intelligentes Schachbrett zu entwickeln, das die Platzierung der Figuren bestimmen kann. Um dies zu erreichen, müssen die verschiedenen Phasen durchlaufenwerden:

* Die Schaffung eines physischen Objekts, das einem intelligenten Schachbrett simuliert und Informationen über den Zustand seiner Bestandteile übermitteln kann;
* Die Schaffung eines mikrocontrollerbasierten Systems, das Daten über auf dem Schachbrett platzierte Figuren abrufen und potenzielle Bewegungen für die vom Benutzeraufgenommene Figur identifizieren kann;
* Erstellen des Codes für die Modulkonfiguration, der die Kommunikation ermöglicht unddie Definition und Einhaltung eines Satzes schachspezifischer Regeln sicherstellt;
* Überprüfen der Software sowie der Hardware, aus denen das System besteht, aufordnungsgemaßen Betrieb.

## Struktur der Arbeit

Die sieben Kapitel der aktuellen Arbeit sind wie folgt gegliedert:

* Kapitel 1 befasst sich mit der Relevanz und Aktualität des Papiers, dem Ziel und den wissenschaftlichen Werkzeugen und Instrumenten sowie dem Aufbau des Papiers.
* Durch die Untersuchung der Funktionsweise mehrere elektrischer Schaltungen, die bereits in der Kategorie der Spielgesellschaft verwendet werden, beschreibt Kapitel 2 den Tätigkeitsbereich, in den das Thema des Artikel fällt;
* In Kapitel 3 werden eine Reihe von Richtlinien festgelegt, die beim Erstellen einesintelligenten Schachbretts eingehalten werden müssen;
* Durch die Untersuchung der funktionalen Bausteine des Systems und der zugrunde liegenden Komponenten jedes Blocks geht Kapitel 4 zur Hardwareimplementierung des Systems über. Außerdem werden in diesem Kapitel die Hauptmerkmale der verwendetenModule skizziert;
* Die Prozesse, die zur Realisierung des Codes führen, werden in Kapitel 5 detaliiert beschrieben, wobei der Schwerpunkt auf den Operationen liegt, die helfen, die Regelndes Schachspiels zu implementieren, und wie das primäre Programm verwendet wird,wenn ein Zug gemach wird;
* Das mechanische Designverfahren und die Schritte, die ausgeführt werden, um dasSchachbrett physikalisch real zu machen, werden in Kapitel 6 beschrieben;
* Die experimentellen Ergebnisse, die nach der Bewertung der Hardware – und Softwarekomponenten des Systems gesammelt wurden, sowie die Quellcodedateien, ausdenen sie bestehen, werden in Kapitel 7 vorgestellt.

# Der Stand der Technik in Intelligente Gesellschaftsspiele

Die Anordnung der Figuren, die auf dem Schachbrett erscheinen können, kann auf mehr als mehrere Millionen verschiedene Arten kombiniert werden, was es zu einem komplexen Sportmit klar definierten Regeln macht. Schach ist ein Brettspiel, das die Verwendung fast bekannter Informationen (wie die ersten 10 – 20 Züge eines Schachspiels, von Experten als „Eröffnungen“bezeichnet) und die Verbindung einiger logischer Ideen erfordert, um einen Plan zu erstellen wird befolgt, um das Spiel zu gewinnen. Schach, Backgammon, das Auflösen des Zauberwürfelsund andere Spiele, die strategische Entscheidungen über Glück belohnen, fallen neben Schach indiese Kategorie.

Anschließend wird die konkrete Umsetzung einiger laufender Projekte in diesem Bereichunter Einsatz eingebetteter Systeme gezeigt. Auch reine Software – Implementierungen sind nicht zu übersehen, sind aber in dieser Arbeit kein Diskussionthema.

Das erste Beispiel besteht aus der Herstellung eines hölzernen Schachbretts, das hochzuverlässige Figurenerkennungssesoren bietet, auf denen jede Figur auf jedem Feld nach Art und Farbe identifiziert werden kann. Die Beantragung erfolgt durch die Firma „Chessnut Tech“, nähere Einzelheiten sind unter dem als Quelle angegebenen Link [1] im Literaturverzeichnis abrufbar.



Abbildung 2. 1 Schachbrett, das die Erkennung von Schachfiguren ermöglicht. Quelle: [1]

Die von Chessnut entwickelte Technologie der elektromagnetischen Resonanz (EMR) besteht aus einer Schicht von Sensoren, die unter dem Schachbrett platziert sind. Es kann die Bewegungen von E – Stücken (die Sensoren enthalten) genau und schnell erkennen. Benutzer können dank der Technologie und des ausgeklügelten Algorithmus dahinter im Blitz – oder sogar Bullet – Stil spielen, sodass sie sich keine Sorgen machen müssen, dass das Erkennungssystem nicht mit ihrer Schachgeschwindigkeit Schritt halten kann.

Chessnut erstellt Multi – Plattform – Software für Benutzer, um viele Möglichkeiten der Nutzung des digitalen Schachspiels zu unterstützen, einschließlich Windows, Mac OS, Android und IOS. Das digitale Schachspiel ermöglicht es Benutzern, Spiele sowohl aufzuzeichnen als auch online an andere zu übertragen. Benutzer der Software können gegen leistungsstarke Schach – KI – Engines spielen und sogar ihre eigenen KI – Engines erstellen oder einbinden, dieUCI – Standards unterstützen [1].

Die nächste Anwendung wird von einer Firma namens GoCube erstellt und besteht aus einem Zauberwürfel, der von einem Softwareprogramm unterstützt wird, das auf Mobilgeräten mit den Betriebssytemen Android oder IOS installiert werden kann [2].



Abbildung 2. 2 Der Zauberwürfel GoCube. Quelle: [2]

Die Anwendung verfolgt die Bewegungen des Benutzers (unter Verwendung einiger Sensoren, die die Positionen der Quadrate auf den Flächen des Würfels

messen) und sendet die Daten an eine Software, die die folgenden Funktionen ausführt: verfolgt die Zeit, die zum Lösendes Zauberwürfels benötigt wird, erstellt Rätsel fur das Benutzertraining und ermöglicht Benutzern, sich mit dem Internet zu verbinden, um an Online – Wettbewerben mit anderen Rubik’s Cube – Besitzern teilzunehmen [2].

Das letzte unten gezeigte Beispiel, das ebenfalls von der Firma GoCube stammt, beinhaltet die Erstellung eines intelligenten Würfelsatzes, der äußerst kompakt in der Größe ist. Würfel sind ein fester Bestandteil der meisten Brettspiele, bei denen Glück über den Sieg entscheidet, auch wenn sie nicht unter die Kategorie der Strategiespiele fallen.



Abbildung 2. 3 Die GoDice Smart Dice von GoCube. Quelle: [3]

Die Würfel sind mit Beschleunigungsmessern ausgestattet, die die Bewegung des Würfels verfolgen, um Daten über die auf seiner Oberseite angezeigte Zahl zu extrahieren, und mit einem Bluetooth-Modul, das die Weiterleitung dieser Daten an eine mobile Anwendung ermöglicht. Während Sie ein Brettspiel mit anderen im selben Raum oder mit Spielern aus der Internetumgebung spielen, können die nach dem Würfelwurf erhaltenen Werte in dieser Softwareanwendung gespeichert werden. Aufgrund des in diesem Gerät enthaltenen Superkondensators, dessen Ladezeit nur etwa zehn Sekunden beträgt, ist es ein gutes Beispiel fürein kompaktes eingebettetes System, dessen Betrieb bis zu einer Stunde dauern kann [3].

Dies sind nur drei Beispiele für aktuelle intelligente Systeme. Die Hardware- und

Softwareoperationen des intelligenten Schachbretts werden im folgenden Abschnitt der Arbeit zusammen mit Anweisungen zu seiner Verwendung detailliert beschrieben, damit der Leser sehen kann, wie dieses System in den Bereich der in diesem Kapitel beschriebenen Aktivitäten passt.

# Anforderungen und Lösungen für die Erstellung eines intelligentes Schachbretts

Die Ermittlung der Bedürfnisse der vorgesehenen Kunden ist der erste Schritt in der Produktentwicklung, da diese Faktoren auch die potenziellen Designanforderungen des Systemsbeeinflussen. Der Zielmarkt für dieses Produkt sind daher diejenigen, die sich zum ersten Mal mit den Anforderungen des Schachspiels auseinandersetzen. Um nur einige zu nennen, gibt es Schachklubs für Anfänger, Schach als Sport, Studenten, die studieren, und nicht zuletzt Schachbegeisterte jeden Alters.

Nach einer gründlichen Überprüfung des Arbeitsumfangs wurde eine Sammlung potenzieller Anforderungen der Zielgruppe an die Nutzung eines intelligenten Schachbretts erstellt, wie in Abbildung 2.1 dargestellt:

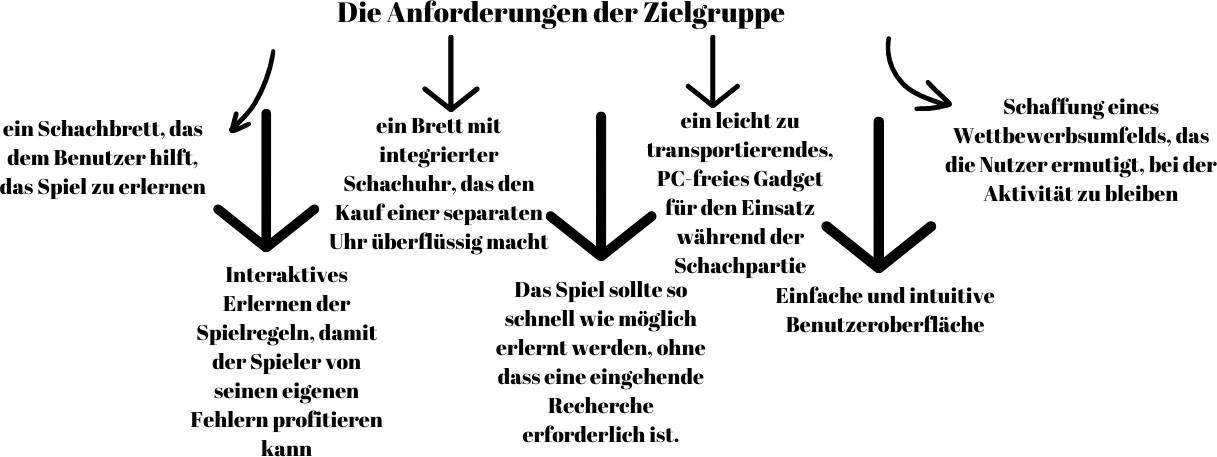


Abbildung 3. 1 Intelligent Schachbrett – Spezifikationen aus der Sicht des Nutzers

Das Verhalten des Produkts, das in der Tiefe erstellt werden muss, ist der nächste Schrittbei der Erstellung eines Bildes der Komponenten des Schachbretts.

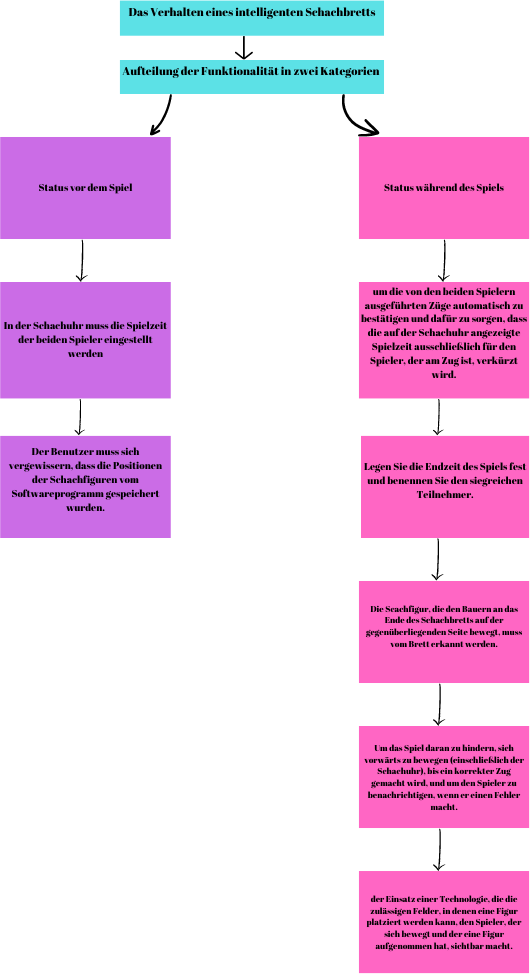


Abbildung 3. 2 Ermittlung des Verhaltens des Intelligenten Schachbretts

Alle Teilfunktionen des intelligenten Schachbretts werden in Abbildung 2.2 zusammengefasst, wobei die Hauptfunktion dieses Systems darin besteht, die Benutzer während des Spiels zu unterstützen, indem es die Spielzeit zählt und mögliche Züge der Figuren gemäß den Schachregeln vorschlägt. Da die Funktionalität komplex ist und nicht mit einer einfachen elektronischen Schaltung erreicht werden kann, ist es notwendig, die Systemaufgaben in mehrere einfache Funktionen aufzuteilen, die zusammenarbeiten, um das gewünschte Ergebnis zu erzielen (Anwendung des Prinzips

„Teile und herrsche“). Dieses Kapitel stellt die elektronischen Module und Softwarekomponenten vor, die diese Funktionen ausführen. die praktische Umsetzung dieser Aufgaben.

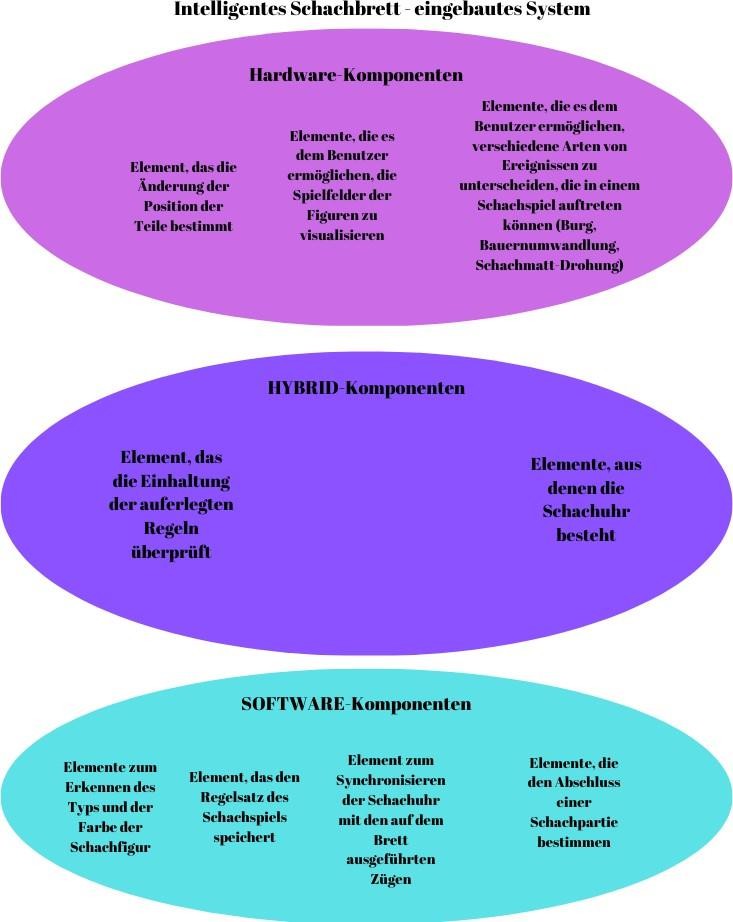


Abbildung 3. 3 Schachbrettfunktionen sind in grundlegende Funktionen unterteilt

Die Lösungen, die zur Erfüllung aller Anforderungen und Spezifikationen gewähltwurden, werden in den vorangegangenen Abschnitten erläutert.

Um dem Benutzer eine interaktive und einfache Möglichkeit zu geben, die Grundlagen des Schachspiels auf eine interaktive und einfache Weise zu lernen, wurden vierundsechzig RGBLEDs unter den 64 Feldern des Bretts platziert und verwendet, um die Felder nach den Regeln des Schachspiels auszurichten. Durch die Änderung der Farben dieser LEDs kann der Benutzer sehen, welches spezifische Ereignis des Spiels sich derzeit auf dem Schachbrett befindet und obdie erforderlichen Regeln befolgt wurden. da es sich um einen Lernprozess handelt, der auf Übung und Experimentieren mit verschiedenen Situationen basiert, die in einer Schachpartie auftreten können Da es sich um einen Lernprozess handelt, der auf dem Üben und Experimentieren mit verschiedenen Situationen basiert, die in einer Schachpartie vorkommen können, kann der Benutzer sein Wissen in viel kürzerer Zeit überprüfen, als es für die Partie erforderlich ist. Der Benutzer hätte denselben Prozess durchlaufen, indem er diese Begriffe mithilfe einer Vielzahl von Seiten untersucht hätte.

Um die tatsächlichen Positionen der Figuren zu verwalten, werden sie von zweiunddreißig RFID-Tags verfolgt, die kontinuierlich Informationen über die Platzierung der Figuren auf dem Schachbrett senden.

Firmware, die in den Flash-Speicher des Raspberry Pi Mikrocontrollers geladen wird, steuert das gesamte System. Daher übernimmt diese Anwendung die meisten Aufgaben, die mit dem Entwurf eines intelligenten Schachbretts verbunden sind, wie z. B. die Bestimmung des Typs der Figur und ihrer Farbe, die Erinnerung a die Regeln, die das Schachspiel definieren, und deren Anwendung, die Synchronisierung der Schachuhr mit den Zügen der Figuren auf dem Brett, die Einstellung der beiden Zustände des oben erwähnten intelligenten Schachbretts und das Einstellen der auf dem Brett, die Figuren, die auf dem Schachbrett erscheinen.

Dank des LCD16x2-Displays, das die Spielzeit der beiden Spieler auf einer achtstelligen Sieben-Segment-Anzeige darstellt, ist es möglich, die Schachuhr in das Spielbrett zu integrieren. Die Uhrfunktionalität ist in der Firmware-Anwendung implementiert.

Die Benutzeranforderungen und vorgeschlagene Lösungen wurden in

diesem Kapitel zusammengefasst, während die Hardware-, Software- und physikalischen Implementierungen inden Kapiteln 3, 4 und 5 detailliert behandelt wurden.

# Hardwareaufbau des Schachbretts

## Blockschaltbild des Systems

Abbildung 4.1 zeigt das Blockdiagramm, das für die Hardware-Implementierung desintelligenten Schachbretts verwendet wurde.

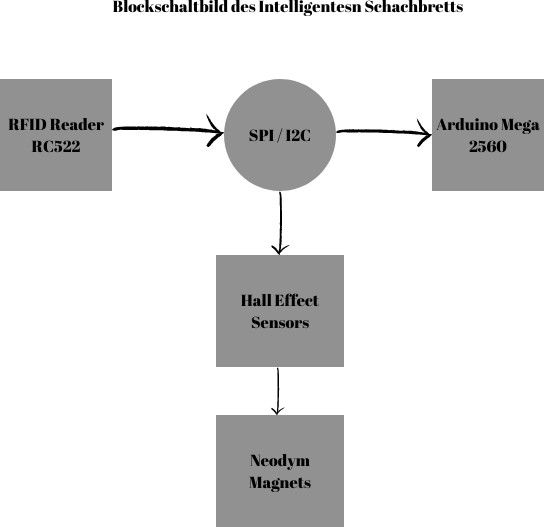


Abbildung 4. 1 Blockschaltbild des intelligenten Schachbretts

Das Blockdiagramm zeigt die Module des eingebetteten Systems sowie die wichtigsten Signale, die seine Zusammenarbeit bestimmen. RFID-Lesegeräte und Reed- Schalter bilden den Eingabeblock und die Benutzeroberfläche des Systems. Die Entwicklungsplatine steuert den Verarbeitungs- und Steuerblock, der auf der Grundlage der empfangenen Daten entscheiden, Steuersignale einschalten, die richtigen RGB-LED-Farben auswählen und die Segmente auf dem 16x2-LCD-Display darstellen muss. Der Ausgangsblock besteht aus RGB-LEDs, die dem Benutzer während eines Zuges die zulässigen Felder anzeigen, und eine 7 -Segment-Anzeige

zeigt die verbleibende Spielzeit für jeden Spieler an.

In den folgenden Abschnitten dieses Kapitels finden Sie weitere Informationen zum Funktionsprinzip der Module und zur Verarbeitung der Daten.

## Der Eingangsblock

## Kurze Vorstellung des RFID-Readers

Der MFRC522 ist ein hochintegriertes Lese-/Schreib-IC für die kontaktlose Kommunikation bei 13,56 MHz. Der MFRC522-Leser unterstützt ISO/IEC 14443A/MIFARE und NTAG. [4]

Wie in der folgenden Abbildung dargestellt, soll das RFID-Lesegerät in eine integrierteSchaltung mit 23 Pins integriert werden:

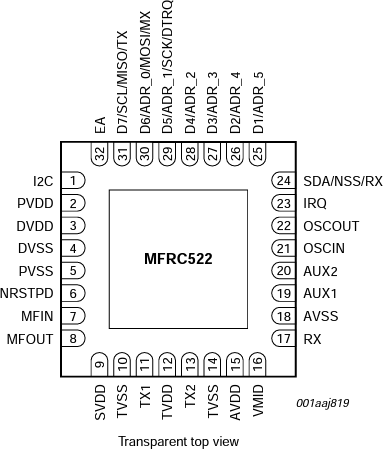


Abbildung 4. 2 Stiftstruktur des RFID-Lesegeräts. Quelle: [4]

Tabelle 4. 1 Stiftstruktur des RFID-Lesegeräts. Quelle: [4]

|  |  |  |
| --- | --- | --- |
| Stift | Zeichen | Description |
| 1 | I2C | I2C-Bus-Freigabe-Eingang |
| 2 | PVDD | Pin-Stromversorgung |
| 3 | DVDD | digitale Energieversorgung |
| 4 | DVSS | digitale Masse |
| 5 | PVSS | Pin Spannungsversorgung Masse |
| 6 | NRSTPD | Reset- und Power-Down-Eingang:   * Power-Down: aktiviert, wenn LOW; interne Stromsenken sind abgeschaltet, der Oszillator ist gesperrt und die Eingangspins sind von der Außenwelt abgeschaltet * Reset: wird durch eine positive Flanke aktiviert |
| 7 | MFIN | MIFARE-Signaleingang |
| 8 | MFOUT | MIFARE-Signaleingang |
| 9 | SVDD | Stromversorgung der MFIN- und MFOUT-Pins |
| 10 | TVSS | Sender Ausgangsstufe 1 Masse |
| 11 | TX1 | Sender 1 modulierter 13,56MHz Energieträgerausgang |
| 12 | TVDD | Transmitter-Stromversorgung: versorgt die Endstufe der Transmitter 1 und 2 |
| 13 | TX2 | Sender 2 modulierter 13,56MHz Energieträgerausgang |
| 14 | TVSS | Senderausgangsstufe 2 Masse |
| 15 | AVDD | Analogstromversorgung |
| 16 | VDIM | interne Referenzspannung |
| 17 | RX | RF-Signaleingang |

|  |  |  |
| --- | --- | --- |
| 18 | AVSS | analoge Masse |
| 19 | AUX1 | Hilfsausgänge für Testzwecke |
| 20 | AUX2 | Hilfsausgänge für Testzwecke |
| 21 | OSCIN | Quarzoszillator- Invertierungsverstärkereingang; auch der Eingang für einen extern erzeugten  Takt (fclk=27,12MHz) |

|  |  |  |
| --- | --- | --- |
| 22 | OSCOUT | Quarzoszillator Invertierungsverstärkerausgang |
| 23 | IRQ | Interrupt-Anforderungsausgang: zeigt  ein Interrupt-Ereignis an |
| 24 | SDA | I2C-Bus serielle Datenleitung Eingang/Ausgang |
| NSS | SPI-Signaleingang |
| RX | UART-Adresseingang |
| 25 | D1 | Prüfanschluss |
| ADR\_5 | I2C-Bus-Adresse 5 Eingang |
| 26 | D2 | Prüfanschluss |
| ADR\_4 | I2C-Bus-Adresse 4 Eingang |
| 27 | D3 | Prüfanschluss |
| ADR\_3 | I2C-Bus-Adresse 3 Eingang |
| 28 | D4 | Prüfanschluss |
| ADR\_2 | I2C-Bus-Adresse 2 Eingang |
| 29 | D5 | Prüfanschluss |
| ADR\_1 | I2C-Bus-Adresse 1 Eingang |
| SCK | SPI serieller Takteingang |
| DTRQ | UART-Anforderung zum Senden der  Ausgabe an den Mikrocontroller |
| 30 | D6 | Prüfanschluss |
| ADR\_0 | I2C-Bus-Adresse 0 Eingang |
| MOSI | SPI-Master raus, Slave rein |
| MX | UART-Ausgang zum Mikrocontroller |
| 31 | D7 | Prüfanschluss |
| SCL | I2C-Bus-Takteingang/-ausgang |
| MISO | SPI-Master ein, Slave aus |
| TX | UART-Datenausgabe an Mikrocontroller |
| 32 | EA | externer Adresseneingang zur Kodierung der I2C-Bus-Adresse |

Die wichtigsten Merkmale, die zur Wahl des RFID-Lesegeräts RC522 geführt haben, sinddie folgenden:

Tabelle 4. 2 Hauptmerkmale des RFID-Lesegeräts. Quelle: [4]

|  |  |  |
| --- | --- | --- |
| Parameter | MIN | MAX |
| VDDA (analoge  Versorgungsspannung) | 2.5V | 3.6V |
| VDDD (digitale  Versorgungsspannung) | 2.5V | 3.6V |
| VDD(TVDD) (TVDD-  Versorgungsspannung) | 2.5V | 3.6V |
| VDD(PVDD) (PVDD  Versorgungsspannung) | 1.6V | 3.6V |
| VDD(SVDD) (SVDD  Versorgungsspannung) | 1.6V | 3.6V |
| Ipd (Abschaltstrom) | - | 5 uA |
| - | 10 uA |
| IDDD (digitaler Versorgungsstrom) | - | 9 mA |
| IDDA (Analoger  Versorgungsstrom) | - | 10 mA |
| - | 5 mA |
| IDD(PVDD) (PVDD-  Versorgungsstrom) | - | 40 mA |
| IDD(TVDD) (TVDD-  Versorgungsstrom) | - | 100 mA |
| Tamb  (Umgebungstemperatur) | -25°C | +85°C |

Der interne Sender des MFRC522 ist in der Lage, eine Lese-

/Schreibantenne anzusteuern, die für die Kommunikation mit ISO/IEC14443A/MIFARE-Karten und - Transpondern ohne zusätzliche aktive Schaltungen ausgelegt ist. Das Empfängermodulbietet eine robuste und effiziente Implementierung zur Demodulation und Dekodierungder Signale von ISO/IEC14443A/MIFARE-kompatiblen Karten und Transpondern. Dasdigitale Modul verwaltet die komplette ISO/IEC14443A Framing- und Fehlererkennungsfunktionalität (Parität und CRC).

Der MFRC522 unterstützt die Produkte MF1xxS20, MF1xxS70 und MF1xxS50.

Der MFRC522 unterstützt die kontaktlose Kommunikation und nutzt die höheren MIFARE-Übertragungsgeschwindigkeiten von bis zu 848kBd in

beiden Richtungen. [4]

## Die Rolle von RFID - Lesegeräten und - Etiketten im System

Ein erster Schritt bei der Schaffung eines intelligenten Schachbretts besteht darin, einen Mechanismus zu wählen, der eine Reihe von Daten auf der Grundlage externer Aktionen erkenntund die Kompatibilität mit den anderen mit den anderen elektronischen Komponenten überwacht. Das RFID-Lesegerät RC522 wurde aufgrund dieser Aussagen und der Analyse der Merkmale in Tabelle 4.2 ausgewählt.

In dieser Arbeit wurden zweiunddreißig RFID-Etiketten verwendet, deren Anzahl sich nach derAnzahl der Schachfiguren und einem RFID-Lesegerät richtete.

RFID-Etiketten liefern Informationen über das Vorhandensein oder Nichtvorhandensein einer Schachfigur auf einem bestimmten Feld. In Kapitel 5 wird gezeigt, wie die Figuren mit Hilfe von Softwareverfahren nach ihrem Typ (z.B. Springer, Läufer, König) und ihrer Farbe (z.B.weiß, schwarz) unterschieden werden können. Alle diese Informationen bilden die Eingabedaten des Systems, die mit Hilfe der im weiteren Verlauf dieses Kapitels vorgestellten Geräte verarbeitet werden, wobei die Verarbeitung und Kontrolle der Ausgabedaten zu den Aufgaben der Firmware gehören, deren detaillierte Beschreibung eines der Themen von Kapitel 5 ist.

RFID-Tags sind kleine elektronische Geräte, die eindeutige Identifizierungsinformationen enthalten. Im Zusammenhang mit dem intelligenten Schachbrett sind RFID-Tags in die Schachfiguren eingebettet. Jede Schachfigur ist mit einem eindeutigen RFID-Tag versehen, der sie identifiziert. Das RFID-Lesegerät liest diese Tags, wenn die Schachfiguren auf dem Brett platziert oder bewegt werden. Die Identifikationsinformationen der Tags werden an den Arduino Mega 2560 - Mikrocontroller gesendet, so dass dieser die Positionen der Schachfiguren auf dem Brett genau bestimmen kann.



Abbildung 4. 3 RFID Tag. Quelle: [5]

Das Schachbrett ist außerdem mit einem RFID-Lesegerät ausgestattet. Das RFID- Lesegerät ist ein mit dem Netz verbundenes Gerät, das tragbar oder fest angebracht sein kann. Es nutzt Funkwellen zur Übertragung von Signalen, die das Etikett aktivieren. Nach der Aktivierungsendet das Etikett eine Welle zurück an die Antenne, wo sie in Daten umgewandelt wird.

Der Transponder befindet sich im RFID-Tag selbst. Die Lesereichweite von RFID- Etiketten hängt von verschiedenen Faktoren ab, u. a. vom Typ des Etiketts, vom Typ des Lesegeräts, von der RFID-Frequenz und von Störungen in der Umgebung oder durch andere RFID-Etiketten und Lesegeräte. Tags, die über eine stärkere Stromquelle verfügen, haben aucheine größere Lesereichweite. [6]

Das RFID-Lesegerät, z. B. das RC522-Modul, wird zum Lesen der in die Schachfiguren eingebetteten RFID-Tags verwendet. Es kommuniziert mit dem Arduino Mega 2560 Mikrocontroller entweder über das SPI (Serial Peripheral Interface) oder das I2C (Inter-Integrated Circuit) Protokoll. Das RFID-Lesegerät sendet Funkwellen aus und erkennt die Antwort der RFID-Tags auf den Schachfiguren. Durch das Lesen der eindeutigen Identifikationsinformationen, die in den RFID-Tags gespeichert sind, identifiziert das Lesegerät die Schachfiguren und leitet diese Informationen zur weiteren Verarbeitung an den Arduino Mega 2560 weiter.



Abbildung 4. 4 RFID-Lesegerät RC522. Quelle: [7]

## Verarbeitungs- und Kontrollblock

## Kurze Vorstellung des Arduino Mega 2560 Mikrocontroller - Entwicklungsboards

Der Arduino Mega 2560 ist ein Mikrocontroller-Board auf Basis des ATmega2560. Es verfügt über 54 digitale Eingangs-/Ausgangs-Pins (von denen 15 als PWM-Ausgänge verwendet werden können), 16 analoge Eingänge, 4 UARTs (serielle Hardware-Ports), einen 16-MHz-Quarzoszillator, einen USB-Anschluss, eine Stromversorgungsbuchse, einen ICSP-Header und eine Reset-Taste. [12]

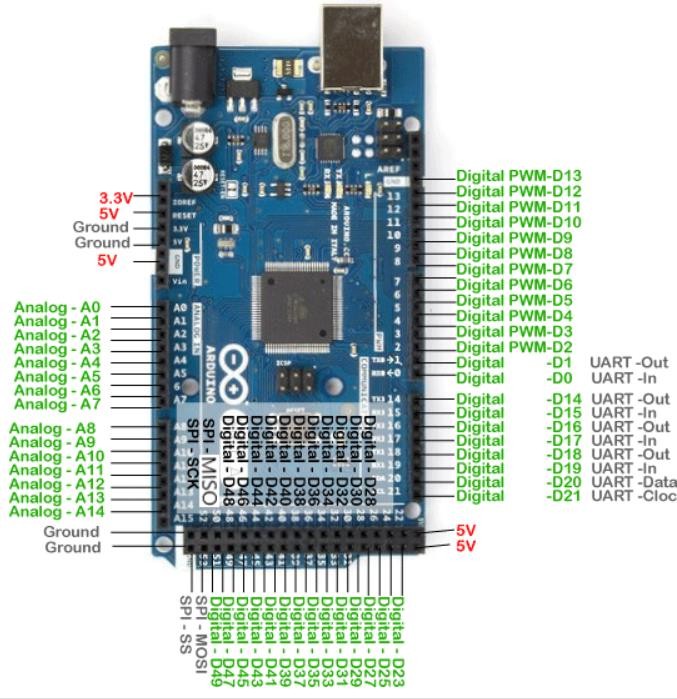


Abbildung 4. 5 Arduino Mega 2560 Pinout. Quelle: [8]

Der Arduino Mega 2560 ist ein Mikrocontroller-Board, das auf dem ATmega2560- Mikrocontroller basiert. Es ist eine verbesserte Version des beliebten Arduino Uno-Boards und bietet eine größere Anzahl von Eingangs-/Ausgangs-Pins und Speicher. Hier sind einige wichtige Details über den Arduino Mega 2560:

1. Mikrocontroller: Der Arduino Mega 2560 verwendet den ATmega2560-Mikrocontroller, einen 8- Bit-AVR-Mikrocontroller mit 256 KB Flash-Speicher für die Speicherung des Programmcodes, 8 KB SRAM für die Datenspeicherung und 4 KB EEPROM für die nichtflüchtige Speicherung.
2. Taktrate: Der Mikrocontroller des Arduino Mega 2560 arbeitet mit einer Taktrate von 16 MHz. Mit dieser Frequenz führt er Anweisungen aus und führt Operationen durch.
3. Digitale E/A-Anschlüsse: Das Mega 2560-Board verfügt über insgesamt 54 digitale Eingangs-

/Ausgangsstifte, von denen 15 als PWM-Ausgänge (Pulsweitenmodulation) verwendet werden können. Diese Pins ermöglichen den Anschluss verschiedener elektronischer Komponenten wie Sensoren, Aktuatoren, Anzeigen und mehr.

1. Analoge Eingänge: Der Mega 2560 verfügt über 16 analoge Eingangsstifte, die mit A0 bis A15 bezeichnet sind. Diese Pins können zum Lesen analoger Spannungspegel von Sensoren oder anderen analogen Geräten verwendet werden.
2. Kommunikationsschnittstellen: Die Karte bietet mehrere Kommunikationsschnittstellen, darunter vier serielle Hardware-UART-Anschlüsse (Universal Asynchronous Receiver/Transmitter), einen USB-Anschluss für serielle Kommunikation und Programmierung, einen SPI-Anschluss (Serial Peripheral Interface), einen I2C-Bus (Inter-Integrated Circuit) und einen CAN-Bus (Controller Area Network).
3. Speicher: Wie bereits erwähnt, verfügt der Arduino Mega 2560 über 256 KB Flash-Speicher, der zum Speichern des Programmcodes verwendet wird. Er verfügt über 8 KB SRAM, der für die Speicherung von Laufzeitdaten und Variablen verwendet wird, und 4 KB EEPROM, der eine nichtflüchtige Speicherung von Daten ermöglicht, die auch nach dem Ausschalten der Stromversorgung erhalten bleibt.
4. Stromversorgung: Der Mega 2560 kann über einen USB-Anschluss oder eine externe Stromquelle mit Strom versorgt werden. Das Board akzeptiert einen Spannungsbereich von 7-12 Volt für die externe Stromversorgung.
5. Entwicklungsumgebung: Arduino-Boards, einschließlich des Mega 2560, werden mit der Arduino- Software (IDE) programmiert. Die IDE bietet eine einfach zu bedienende Schnittstelle zum Schreiben,

Kompilieren und Hochladen von Code auf das Board. Sie unterstützt die Programmiersprachen C und C++.

## Die Rolle des Arduino Mega 2560 - Mikrocontrollers bei der Datenverarbeitung und Systemsteuerung

Der Arduino Mega 2560 kann eine wichtige Rolle bei der Realisierung eines intelligenten Schachbretts spielen, indem er die notwendige Hardware und Programmierfähigkeiten bereitstellt. Hier sind einige Möglichkeiten, wie der Arduino Mega 2560 für diesen Zweck genutzt werden kann:

1. Erfassen des Schachbretts: Der Arduino Mega 2560 kann an eine Matrix von Sensoren angeschlossen werden, wie z. B. Hall-Effekt-Sensoren oder kapazitive Berührungssensoren, die unter jedem Schachbrettfeld angebracht werden. Diese Sensoren können das Vorhandensein und die Bewegung von Schachfiguren auf dem Brett erkennen. Die zahlreichen digitalen und analogen Eingangspins des Mega 2560 können verwendet werden, um mit diesen Sensoren zu kommunizieren und Daten von ihnen zu lesen.
2. Verbindung mit Schachfiguren: Der Arduino Mega 2560 kann elektronische Schachfiguren mit Hilfe von Aktuatoren wie Solenoiden oder Servomotoren steuern. Indem er Eingaben von einem Computer oder einem Spieler empfängt, kann der Arduino den entsprechenden Aktuatoren den Befehl geben, die Schachfiguren auf dem Brett zu bewegen.
3. Kommunikation: Der Mega 2560 verfügt über mehrere Kommunikationsschnittstellen, darunter UART, SPI und I2C. Diese Schnittstellen können verwendet werden, um die Kommunikation zwischen dem Arduino-Board und einem externen Computer oder anderen Geräten herzustellen. Zum Beispiel kann der Arduino mit einem Computer kommunizieren, auf dem eine Schachspielsoftware läuft, um Zugvorschläge zu erhalten oder Spielinformationen zu senden.
4. Schachlogik und KI: Der Arduino Mega 2560 ist zwar ein 8-Bit-Mikrocontroller, kann aber dennoch einfache Schachlogik und KI-Algorithmen implementieren. Er hat zwar nicht die Rechenleistung, um selbst fortgeschrittene Schachalgorithmen auszuführen, aber er kann bei der grundlegenden Zugüberprüfung und der Verwaltung des Spielzustands helfen. Er kann auch mit externen Geräten wie einem Raspberry Pi oder einem Computer verbunden werden, die komplexere Schachberechnungen und KI durchführen können.
5. Benutzerschnittstelle: Der Arduino Mega 2560 kann mit verschiedenen Anzeigemodulen, wie

LCD-Bildschirmen oder LED-Matrizen, verbunden werden, um eine visuelle Darstellung des Schachbretts und des Spielzustands zu liefern. Er kann auch Knöpfe oder einen Touchscreen für Benutzereingaben enthalten, so dass die Spieler mit dem Brett interagieren können.

1. Energieverwaltung: Der Mega 2560 kann Aufgaben der Energieverwaltung übernehmen, wie

z. B. die Überwachung der Stromversorgung, die Steuerung des Stromverbrauchs und die Steuerung des Betriebs des Boards auf der Grundlage der Energiebedingungen.

Insgesamt fungiert der Arduino Mega 2560 als zentrale Steuereinheit für ein intelligentes Schachbrett, das mit Sensoren, Aktoren, Kommunikationsmodulen und Benutzerschnittstellen verbunden ist. Er ermöglicht die Implementierung von Schachlogik, die Kommunikation mit externen Geräten und bietet eine Plattform für die Integration verschiedener Hardwarekomponenten, um ein interaktives und intelligentes Schachspielerlebnis zu schaffen.

## SPI-Protokoll

SPI ist ein synchrones Vollduplex-Kommunikationsprotokoll (Geräte senden und empfangen gleichzeitig Daten). Daten gleichzeitig senden und empfangen), das von der FirmaMotorola eingeführt wurde, um eine nahtlose Daten (bei Frequenzen bis zu einigen zehn MHz[9]) zwischen mehreren Geräten.

Um die Datenübertragung zu erreichen, führt das SPI-Protokoll vier zusätzliche Drähte indie Schaltung ein. Die Klemmen, an denen diese Drähte angeschlossen sind, haben die folgendenstandardisierten Namen:

* + - * SCLK ("Serieller Takt") ist der Pin, an dem das Taktsignal zur Datensynchronisation verwendet wird;
      * MISO ("Master Input - Slave Output") ist der Pin, an dem Datensignale vom Slave"-Gerät an das "Master"-Gerät übertragen werden;
      * MOSI (Master Output - Slave Input) ist der Pin, auf dem Datensignale vom "Master"-Gerät zum "Master"-Gerät übertragen werden. Master"- Gerät an das "Slave"-Gerät übertragen werden;
      * SS ("Slave Select") ist der Pin, auf dem das Select-Signal an das

"Slave"-Gerät übertragen wird. Slave-Gerät übertragen wird, mit dem die Kommunikation aufgenommen werden soll.

Als "Single-Master-Multi-Slave"-Protokoll ermöglicht SPI die Kommunikation zwischeneiner einer beliebigen Anzahl von Geräten, vorausgesetzt, nur ein Gerät wird zu einem bestimmten Zeitpunkt Zeitpunkt zum "Master" erklärt wird. Zwei konkrete Beispiele für die Zusammenschaltung von Geräten sind in Abbildung 3.10.

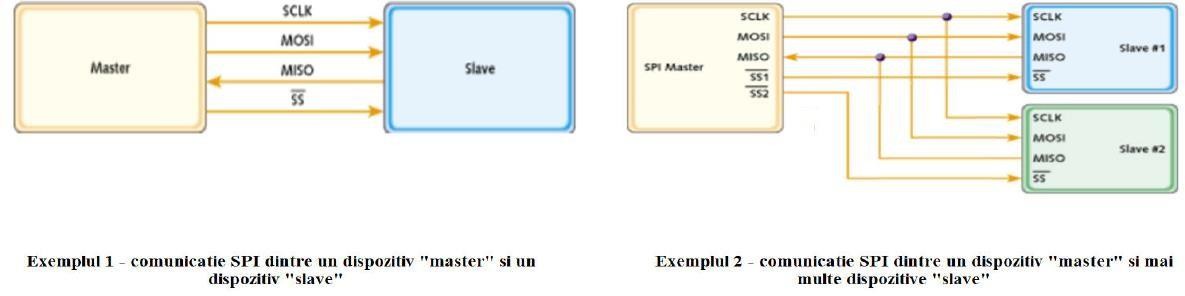


Abbildung 4. 6 Anschluss von SPI-Protokoll-spezifischen Pins. Quelle: [9]

In dieser Arbeit wurde das Verbindungsschema in Beispiel 2 der obigen Abbildung verwendet, wobei der Arduino Mega 2560 Mikrocontroller das "Master"-Gerät und das RFID-Lesegerät RC522 die "Slave"-Geräte sind.

Um die Kommunikation über SPI zu initiieren, muss das Master-Gerät die folgenden Parameter einstellen Taktsignalparameter am SCLK-Pin, wobei der SS-Pin auf die logische '1' gesetzt wird. Außerdem müssen die "Master"- und "Slave"-Geräte müssen die zu übertragenden Daten in die Schieberegister der SPI-Schnittstelle laden. Der Beginnder Kommunikation wird durch das Setzen der logischen '0' am SS-Pin durch das Der Start der Kommunikation erfolgt durch Setzen der logischen "0" am SS-Pin durch das "Master"- Gerät, wobei die Datenübertragung in Gruppen von acht Bits auf den MOSI- und MISO- PINS. Nach der Übertragung einer oder mehrerer 8-Bit-Gruppen muss das Master-Gerät die logische "1" auf den SS-Pin, was den Abschluss der Kommunikation zwischen den Geräten anzeigt. Alle diese sind in Abbildung 4.8 dargestellt.

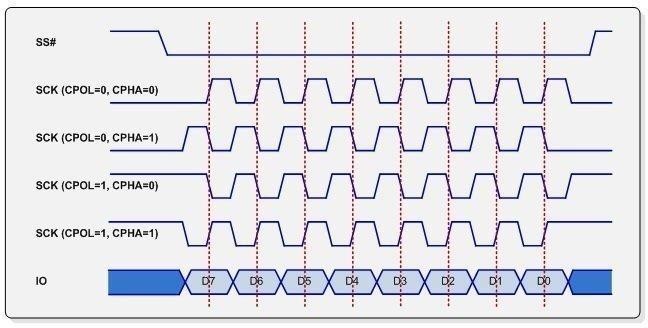


Abbildung 4. 7 Übertragung eines Datenpakets über das SPI-Protokoll. Quelle: [10]

Abbildung 4.8 zeigt zusätzlich zu den oben genannten Informationen das Vorhandensein von zusätzlichen Bits, nämlich CPOL und CPHA. Abhängig von den Werten dieser Bits werden die endgültigen Einzelheiten festgelegt, und zwar in Bezug auf die Datenübertragung, wobei diegestrichelten Linien die aktiven Flanken des Taktsignals in den vier Betriebsmodi darstellen. So legt CPOL die Polarität des Taktsignals fest (bei CPOL = 0 beginnt das Taktsignal mit der logischen '0', und bei CPOL = 1 beginnt das Taktsignal 1') und CPHA legt die Phase des Taktsignals fest (bei CPHA = 0 hat das Taktsignal hat das Taktsignal eine Phasenverschiebungvon Null, und bei CPHA = 1 beträgt die Phasenverschiebung 180º). Die Bits CPOL und CPHA befinden sich in einem internen Steuerregister der SPI-Schnittstelle und werden vom Benutzer gesetzt.

Um die SPI-Kommunikation in dieser Arbeit zu realisieren, werden die folgenden Schnittstellenverwendet SPI-Schnittstellen der Module, wobei die Konfiguration der Register zur Erzeugungdes Taktsignals bzw. der Signale an den SS-Pins, die von der in Kapitel 5 vorgestellten Firmware-Anwendung ausgeführt werden.

## Die Verwendung von LEDs bei der Erkennung von Schachfiguren

LEDs können bei der Erstellung eines intelligenten Schachbretts verwendet werden, umdie Erkennung der Schachfiguren zu unterstützen. Hier ist eine Erklärung ihrer Rolle:

1. Erkennung von Schachfiguren: LEDs können unter jedem Feld des Schachbretts angebrachtwerden, ausgerichtet auf die Mitte des Feldes. Wenn eine Schachfigur auf einem Feld platziertwird, behindert die Figur das Licht, das von der LED unter der Figur ausgesendet wird. Durch Überwachung der Lichtintensität, die von Lichtsensoren oder Fotodetektoren auf dem Schachbrett empfangen wird, kann das System das Vorhandensein oder Fehlen von Schachfiguren auf jedem Feld erkennen.
2. Lichtsensorik: Lichtsensoren oder Photodetektoren, die auf dem Schachbrett angebracht sind, sind für die Erkennung des von den LEDs ausgesandten Lichts verantwortlich. Diese Sensorenmessen die Intensität des empfangenen Lichts und wandeln es in elektrische Signale um. Durch die Überwachung der Änderungen der Lichtintensität kann das System feststellen, ob eine Schachfigur vorhanden ist oder bewegt wurde.
3. Schachfigurenverfolgung: Die LEDs und Lichtsensoren ermöglichen es dem intelligenten Schachbrett, die Bewegung von Schachfiguren zu verfolgen. Wenn eine Schachfigur von einem Feld auf ein anderes bewegt wird, kann das System die Änderung der Lichtintensität erkennen,wenn die LED unter dem Ausgangsfeld nicht mehr verdeckt ist, während die LED unter dem Zielfeld verdeckt wird. Diese Information hilft dem System, den Zustand des Schachbretts zu aktualisieren und die Positionen der Schachfiguren genau zu verfolgen.
4. Aktualisierungen in Echtzeit: Durch die Verwendung von LEDs und Lichtsensoren kann das intelligente Schachbrett Echtzeit-Updates des Schachbrettstatus liefern. Wenn die Schachfigurenbewegt werden, kann das System die Änderungen schnell erkennen

und sie in der digitalen Darstellung des Schachbretts wiedergeben. Dadurch wird sichergestellt, dass der angezeigte Zustand des Brettes mit den physischen Positionen der Schachfiguren synchronisiert bleibt.

Die Verwendung von LEDs in Verbindung mit Lichtsensoren ermöglicht eine zuverlässige und genaue Erkennung der Schachfiguren auf dem intelligenten Schachbrett. Durch die Überwachung der Lichtintensität kann das System die Bewegung der Schachfiguren verfolgen und den Zustand des Schachbretts in Echtzeit aktualisieren, wodurch die Gesamtfunktionalität und Interaktivität des Schachbretts verbessert wird.

## Eine kurze Einführung in Hall-Effekt-Sensoren

Hall-Effekt-Sensoren sind elektronische Geräte, die den Hall-Effekt nutzen, d.

h. die Erzeugung einer Spannungsdifferenz senkrecht zur Richtung eines elektrischen Stroms, der durch einen Leiter fließt, wenn ein Magnetfeld vorhanden ist. Hier sind einige wichtige Punkte über Hall-Effekt-Sensoren:

Arbeitsprinzip: Hall-Effekt-Sensoren bestehen in der Regel aus einem dünnen Halbleitermaterial wie Galliumarsenid oder Indiumantimonid. Wenn ein Magnetfeld senkrecht zu dem durch den Sensor fließenden Strom angelegt wird, entsteht eine Lorentz-Kraft, die die Ladungsträger (Elektronen oder Löcher) auf eine Seite des Materials drückt. Diese Ladungsanhäufung erzeugt eine Spannungsdifferenz über dem Sensor, die als Hall-Spannung bezeichnet wird.

Arten von Hall-Effekt-Sensoren: Es gibt drei Haupttypen von Hall-Effekt- Sensoren:

1. Hall-Effekt-Schalter: Diese Sensoren werden verwendet, um das Vorhandensein oder Nichtvorhandensein eines Magnetfeldes zu erkennen. Sie haben einen digitalen Ausgang, der seinen Zustand ändert, wenn ein Magnetfeld einen bestimmten Schwellenwert überschreitet. Hall-Effekt-Schalter werden häufig in Anwendungen wie Näherungssensorik, Geschwindigkeitserfassung und Positionserfassung eingesetzt.
2. Hall-Effekt-Speicher: Latch-Sensoren behalten ihren Ausgangszustand bei, auch nachdem das Magnetfeld entfernt wurde. Sie liefern einen digitalen Ausgang, der stabil bleibt, bis ein umgekehrtes Magnetfeld angelegt wird, um sie

zurückzusetzen. Diese Sensoren eignen sich für Anwendungen, die einen stabilen Zustand erfordern, wie z. B. Tür-/Fenstersensoren, Motorsteuerung und Sicherheitssysteme.

1. Lineare Hall-Effekt-Sensoren: Lineare Sensoren liefern einen analogen Ausgang, der proportional zur Stärke des Magnetfelds ist. Sie werden häufig für Positions- und Verschiebungsmessungen verwendet, z. B. in Drosselklappen- Positionssensoren für Kraftfahrzeuge, industriellen Steuersystemen und in der Robotik.

## Verwendung von Hall-Effekt-Sensoren im System

Hall-Effekt-Sensoren können in der Tat für die Realisierung eines intelligenten Schachbretts verwendet werden. Der Hall-Effekt ist die Erzeugung einer Spannungsdifferenz über einem elektrischen Leiter, wenn dieser sich in einem Magnetfeld befindet und einen elektrischen Strom führt. Im Folgenden wird erläutert, wie Hall-Effekt-Sensoren in ein intelligentes Schachbrett integriert werden können:

* + - * Erkennung der Figuren: Jedes Feld des Schachbretts kann mit einem Hall-Effekt-Sensor unter dem Feld ausgestattet werden. Schachfiguren können mit kleinen Magneten versehen werden. Wenn eine Schachfigur auf ein Feld gesetzt wird, interagiert das vom Magneten erzeugte Magnetfeld mit dem Hall-Effekt-Sensor und erzeugt eine Spannungsänderung. Durch die Überwachung der Spannungsänderungen in den Sensoren kann das Schachbrett das Vorhandensein und die Position jeder Figur erkennen.
      * Identifizierung der Figuren: Verschiedene Figuren können mit Magneten unterschiedlicher Stärke oder Anordnung ausgestattet werden. Dadurch können die Hall-Effekt-Sensoren zwischen verschiedenen Arten von Schachfiguren unterscheiden. Die Sensoren können zum Beispiel die magnetische Signatur eines Bauern, Turms, Springers, Läufers, einer Dame oder eines Königs erkennen.
      * Zugverfolgung: Im Laufe des Spiels kann das intelligente Schachbrett die Bewegung der Figuren verfolgen. Jedes Mal, wenn eine Figur bewegt wird, erkennt der entsprechende Hall- Effekt-Sensor die Positionsänderung und ermöglicht es dem System, den Zustand des Brettes zu aktualisieren.
      * Spiellogik und Unterstützung: Durch die Kombination der Fähigkeiten zur Erkennung und

Identifizierung von Figuren kann das intelligente Schachbrett den aktuellen Spielstand analysieren und verschiedene Formen der Unterstützung anbieten. Dazu gehören das Hervorheben zulässiger Züge, das Vorschlagen optimaler Züge, das Erkennen von Schachmatt- oder Patt-Situationen und das Bereitstellen interaktiver Anleitungen für Spieler verschiedener Spielstärken.

* + - * Konnektivität und Integration: Das intelligente Schachbrett kann mit einem Computer oder einem mobilen Gerät verbunden werden, so dass es mit Schachspielsoftware, Online- Datenbanken oder KI-Algorithmen interagieren kann. Diese Integration eröffnet Möglichkeiten für erweiterte Analysen, Spielstatistiken und sogar Multiplayer-Funktionen.
      * Durch den Einsatz von Hall-Effekt-Sensoren kann ein intelligentes Schachbrett den Spielern ein einzigartiges und interaktives Erlebnis bieten, indem es Echtzeit-Feedback und Unterstützung bietet und gleichzeitig den traditionellen Reiz des Spiels beibehält.

# Software-Struktur des Schachbretts

## Darstellung der Softwarestruktur

In diesem Kapitel stellen wir die Firmware-Anwendung vor, die für das SchachbrettSchachbrett. Die Firmware ist die Softwarekomponente des Systems und spielt eine Schlüsselrolle, denn sie muss die schwierigsten Aufgaben lösen, um die gewünschte Funktionalität zu erreichen.

In dieser Arbeit wird die Arduino IDE für die Programmierung verwendet. Die integrierte Entwicklungsumgebung von Arduino - oder Arduino-Software (IDE) - enthält einen Texteditor zum Schreiben von Code, einen Nachrichtenbereich, eine Textkonsole, eine Symbolleiste mit Schaltflächen für allgemeine Funktionen und eine Reihe von Menüs. Sie stellt eine Verbindung zur Arduino-Hardware her, um Programme hochzuladen und mit ihr zu kommunizieren.

## Codestruktur für die Modulkonfiguration und -implementierung Kommunikationsprotokolle

## Erkennung des Bauteiltyps durch Softwareverfahren

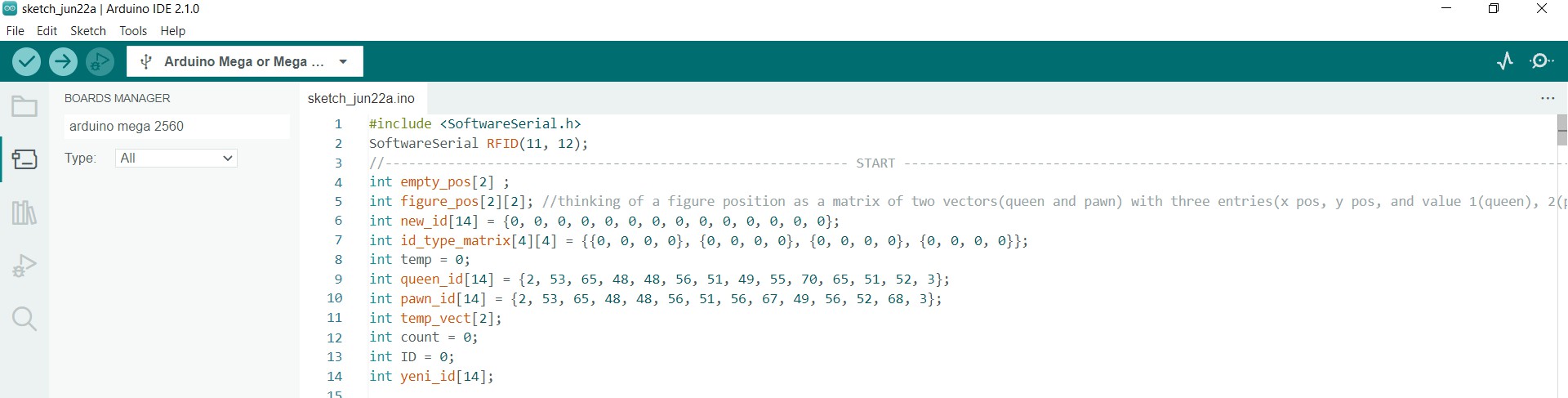
In diesem Unterkapitel wird der Software-Ansatz für diese Arbeit vorgestellt, indem einige der im Code verwendeten Funktionen und Konstanten erläutert werden.

Abbildung 5. 1 Software-Einführung von Schachfiguren. Quelle: [11]

Die ersten beiden Zeilen binden die SoftwareSerial-Bibliothek ein und erstellen eine Instanz namens RFID, die die Pins 11 und 12 verwendet. Dies wird normalerweise für die serielle Kommunikation mit einem RFID-Modul verwendet.

**empty\_pos[2]** - Dies deklariert ein Array namens empty\_pos, das zwei Integer-Werte speichern kann. Es wird wahrscheinlich verwendet, um die Koordinaten einer leeren Position auf dem Schachbrett zu speichern.

**figure\_pos[2][2]** - Dies deklariert ein zweidimensionales Array namens figure\_pos, das die Positionen von Schachfiguren speichern kann. Es hat zwei Zeilen, die jeweils zwei ganze Zahlen enthalten. Die erste Zeile ist für die Stellung der Dame und die zweite Zeile für die Stellung des Bauern.

**int new\_id[14]** - Damit wird ein Array namens new\_id mit 14 Elementen initialisiert, die alle auf 0 gesetzt sind. Es wird verwendet, um eine neue Identifikationsnummer für Schachfiguren zu speichern.

**int id\_type\_matrix[4][4]** - Dies deklariert ein zweidimensionales Array namens id\_type\_matrix mit 4 Zeilen und 4 Spalten. Es wird verwendet, um den Typ der Schachfigur (Dame oder Bauer) auf der Grundlage ihrer Position zu speichern.

**Int temp** - Dies deklariert eine Integer-Variable namens temp und initialisiert sie mit dem Wert 0. Sie wird als temporäre Variable zur Speicherung von Zwischenwerten während der Berechnungen verwendet.

**Int queen\_id** - Dadurch wird ein Array namens queen\_id mit 14 Elementen initialisiert, die eine eindeutige Identifikationsnummer für eine Schachfigur darstellen.

**Int pawn\_id** - Dadurch wird ein Array namens pawn\_id mit 14 Elementen initialisiert, die eine eindeutige Identifikationsnummer für eine Schachfigur darstellen.

**Int temp\_vect** - Damit wird ein Array namens temp\_vect deklariert, das zwei Integer-Werte speichern kann. Es wird als temporäre Variable verwendet, um einen Vektor oder eine Position zu speichern.

**Int count** - Dies deklariert eine Integer-Variable namens count und initialisiert sie mit dem

Wert 0.

**Int\_ID** - Dies deklariert eine Integer-Variable namens ID und initialisiert sie mit dem Wert 0.

Sie wird zum Speichern einer Identifikationsnummer verwendet.

**Int yeni\_id** - Damit wird ein Array namens yeni\_id mit 14 Elementen deklariert. Es wird verwendet, um eine neue Identifikationsnummer zu speichern

## RFID-ID-Tracking und Figurenerkennung

In Unterkapitel 5.2.2 wurden die theoretischen Konzepte, die das Funktionsprinzip des RFID - ID - Nachverfolgung beschreiben, vorgestellt. In diesem Unterkapitel wird die Implementierung dieses Standards in das des intelligenten Schachbretts vorgestellt.

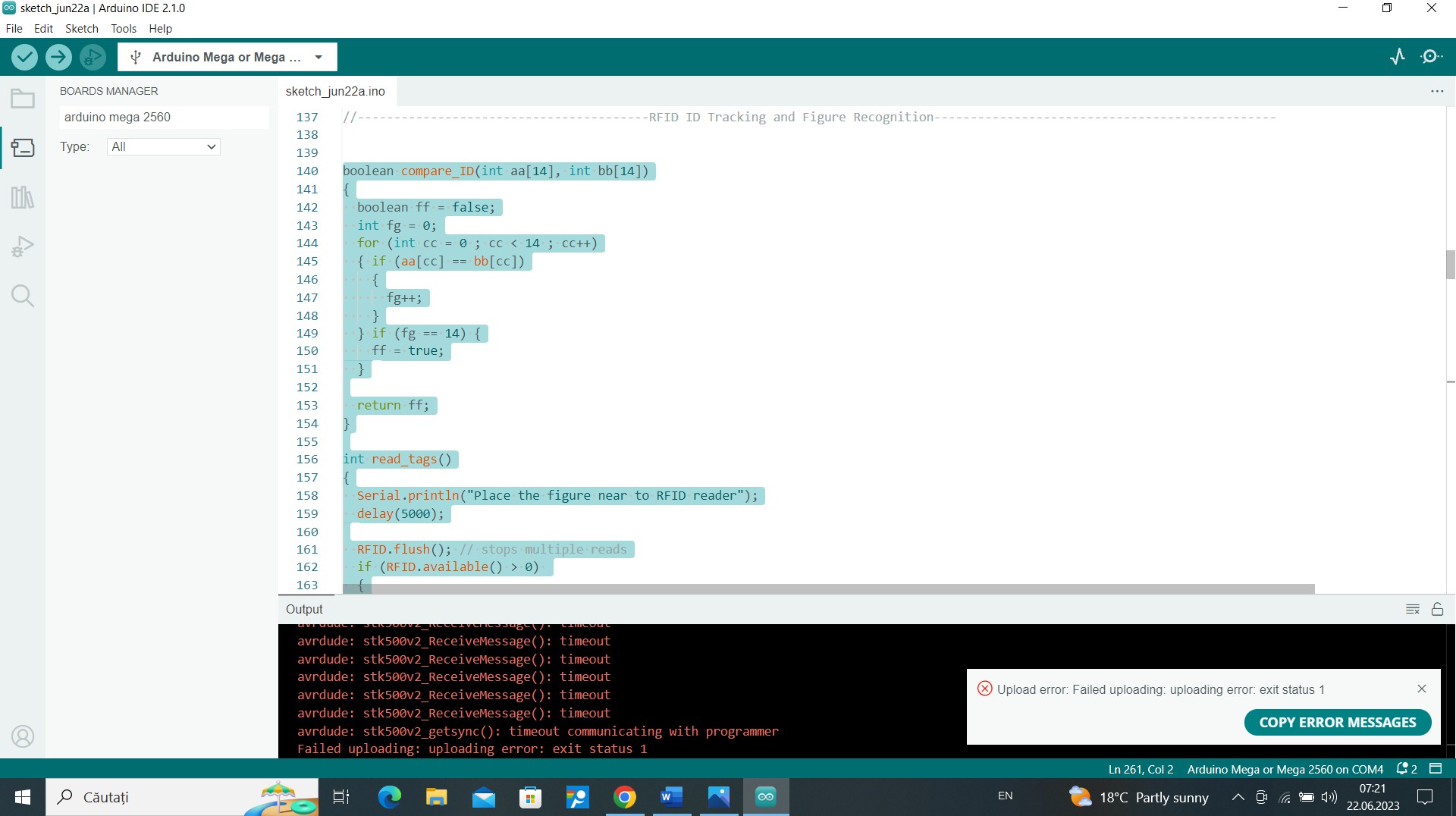


Abbildung 5. 2 RFID-ID-Tracking und Figurenerkennung. Quelle: [11]

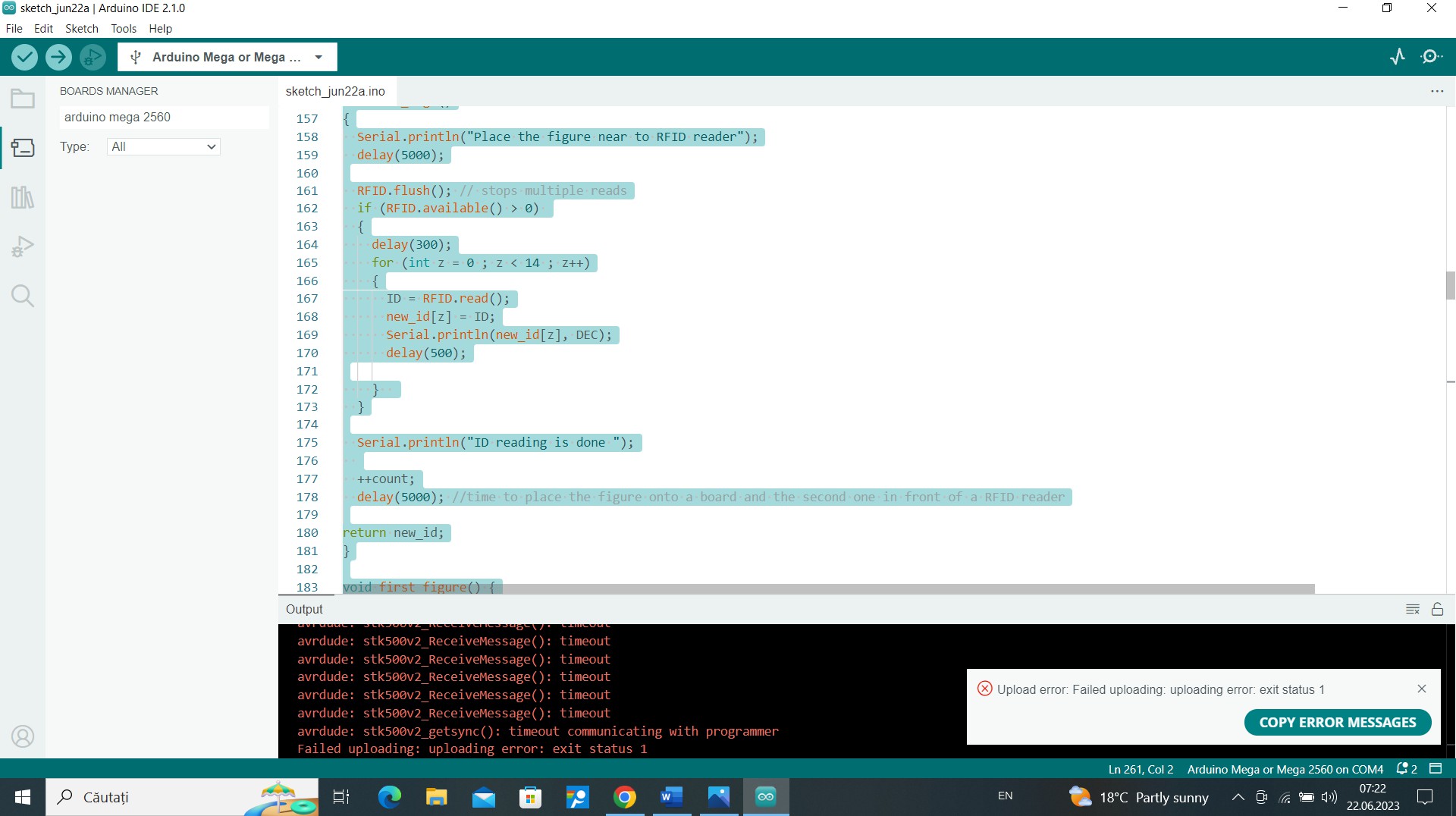


Abbildung 5. 3 RFID-ID-Tracking und Figurenerkennung. Quelle: [11]

Die Funktion **compare\_ID** nimmt zwei Arrays von Ganzzahlen (aa und bb) als Eingabeparameter. Sie vergleicht die Werte bei jedem Index der beiden Arrays und zählt die Anzahl der übereinstimmenden Werte. Wenn alle 14 Werte übereinstimmen, gibt sie true zurück, andernfalls false. Diese Funktion wird verwendet, um Identifikationsnummern von Schachfiguren zu vergleichen. Die Funktion **read\_tags** ist für das Lesen von Identifikationsnummern aus einem RFID- Lesegerät zuständig. Sie fordert den Benutzer auf, eine Schachfigur in die Nähe des RFID-Lesegeräts zu stellen, wartet 5 Sekunden und liest dann die Identifikationsnummer Byte für Byte aus. Die Identifikationsnummer wird in dem Array new\_id gespeichert. Nach Abschluss des Lesevorgangs

erhöht die Funktion die Variable count und gibt das Array new\_id zurück.

Die Funktion **first\_figure** wird aufgerufen, nachdem die erste Schachfigur gelesen wurde. Sie vergleicht die new\_id mit den vordefinierten Arrays queen\_id und pawn\_id unter Verwendung der Funktion compare\_ID. Wenn eine Übereinstimmung gefunden wird, identifiziert sie den Typ der Schachfigur (Dame oder Bauer) und führt die notwendigen Aktionen aus, wie z. B. das Platzieren der Figur auf dem Schachbrett oder das Ausführen einer spezifischen Logik für die identifizierte Figur. Wenn keine Übereinstimmung gefunden wird, wird der Fall einer undefinierten Figur behandelt.

Die Funktion **second\_figure** wird aufgerufen, nachdem die zweite Schachfigur gelesen wurde.

Sie führt einen ähnlichen Vergleich wie die Funktion first\_figure durch, um den Typ der Schachfigur zu ermitteln. Auf der Grundlage der Übereinstimmung führt sie geeignete Aktionen durch, um die zweite Figur auf dem Schachbrett zu platzieren.

## Beginn der Ablesung der Anzahl der Figuren und Speicherung der Positionen für jede Figur

In diesem Unterkapitel werden die Funktionen zur Definition der Felder, die den Zügen der zwei Schachfigurentypen (Bauer,Dame) entsprechen, sowie die an diesem Prozess beteiligten Variablen beschrieben. Die Funktionen, die im Folgenden besprochen werden, sind also die in Abbildungen 5.4.

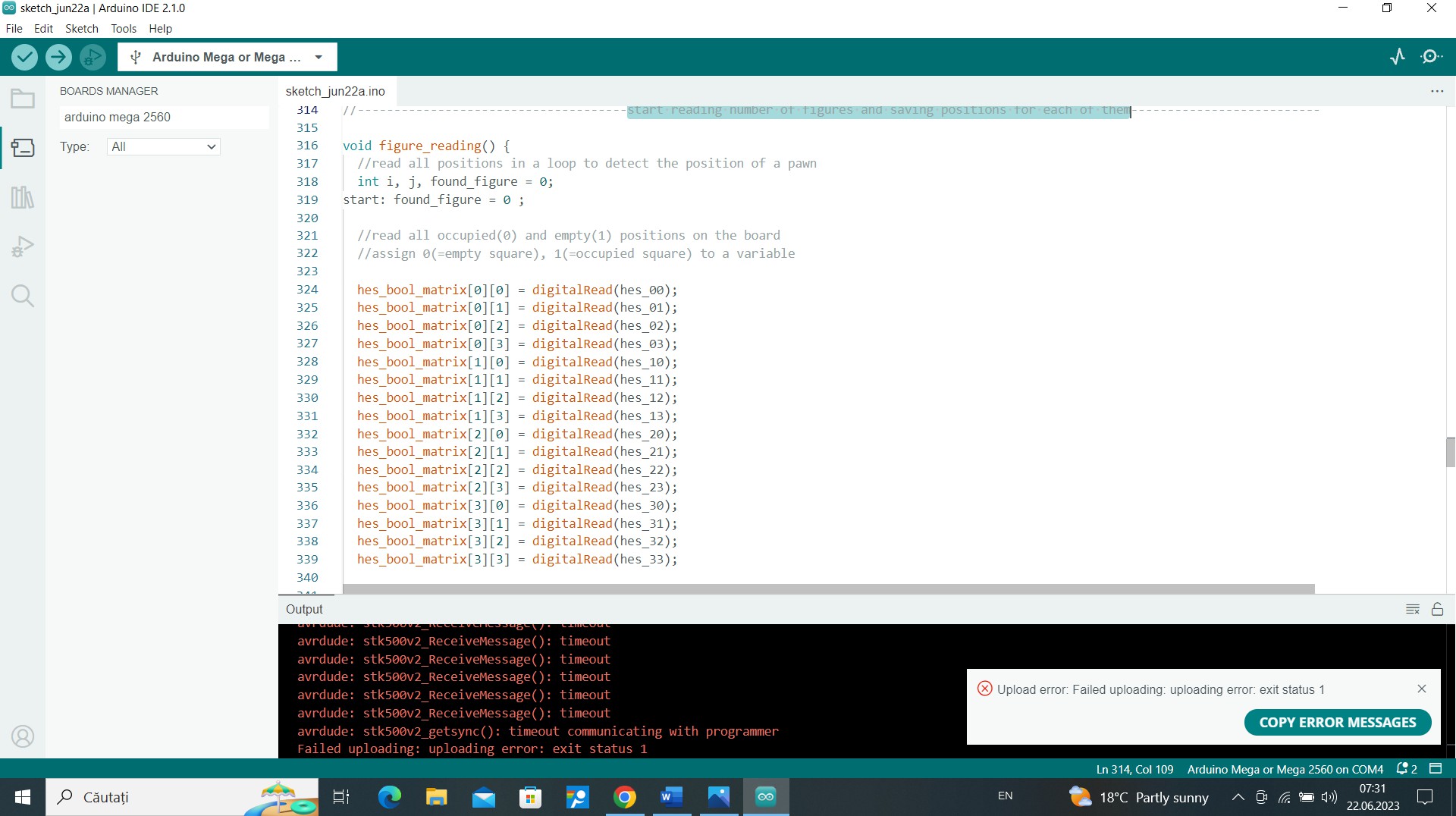


Abbildung 5. 4 Beginn der Ablesung der Anzahl der Figuren und Speicherung der Positionen für jede Figur.

Quelle: [11]

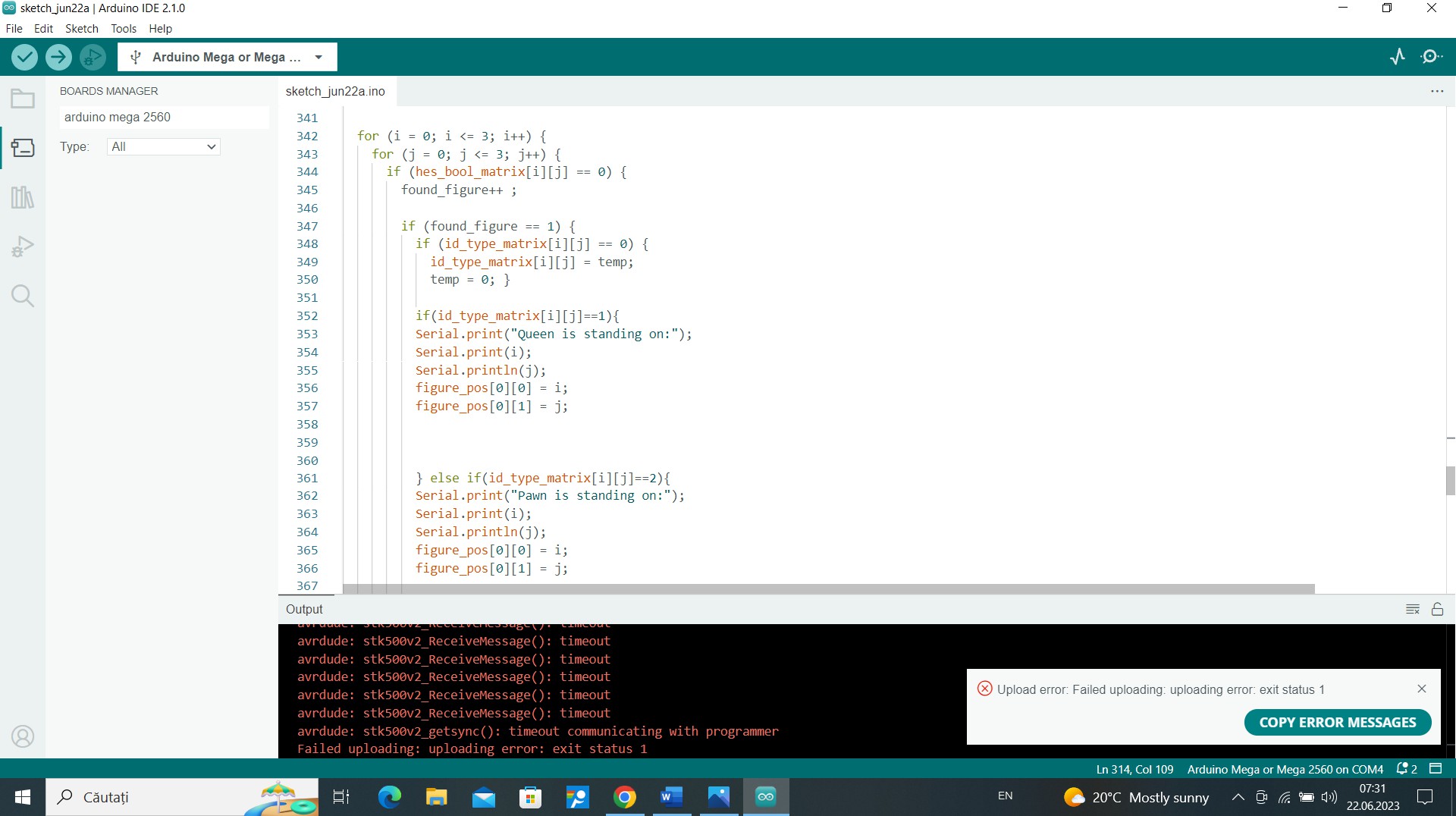


Abbildung 5. 5 Beginn der Ablesung der Anzahl der Figuren und Speicherung der Positionen für jede Figur.

Quelle: [11]

Die Funktion **figure\_reading** beginnt mit der Initialisierung von Variablen und Etiketten (start), die für die Flusskontrolle verwendet werden. Dann liest sie den Zustand aller Positionen auf dem Schachbrett in das Array hes\_bool\_matrix ein.

Als Nächstes durchläuft sie jede Position auf dem Schachbrett und prüft, ob die Position besetzt ist (dargestellt durch den Wert 0 in **hes\_bool\_matrix**). Wenn eine besetzte Stellung gefunden wird, wird der Zähler found\_figure erhöht.

Für die erste gefundene Figur (**found\_figure == 1**) wird der Typ der Figur anhand des id\_type\_matrix-Wertes an dieser Position überprüft. Wenn der Typ 1 ist, wird ausgegeben, dass eine Königin an der Position steht, und das Array figure\_pos wird mit den Koordinaten der Figur aktualisiert. Wenn der Typ 2 ist, wird ausgegeben, dass ein Bauer auf der Position steht, und figure\_pos wird aktualisiert.

Für die zweite erkannte Figur (**found\_figure == 2**) werden die gleichen Operationen wie für die erste Figur durchgeführt.

Nach der Erkennung der Figuren prüft der Code den Wert von found\_figure. Ist er 0 oder 1, springt der Code zurück zum Startlabel, um die Positionen erneut einzulesen und erneut Figuren zu erkennen. Wenn found\_figure den Wert 2 hat, ruft er die Funktion leds\_off auf (im

mitgelieferten Code nicht gezeigt).

Als Nächstes wählt der Code die zu wählende Figur auf der Grundlage ihrer Position aus. Er prüft den Status der Position mit digitalRead mit dem entsprechenden Pin aus hes\_pin\_matrix. Wenn die Position leer ist (**digitalRead liefert 1**), wird **empty\_pos** mit den Koordinaten der leeren Position aktualisiert und der Typ der Figur in der Variablen temp gespeichert. Anschließend wird der Typ der Figur an der leeren Position auf 0 gesetzt.

# Physischer Aufbau des Schachbretts

Für die eigentliche Konstruktion des intelligenten Schachbretts müssen folgende Entscheidungen getroffen werden Materialien und Werkzeuge:

* 25W Lötkolben
* Arduino Mega 2560
* 16 x LED
* 1 x RC522 RFID-Modul
* 28 Drähte von Stecker zu Buchse
* 2 x 20mm 13.56MHZ RFID-Tags

Die technologischen Schritte, die zur physischen Verwirklichung des Tisches führten,sind die folgenden:

* + Zuschneiden der transparenten Plexiglasplatte auf die volle Größe des Spielbretts (24 cm x 19.7 cm).



*Abbildung 6. 1 Plexiglasplatte für Schachbrett*

* Ausschneiden der Bereiche für den Stromanschluss des Schachbretts und den USB-Anschluss für die Programmierung des Entwicklungsboards und das Einsetzen der Streifen, die den Umfang

des Schachbretts definieren. innerhalb der im vorherigen Schritt beschriebenen Struktur.



Abbildung 6. 2 USB-Verbindung zwischen PC und Arduino Mega 2560-Board

* Bestückung jeder Schachfigur mit Neodym-Magneten und RFID-Tags.



Abbildung 6. 3 Schachfiguren mit RFID-Tag und Neodym-Magneten montiert

* Fertigstellung der elektrischen Verbindungen zwischen den Bauteilen im Inneren der Wabe und denen im und allen Bauteilen an den Leistungs- und Erdungsklemmen des Netzteils.

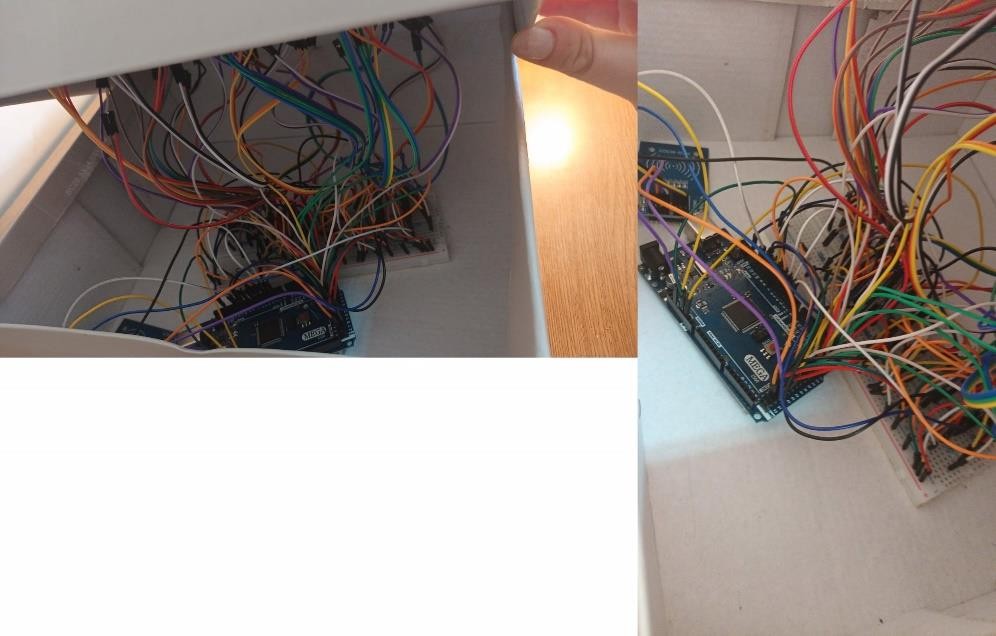


Abbildung 6. 4 Endgültige elektrische Verbindungen

* Präsentation des Endprodukts



Abbildung 6. 5 Abschlusspräsentation des intelligenten Schachbretts

# Erzielte Ergebnisse

Ein Hauptproblem bei dieser Arbeit war das Testen des Codes, der die Regeln des Schachspiels beschreibt, da allein dieser Teil der gesamten Firmware-Anwendung mehr als mehrals 917 Zeilen Code umfasst.

Daraus wurde gefolgert, dass das Debuggen des Codes nur mit Hilfe der Arduino-IDE (Die integrierte Entwicklungsumgebung von Arduino - oder Arduino- Software (IDE) - enthält einen Texteditor zum Schreiben von Code, einen Nachrichtenbereich, eine Textkonsole, eine Symbolleiste mit Schaltflächen für allgemeine Funktionen und eine Reihe von Menüs) möglich ist. Sie wird mit der Arduino-Hardware verbunden, um Programme hochzuladen und mit ihr zu kommunizieren), in einer intuitiven und leicht zu entschlüsselnden Weise, deren Interpretation für das Debuggen von Code und die Anzeige der erzielten Ergebnisse dem physisch realisierten intelligenten Schachbrett so nahe wie möglich kommt.

Trotz ihrer weiten Verbreitung kann ich aus eigener Erfahrung sagen, dass die Verwendung der RFID-Technologie für Heimwerkeranwendungen nicht die beste Wahl ist (natürlich nur, wenn es Alternativen gibt). Ich habe unzählige Male versucht, alles zum Funktionieren zu bringen, indem ich Schachfiguren in die Nähe des Scanners gestellt und beobachtet habe, bis er die ID korrekt gelesen hat. Die Art und Weise, wie ein RFID-Lesegerät eine ID liest, ist mühsam, daher sollte ein serieller Anschluss dafür eingerichtet werden.

Es folgen die Ergebnisse der Implementierung des Schachbrettcodes, die die direkt in der Programmkonsole durchgeführten Softwaretests mit der Darstellung von Situationen, die denen auf dem physischen Schachbrett ähneln, bestätigen.

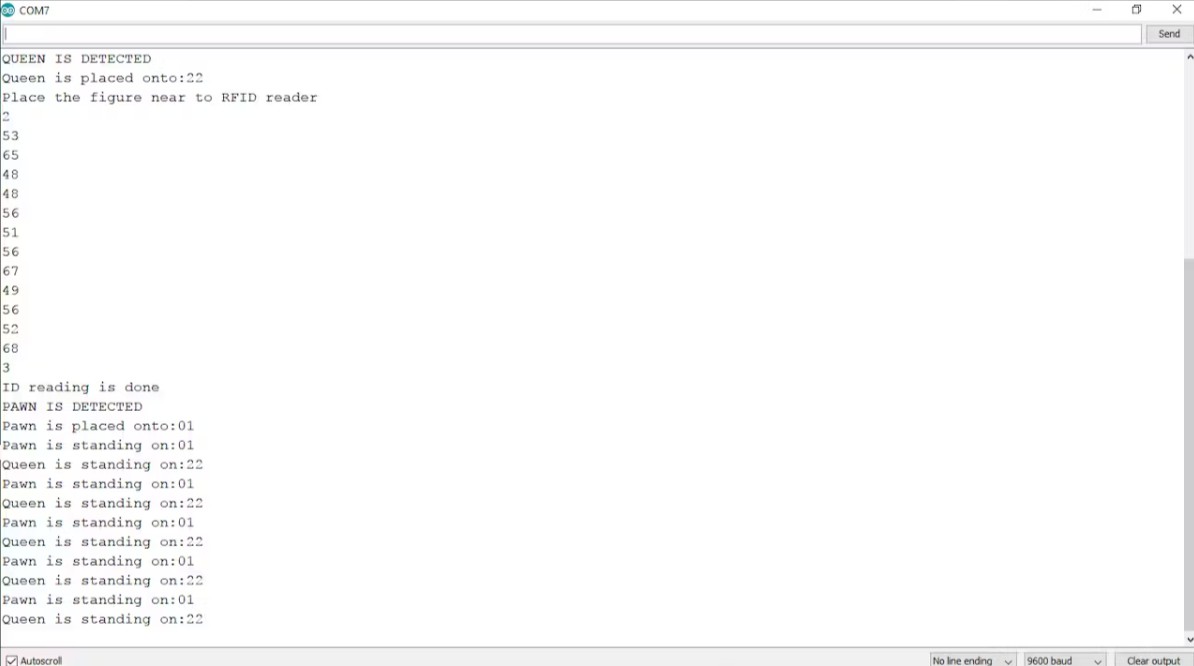


Abbildung 7. 1 Arduino IDE Serieller Monitor Ergebnisse



Abbildung 7. 2 Physikalische Ergebnisse

# Schlussfolgerungen, persönliche Beiträge und weitere Verbesserungen

## Schlussfolgerungen

Die vorliegende Arbeit zielte auf den Entwurf und die physische Realisierung eines intelligenten Schachbretts das seine Benutzer beim Erlernen des Schachspiels anleitet, indem es eine Funktionalität, die in Echtzeit auf angewandte Stimuli reagiert (Anzeige möglicher Spielfelder beim Aufnehmen einer Figur, automatische Aktivierung der Schachuhr nach einem korrekten Zug und Warnung des Benutzers nach einem falschen Zug, Anzeige spezifischer Schachereignisse, durch unterschiedliche LED- Farben. Um zum fertigen Produkt zu gelangen, mussten die folgenden Schritte durchgeführt werden: Entwurf und Implementierung des Hardware-Schemas in Anhang 1, das aus den in Kapitel 4 vorgestellten Funktionsblöcken besteht, Entwicklung einer Firmware- Anwendung, die sowohl Funktionen zur Konfiguration der Hardware- Blöcke als auch Funktionen zur Implementierung der Schachspielregeln umfasst (Thema Kapitel 5), der Entwurfund die Implementierung einer physischen Struktur, die die elektronischen Schaltkreise des Schachbretts kapselt (in Kapitel 6 beschrieben) und so den Nutzern ein einzigartiges Erlebnis während der Schachpartie bietet, und das Testen von Hardware- und Softwarekomponenten während der Entwicklung des intelligenten Schachbretts (Kapitel 7).

## Persönliche Beiträge

* Entwurf und Implementierung des Hardware-Schemas des gesamten Systems, um das Lesen das korrekte Lesen der Eingangsdaten von der Firmware-Anwendung und das Senden der von der Software verarbeiteten Daten an Ausgabegeräte;
* Erstellung einer Bibliothek von Funktionen, die die Schachregeln für die zwei vorhandenen Figuren (Bauer, Dame) und das Hauptprogramm, das die die in den beiden Bibliotheken definierten Funktionen aufruft, um die endgültige

Funktionalität des gesamten Systems zu erhalten Systems zu erhalten;

* Herstellung einer physischen Struktur, die zeigt, dass die endgültige Anwendung der elektronischen Schaltung die eines intelligenten Schachbretts ist.
* Durchführung der in Kapitel 7 beschriebenen Tests und Einführung einer schnellen Methode zum Debuggen von Code, der Schachregeln implementiert.

## Weitere Verbesserungen

Da sich das Smart Chessboard-Papier aufgrund der hohen Preise der Komponenten auf einen Prototyp eines Schachbretts mit der Bestimmung der Position der Schachfiguren konzentriert, zielen die meisten Verbesserungen, die vorgenommen werden können, auf den Prozess der einfachen Lagerung und Verwaltung des Schachbretts ab gespieltes Schachspiel zur Analyse der Benutzer. Eine vorgeschlagene Lösung in dieser Richtung ist die Hinzufügung einer Uhr zur Speicherung der Spielzeit.

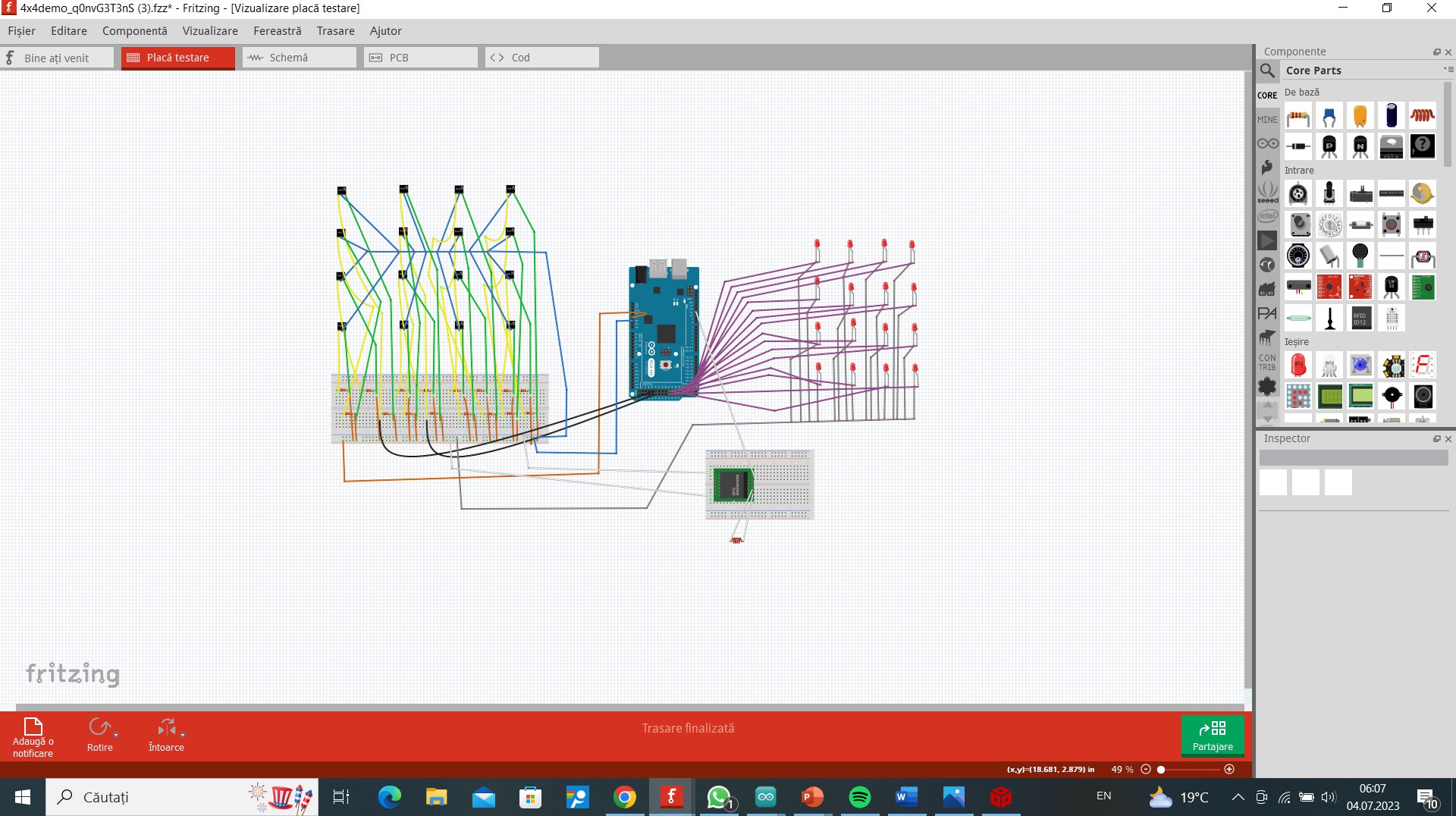
Eine weitere Verbesserung könnte darin bestehen, ein Schachbrett in Originalgröße herzustellen.

# Literaturverzeichnis

1. Schachbrett, das die Erkennung von Schachfiguren ermöglicht, [USER\_MANUAL.pdf(pendobucket.us-east-1.linodeobjects.com),](https://pendobucket.us-east-1.linodeobjects.com/USER_MANUAL.pdf) abruft am Datum 25.03.2023
2. Der Zauberwürfel GoCube, [https://particula-tech.com/shop/gocube-2x2/,](https://particula-tech.com/shop/gocube-2x2/) abruft am Datum25.03.2023
3. die GoDice Smart Dice von GoCube, [https://getgocube.com/godice/,](https://getgocube.com/godice/) abruft am Datum25.03.2023
4. [www.nxp.com/docs/en/data-sheet/MFRC522.pdf,](http://www.nxp.com/docs/en/data-sheet/MFRC522.pdf) abruft am Datum 6.06.2023
5. [https://www.aliexpress.com/item/32898752493.html,](https://www.aliexpress.com/item/32898752493.html) abruft am Datum 6.06.2023
6. [https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-](https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification) [identification](https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification), abruft am Datum von 6.06.2023
7. [http://www.handsontec.com/dataspecs/RC522.pdf,](http://www.handsontec.com/dataspecs/RC522.pdf) abruft am Datum von 6.06.2023
8. https:/[/www.javatpoint.com/arduino](http://www.javatpoint.com/arduino-mega-pinout)-[mega-pinout,](http://www.javatpoint.com/arduino-mega-pinout) abruft am Datum 6.06.2023
9. [https://speed.pub.ro/speed3/wp-](https://speed.pub.ro/speed3/wp-content/uploads/2020/07/MOCANU_Victor_Valentin_2020.pdf) [content/uploads/2020/07/MOCANU\_Victor\_Valentin\_2020.pdf](https://speed.pub.ro/speed3/wp-content/uploads/2020/07/MOCANU_Victor_Valentin_2020.pdf), abruft am Datum 15.03.2023
10. [https://www.corelis.com/education/tutorials/spi-tutorial/,](https://www.corelis.com/education/tutorials/spi-tutorial/) abruft am Datum 6.06.2023
11. https:/[/www.ha](http://www.hackster.io/tahir_miriyev/electronic-chessboard-4x4-demo-prototype-)c[kster.io/tahir\_miriyev/electronic-chessboard-4x4-demo-prototype-](http://www.hackster.io/tahir_miriyev/electronic-chessboard-4x4-demo-prototype-) 7db5e0#schematics, abruft am Datum 6.06.2023
12. [https://docs.arduino.cc/hardware/mega-2560,](https://docs.arduino.cc/hardware/mega-2560) abruft am Datum 6.06.2023

# Anhänge

**Anhang 1 - Hardware-Diagramm des intelligenten Schachbretts**



**Anhang 2 - Bauer oder Dame bewegen**

