



# Nappy's Home

Smarthome Steuerung auf einem 8051 Mikrokontroller

von

**Heiko Faller, Mehmet Ali Incekara und Tom Wolske**

# Ehrenwörtliche Erklärung

gemäß § 5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.

Wir (Heiko Faller, Mehmet Ali Incekara und Tom Wolske) haben die vorliegende Arbeit mit dem Titel

„ Nappy’s Home “

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Karlsruhe, den 16.6.2016

---

Ort, Datum

# Inhaltsverzeichnis

Abbildungsverzeichnis . . . . .	IV
Tabellenverzeichnis . . . . .	V
<b>1 Einleitung . . . . .</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Ziel der Arbeit . . . . .	1
1.3 Aufbau der Arbeit . . . . .	1
<b>2 Grundlagen . . . . .</b>	<b>2</b>
2.1 Der Mikrocontroller . . . . .	2
2.2 Assembler . . . . .	2
2.3 MSC-8051 . . . . .	2
2.4 MCU 8051 IDE . . . . .	3
<b>3 Konzept . . . . .</b>	<b>4</b>
3.1 Konzept und Idee . . . . .	4
<b>4 Umsetzung . . . . .</b>	<b>6</b>
4.1 Rolladen-Steuerung . . . . .	6
4.1.1 Manuelle Steuerung . . . . .	6
4.1.2 Audomodus . . . . .	6
4.2 Licht-Steuerung . . . . .	6
4.3 Heizung-Steuerung . . . . .	7
<b>5 Zusammenfassung . . . . .</b>	<b>8</b>

# Abbildungsverzeichnis

1	MCU 8051 IDE . . . . .	3
2	Schaltung für die Heizung . . . . .	5
3	Schaltung für das Licht . . . . .	5

# Tabellenverzeichnis

1	Schaltplan Heizungssteuerung . . . . .	4
2	Schaltplan Lichtsteuerung . . . . .	5
3	My caption . . . . .	6

# 1 Einleitung

Die Menschen wollen immer weiter vernetzt sein! Das betrifft auch das Haus in denen Sie leben.

Smart Home dient als Oberbegriff für technische Verfahren und Systeme in Häusern, in deren Mittelpunkt eine Erhöhung von Wohn- und Lebensqualität, Sicherheit und effizienter Energienutzung steht.

## 1.1 Motivation

Auf die Idee ein Smarthome zu Entwickeln sind wir schnell gekommen.

Ältere Menschen in unserer Gesellschaft haben oft angst in den Urlaub zu fahren, weil sie ihr Eigenheim nicht alleine lassen wollen. Andere wollen Strom sparen um die Umwelt zu schonen. Aus diesem Grund wollen wir ein SmartHome entwickeln. Damit ältere Menschen wieder in den Urlaub gehen können und Umweltbewusste Menschen die Umwelt besser schonen können.

## 1.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist es einen Mikrocontroller zur Haussteuerung zu entwickeln. Zunächst findet die Implementierung und Ausführung als Simulation in der MCU 8051 IDE statt.

## 1.3 Aufbau der Arbeit

Das Ziel dieser Arbeit lässt sich in Teilziele kategorisieren, um die Zielerreichung zu garantieren.

Das nachfolgende Kapitel befasst sich mit den Grundlagen der Mikrocontrollerentwicklung. Anschließend wird das Konzept und die Implementierung näher erläutert. Als Abschluss folgt eine Zusammenfassung.

# 2 Grundlagen

## 2.1 Der Mikrocontroller

Ein Mikrocontroller ist ein Ein-Chip-Computersystem, der einen Prozessor und Peripheriefunktionen beinhaltet. In den meisten Fällen befindet sich der Arbeits- und Programmspeicher teilweise oder auch komplett auf demselben Chip.

In der Industrie ist der Mikrocontroller bereits seit vielen Jahren ein nicht mehr wegzudenkendes Bauteil. Sie werden im Alltag in eingebetteten Systemen verwendet, zum Beispiel in Waschmaschinen, Fernsehgeräte und sogar im Fahrzeug als Steuergeräte für das Antiblockiersystem (ABS), Airbag und weitere<sup>1</sup>.

Mikrocontroller können in verschiedenen Sprachen programmiert werden. Welche sich jedoch am besten eignet, ist vom Anwendungsfall abhängig. Assembler eignet sich besonders, da es einen direkten Einstieg in die Hardware bietet und keine Abhängigkeiten zu anderen Compilern hat<sup>2</sup>.

## 2.2 Assembler

Ein Assembler ist ein Übersetzer für Programmcode, der sich aus Maschinenbefehlen zusammensetzt. In der Art der verwendeten Befehle besteht der wesentliche Unterschied zu allen anderen Programmiersprachen.

Während sich Befehle bei den Hochsprachen, wie beispielsweise Java und C++, in der Übersetzung aus mehreren Anweisungen im endgültigen Code zusammensetzen, wird der Assemblerbefehl durch den Assembler lediglich in die entsprechende Binärf orm übersetzt. Weiterhin ersetzt der Assembler Variablen durch die entsprechenden Speicheradressen<sup>3</sup>.

## 2.3 MSC-8051

Der MSC-8051 ist die Bezeichnung einer von Intel vorgestellten Familie von 8-Bit-Mikrocontrollern.

Die Architektur enthält folgende Features<sup>4</sup>.

- 8 Bit CPU
- 2x 16 Bit Timer

---

<sup>1</sup>Vgl. <http://www.mikrocontroller.net/articles/AVR-Tutorial>

<sup>2</sup>Vgl. Kenneth J. Ayala, The 8051 Microcontroller, West Publishing Company, 1991, Seite 3

<sup>3</sup>Vgl. <http://assembler.hpfsc.de/>

<sup>4</sup>Vgl. Kenneth J. Ayala, The 8051 Microcontroller, West Publishing Company, 1991, Seite 8

- ROM mit 4 KByte ROM
- RAM mit 128 Byte
  - 4 Register mit 8 Bit

## 2.4 MCU 8051 IDE

Die MCU 8051 IDE ist eine grafische Entwicklungsumgebung für Mikrocontroller, die auf dem MSC-8051 basieren. Das folgende Bild stammt von der Homepage<sup>5</sup> der Entwicklungsumgebung. Es zeigt im Hintergrund den Aufbau der IDE und im Vordergrund einige Simulationsfeatures.

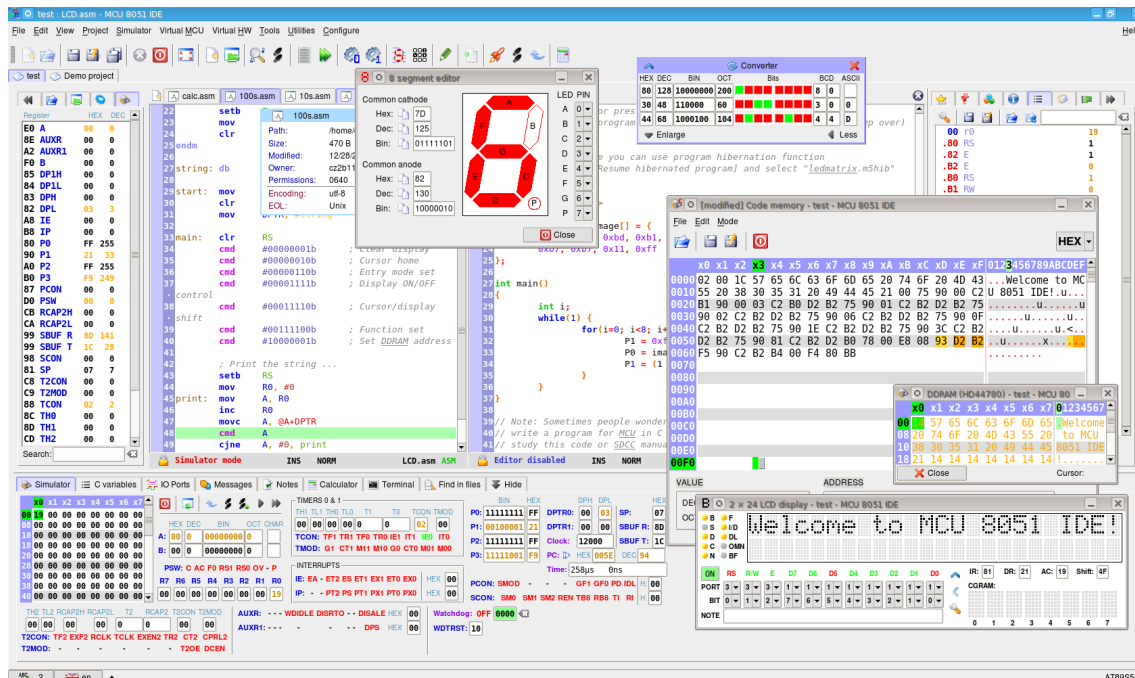


Abbildung 1: MCU 8051 IDE

Die Simulation ist eine Softwarekomponente zur Simulation des Mikrocontrollers in einer virtuellen Umgebung. Zusätzliche Features erweitern die Möglichkeit den Mikrocontroller zu simulieren. Beispielsweise gibt es Hardware-Komponenten wie Schalter, Timer und Temperatur Sensor.

<sup>5</sup><http://www.moravia-microsystems.com/mcu-8051-ide/>



Tabelle 1: Schaltplan Heizungssteuerung

Heizungsschalter	Temperatursensor Schalter	Temperatur < Wert	Heizung An?
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

## 3 Konzept

### 3.1 Konzept und Idee

Die Idee ist es, ein SmartHome zu entwickeln, welches auf einem 8051 Microcontroller als Grundlage aufbaut. Dabei sollen drei Hauptfunktionen eines modernen SmartHomes über diesen gesteuert werden.

Die erste Funktion ist eine Temperatursteuerung für die Fußbodenheizung. Dabei soll die Fußbodenheizung in einem Haus so gesteuert werden, dass diese automatisch eingeschaltet wird, sobald die Temperatur unter einen festen Wert fällt. Außerdem soll der Nutzer mit Hilfe eines Schalters festlegen können, ob die Heizung dauerhaft an oder in einen automatischen Modus eingestellt ist. Der Temperatursensor ist zusätzlich abschaltbar, damit die Heizung ausgeschaltet werden kann.

Wird der Heizungsschalter auf An gestellt, läuft die Heizung unabhängig von der Temperatur die ganze Zeit. Wird dieser auf den Automodus gestellt, kommt es auf den Temperatursensor an, da dieser unabhängig von der Heizung ein bzw. ausgeschaltet werden kann. Ist dieser eingeschaltet, schaltet sich die Heizung an, sobald die Temperatur unter einen bestimmten Wert fällt. Aus diesen Anforderungen ergibt sich eine Tabelle aus der alle möglichen Eingaben und Ausgabemöglichkeiten ausgelesen werden können. Aus diesen geht ein Schaltplan hervor, welcher für die Programmierung des 8051 Microcontrollers genutzt wird. Die Tabelle sieht wie folgt aus:

Für diese Steuerung wird die Schaltung nach diesem Schaltplan programmiert:

Dabei kommen die Eingaben aus dem Port 2.0 bis 2.2 und die Ausgabe ist auf dem Port 3.4. Die zweite Funktion ist eine Lichtsteuerung für das Licht. Der Nutzer soll die Möglichkeit haben das Licht mit Hilfe eines Bewegungssensors ein beziehungsweise aus zu schalten. Dieser Sensor ist zusätzlich an einen Schalter angeschlossen,

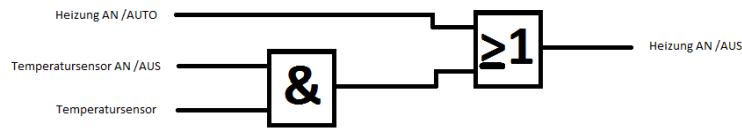


Abbildung 2: Schaltung für die Heizung

Tabelle 2: Schaltplan Lichtsteuerung

Bewegungssensor	Schalter	Licht an?
0	0	0
0	1	0
1	0	0
1	1	1

um diesen An bzw. Aus zu schalten.

Aus diesen Anforderungen ergibt sich eine Tabelle aus der alle möglichen Eingaben und Ausgabemöglichkeiten ausgelesen werden können. Aus diesen geht ein Schaltplan hervor, welcher für die Programmierung des 8051 genutzt wird.

Für diese Steuerung wird die Schaltung nach diesem Schaltplan programmiert:

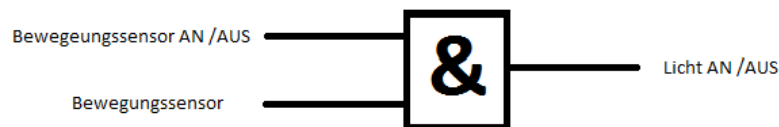


Abbildung 3: Schaltung für das Licht

Dabei kommen die Eingaben aus den Ports P1.0 und 1.1 und die Ausgabe ist auf dem Port 3.5. Der dritte Teil der SmartHome Steuerung ist eine Steuerung für Rollläden. Diese sollen automatisch aufgehen, sobald es draußen hell, beziehungsweise zu gehen, sobald es draußen dunkel wird. Dabei gehen diese komplett auf und zu. Dafür werden kontakte an der Ober- und Unterseite der Rollläden genutzt. Für die automatische Steuerung wird ein Helligkeitssensor genutzt. Aus diesen Anforderungen geht wie zuvor auch eine Tabelle hervor:

Tabelle 3: My caption

SR Oben	SR Unten	SR Hoch	SR Runter	MR Hoch	MR Runter
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

## 4 Umsetzung

### 4.1 Rolladen-Steuerung

Die Schwierigkeit bei der Entwicklung der Rollladenschaltung war, dass eine Menge Port-Bits benötigt werden, weshalb wir uns dazu entschieden haben, ein Modell zur Steuerung von zwei Rollläden zu wählen. Als erstes wird das Automodus-Bit abgefragt, um zu entscheiden, ob der Automodus eingesetzt wird oder die Rollläden manuell gesteuert werden.

#### 4.1.1 Manuelle Steuerung

Im manuellen Modus wird bei den Rollläden abwechselnd überprüft, ob sie hoch oder runter fahren sollen. Dazu wird mittels Polling jeweils abgefragt, wie die Schalterpositionen sind und ob der zu überprüfende Rollladen oben oder unten ist. Dementsprechend ist die Rollladenposition

#### 4.1.2 Audomodus

### 4.2 Licht-Steuerung

einasd

## **4.3 Heizung-Steuerung**

einasc

# 5 Zusammenfassung

shudubinoimapkül