

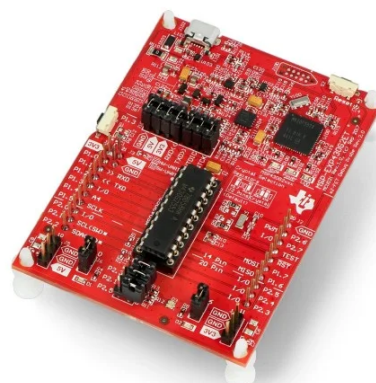
Gedächtnistrainer mit einem TI-Microcontroller

1 Einleitung

Einer der Schwerpunkte im Studiengang *Elektrotechnik* der Beuth Hochschule für Technik Berlin ist die analoge Schaltungstechnik in Kombination mit der Mikrocontrollertechnik. Diese kleinen Mikrocontroller werden dann programmiert, hier passiert das in der Programmiersprache C.

Das Ziel der Bemühungen ist die Realisierung dieses *Gedächtnistrainers*.

Die Firma *Texas Instruments* bietet mit dem LaunchPad (siehe nebenstehendes Bild) eine kostengünstige Plattform für den Einstieg in diese Technik. Man kann dieses komplette Entwicklungssystem ab etwa 12€ erwerben. Mit diesem wird die Software entwickelt und der Mikrocontroller, hier ein MSP430G2313, programmiert. Danach wird er in die Zielhardware gesteckt, die hier bei der *Lange Nacht der Wissenschaften* gelötet wird.



2 Schaltung und Software

In Abb. 2 können wir die Schaltung unserer Platine sehen. Welches Bauteil ist wofür zuständig? Herzstück ist der Mikrocontroller MSP430G2313, die die gesamte Ansteuerung übernimmt. Er arbeitet mit 1 MHz und ist ein 16-bit Mikrocontroller. An ihn angeschlossen sind 4 Taster als Eingänge, damit wir etwas zum Bedienen haben. Als Ausgänge haben wir 4 LEDs in verschiedenen Farben passend zum Taster. Das Ganze läuft mit einer 3 V-Knopfzelle.

Die Spielidee ist, dass der Computer eine Farbsequenz vorspielt, deren Länge sich mit jedem Durchlauf erhöht. In jedem Schritt wiederholt der Spieler die wachsende Sequenz, die maximale Länge ist 16. Das kann man aber im Programm erhöhen.

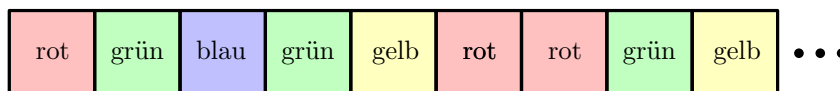


Abbildung 1: **Gedächtnisspiel**-array als Speicher der Farben

Die Software realisiert einen kleinen Zustandsautomaten. Das Spiel kennt die beiden wichtigen Zustände *SimonSays*, *PlayerSays* sowie Zustände für Neustart, Gewinnen und Verlieren. Wenn der Computer die Sequenz vorgibt wird eine Zufallszahl erzeugt, diese einer Farbe zugeordnet und das wird in einem `array`, einem Speicherfeld, abgelegt. Nun ist der Spieler dran, die Sequenz zu wiederholen. Hierzu werden die Tasteneingaben mit den Werten im `array` verglichen. Hat man alle 16 geschafft, dann hat man gewonnen. Bei einer Fehleingabe hat man diese Runde verloren.

Bedient man das Spiel nicht, so wird der Stromverbrauch reduziert, die LEDs werden ausgeschaltet. Im Hintergrund werden weiterhin Zufallszahlen erzeugt. Das ist ja für einen deterministischen Automaten¹ nicht so einfach, zufällige Dinge zu machen, daher macht der das einfach im Hintergrund immer weiter. Ein neuer Tastendruck führt zum nächsten Spiel. Spielt man lange nicht, dann nimmt man am Besten die Batterie raus.

¹Und wir wollen ja normalerweise, dass Computer exakt voraussagbares Verhalten haben. Oder nicht?

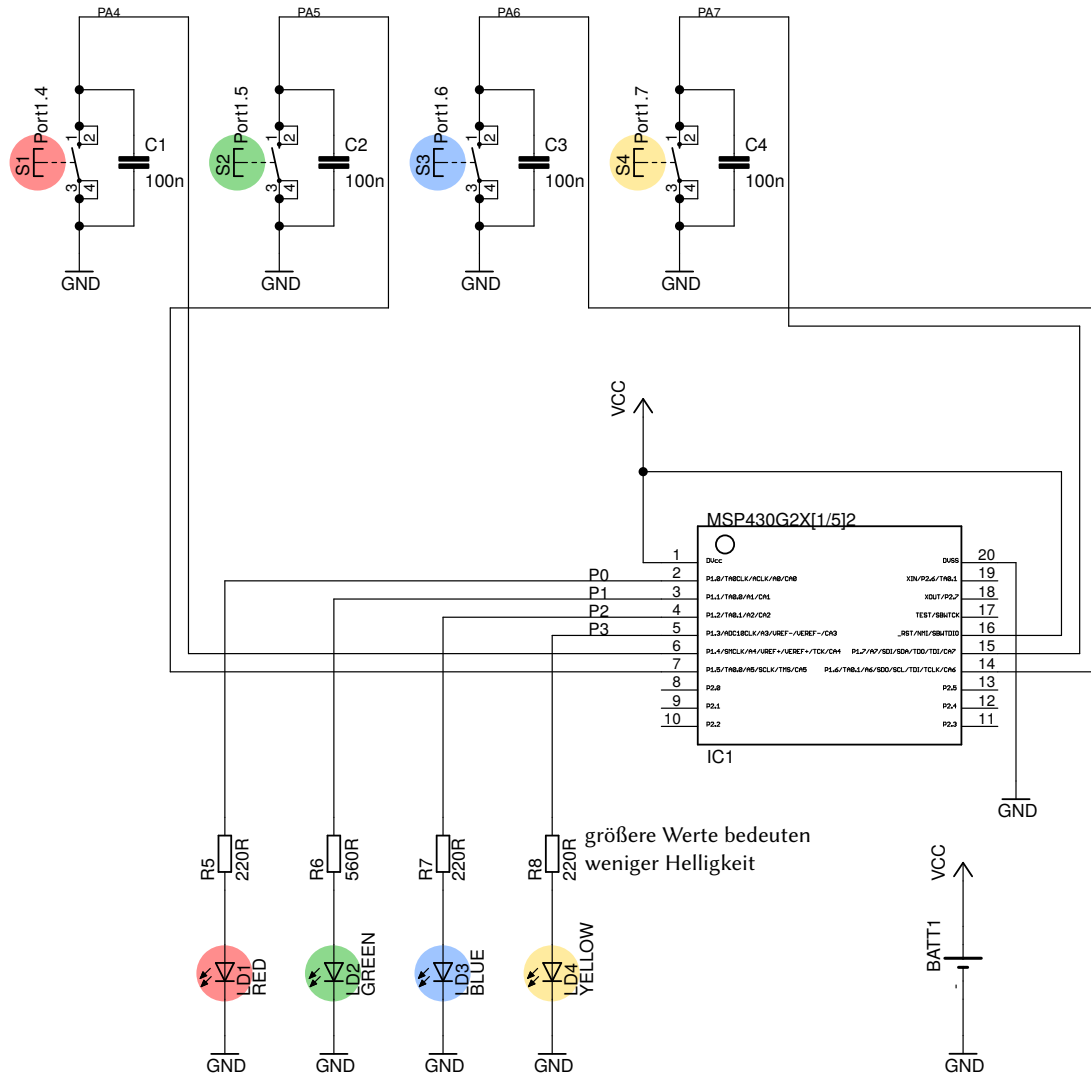


Abbildung 2: Gedächtnisspiel– der Schaltplan

3 Bestückung der Leiterplatte

Der Bestückungsplan zeigt, wo die Bauelemente hinkommen. Bauen Sie die Schaltung auf. Beachten Sie dabei folgendes:

- Beginnen Sie mit den flachen Bauelementen wie Widerständen und Dioden. Dann stückweise zu den höheren Bauteilen.
- Beachten Sie die Ausrichtung der IC-Fassung. Die hat eine Kerbe an einer Schmalseite, die auch der Mikrocontroller hat.
- Löten Sie zügig und vermeiden Sie langes *Herumbraten*, hierbei nehmen die Bauteile Schaden

Inbetriebnahme

Die Inbetriebnahme geschieht in zwei Schritten. Erst wird die Schaltung geprüft und an die Betriebsspannung angeschlossen. Zur Inbetriebnahme der Schaltung gehen Sie wie folgt vor

- überprüfen Sie Ihre Schaltung visuell:
 - Sind alle Lötstellen wirklich gelötet?
 - Haben Sie keine Bauteile verwechselt?
 - Stimmt die Polung von Dioden und Kondensatoren?

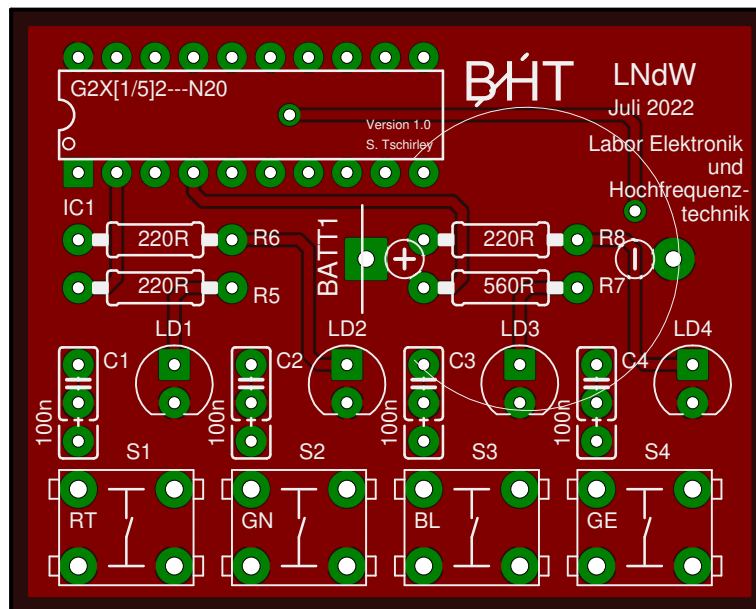


Abbildung 3: **Gedächtnisspiel**: Bestückungsplan, Ansicht von **oben** und in doppelter Größe

- Stellen Sie die Betriebsspannung von 3 V am Labornetzteil ein.
 - Im Falle des Netzteils stellen Sie die Strombegrenzung so ein, dass nur ein Strom von $I = 200 \text{ mA}$ fließen kann.
 - Schließen Sie die Schaltung **ohne Mikrocontroller** an das Labornetzteil an. Vergewissern Sie sich dass Ruhestrom $I < 100 \text{ mA}$ fließt.
 - Messen Sie mit dem Voltmeter nach, ob die Versorgungsspannung an allen Bauelementepins der Schaltung anliegt.
- ➔ **Erstes Ziel** – Ihre Schaltung funktioniert gleichspannungsmäßig.

Nun soll das **Gedächtnisspiel** betrieben werden.

- Schließen Sie die Batterie an.
- Spielen Sie los.

4 Software zum Projekt

Die komplette Software zu der kleinen Leiterplatte, die Schaltplan- und Leiterplattendateien können auf der Webseite zur Veranstaltung heruntergeladen werden

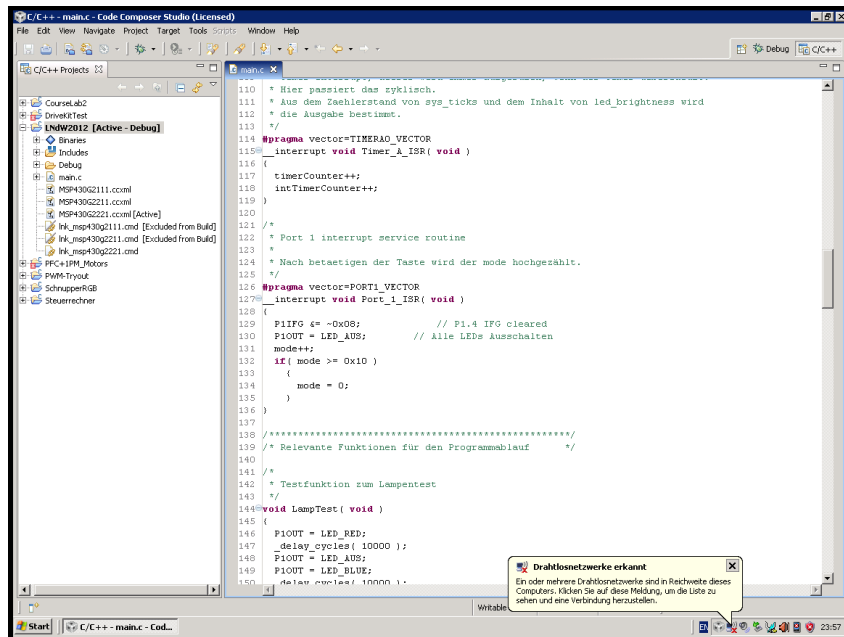
<https://prof.bht-berlin.de/tschirley/lange-naechte-der-wissenschaften/>



5 Ich will mehr!!!!

Die Programmierung des Mikrocontrollers ist mit einem LaunchPad realisiert worden. Dieses kleine Werkzeug erlaubt einen Einstieg in die Mikrocontrollertechnik. Das LaunchPad ist ein komplettes Entwicklungssystem, es beinhaltet alles, was man zum Programmieren eines kleinen Controllers benötigt:

- Eine freie Entwicklungsumgebung zum Schreiben und Übersetzen der eigenen Programme



- Einen *Emulator-Anschluss*, mit dem man die Programme auf den Mikrocontroller herunterladen kann. Dort werden diese in einen Flash-Speicher² programmiert und sind dann dauerhaft dort verwendbar.
- Zwei LEDs und einen Schalter für die ersten kleinen Softwareprojekte

Mit dem LaunchPad kann man viele kleine Projekte realisieren. Hinweise dazu sind auf der Webseite zum LaunchPad von *Texas Instruments* zu finden:

<https://www.ti.com/tool/MSP-EXP430G2ET>

Dort kann man auch die Entwicklungssoftware herunterladen. Weiterhin findet man dort ein Video-basiertes Tutorial³

Dieser kleine 16-bit RISC Mikrocontroller ist recht leistungsfähig. Die maximale Taktfrequenz ist 16 MHz, neben 256byte RAM stehen 4 kByte Flash-Speicher zur Verfügung. Typen mit mehr RAM und mehr Flashspeicher sind verfügbar. In diesem Projekt wird mit dem MSP430G2313 einer der Controller mit kleinstem Speicher verwendet, der etwa zu 30% ausgenutzt wird.

²Na den kennt man ja vom USB-Stick

³Eigentlich ist das eine Einführung, Tutorial klingt aber irgendwie internationaler.

