

---

# Digitales Multimeter

## DMM

### Software

Autor	:	Christoph Capiaghi, MSc NTB Interstaatliche Hochschule für Technik Buchs Werdenbergstrasse 4 9471 Buchs / Schweiz <a href="mailto:christoph.capiaghi@ntb.ch">christoph.capiaghi@ntb.ch</a>
Projekt	:	10707
Version	:	1.10
Erstellt am	:	12. Dezember 2017
Letzte Änderung	:	14. Dezember 2017
Geprüft	:	-
Genehmigt	:	-

## Zusammenfassung

Dieses Dokument beschreibt die Software des EuR I Projekts.

## Dokument Versionsverwaltung

Version	Autor	QS	Datum	Status	Änderungen
1.0	C. Capiaghi	-	13. Dez. 2017	Freigabe	Erste Freigabe
1.1	C. Capiaghi	-	14. Dez. 2017		UART Einstellungen hinzugefügt

Original Dokument abgelegt in:

:\10707\_NTB\_EuR1\_Projekt\Software\Testsoftware\Doc\Beschreibung Software.docx

# Inhaltsverzeichnis

<b>1</b>	<b>Hinweise zum Dokument .....</b>	<b>5</b>
1.1	Software .....	5
<b>2</b>	<b>Installation.....</b>	<b>6</b>
2.1	Arduino IDE.....	6
2.2	Bootloader.....	6
2.3	Fuse Bits .....	8
<b>3</b>	<b>Verwendung .....</b>	<b>9</b>
3.1	STM_STATE_STARTUP .....	9
3.2	STM_STATE_CONFIGURATION .....	9
3.3	STM_STATE_VOLTAGE_MEASUREMENT .....	10
3.4	STM_STATE_VOLTAGE_OSCILLOSCOPE .....	10
3.5	STM_STATE_VOLTAGE_UART .....	11
3.6	STM_STATE_CURRENT_MEASUREMENT .....	11
3.7	STM_STATE_LED_TEST.....	11
3.8	STM_STATE_POWER_TEST .....	12
3.9	STM_STATE_NTC_TEST .....	12
3.10	STM_STATE_DIY .....	12
<b>4</b>	<b>Verbesserungen.....</b>	<b>13</b>

## **Verzeichnis der Bilder**

Bild 2-1 Programmieradapter auswählen .....	7
Bild 2-2 Bootloader programmieren (Flash) .....	7
Bild 2-3 Einstellungen Fuse Bits ATmega32u4 .....	8

## **Verzeichnis der Tabellen**

<b>Tabelle 1-1: Beschreibung DUT, Software</b> .....	5
<b>Tabelle 1-2: Einstellungen UART</b> .....	5
<b>Tabelle 2-1: Einstellungen Fuse Bits, Zusammenfassung</b> .....	8
<b>Tabelle 3-1: Konfiguration Multimeter</b> .....	9

# 1 Hinweise zum Dokument

Dieses Dokument beschreibt die Software zum EuR I Projekt Digitales Multimeter. Die Software hat zum Zweck, das Testen zu vereinfachen und die Stärken und Schwächen des Multimeters aufzuzeigen. Um eine einfache Anpassung zu gestatten, verwendet die Software Arduino Bibliotheken. Es ist ausdrücklich erwünscht, dass diese Software den eigenen Bedürfnissen angepasst wird. Aber: **Das Anpassen der Software ist kein Bestandteil des EuR-I Projekts.** Das vorliegende Beispiel soll einen Denkanstoss geben.

## 1.1 Software

Die verwendete Software ist in **Tabelle 1-1** beschrieben.

**Tabelle 1-1: Beschreibung DUT, Software**

Objekt	Beschreibung
Bezeichnung	Testprogramm.ino
Revision	V2.0, svn 268
Seriennummer	-

Die Einstellungen der UART Schnittstelle sind in der **Tabelle 1-2** beschrieben

**Tabelle 1-2: Einstellungen UART**

Parameter	Wert
Speed (baud)	115200 bps
Data bits	8 bits
Stop bits	1
Parity	None
Flow Control	None

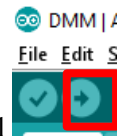
## 2 Installation

Zuerst muss auf das Board mit JTAG ein Bootloader geladen werden. Die Studenten bringen dafür das Board zu einem Betreuer. Falls ein JTAG Programmiergerät vorhanden ist (z.B. JTAG ICE mkII), kann der Bootloader selbständig auf das Board geladen werden. Dazu muss der Bootloader auf Kapitel 2.2 verwendet werden. Ebenfalls müssen die Fuse Bits korrekt gesetzt werden. Siehe dazu Kapitel 2.3.

### 2.1 Arduino IDE

Ihr findet im svn unter *10707\_NTB\_EuR1\_Projekt\Studenten* die komplette Arduino Umgebung und die dazugehörige Software. Schritte:

1. Entpacke die Dateien *arduino-1.x.y*
2. Starte das Programm *arduino-1.x.y \ arduino-1.x.y \ arduino.exe*
3. Öffne das Testsoftware.ino file: File → Open
4. Wähle *Tools* → *Board* → *PASE* aus
5. Schliesse das PASE Board an den PC an
6. Wähle den korrekten Com Port aus: *Tools* → *Port*



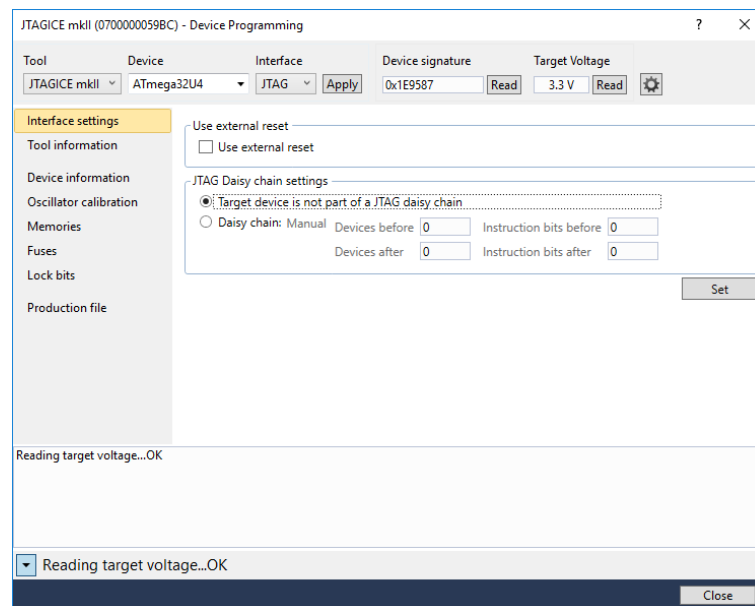
7. Programme das Board mit dem Pfeil Symbol
8. Ignoriere die SD Karten Warnungen
9. Das Board ist mit dem neuen Programm programmiert

### 2.2 Bootloader

Der Bootloader ist ebenfalls im svn unter *10707\_NTB\_EuR1\_Projekt\Studenten* abgelegt und heisst *Caterina-promicro8.hex*. Für die Programmierung wird das Programm Atmel Studio 7.0 verwendet. Der Bootloader wird von einem Betreuer programmiert und muss nur einmal programmiert werden. Danach kann die Arduino DIE verwendet werden.

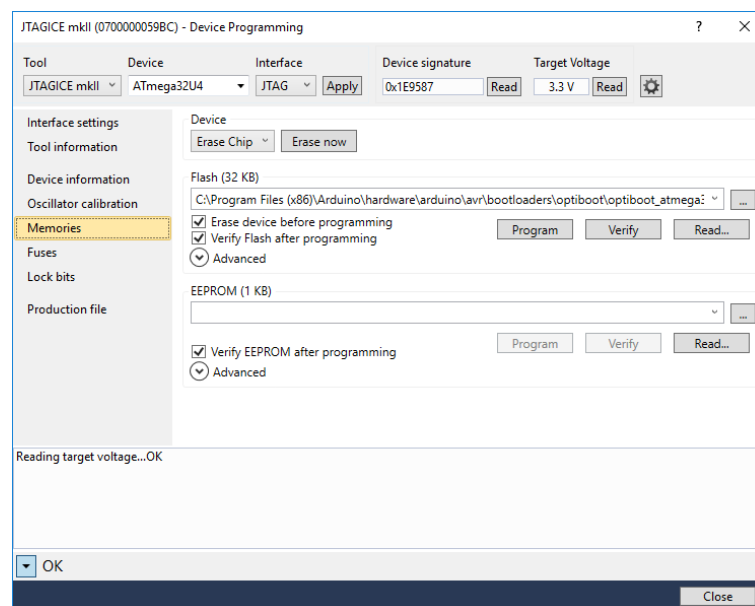
Vorgehen:

1. Unter *Tools* → *Device Programming* den richtigen Programmieradapter (*Tool*) und richtiges Interface auswählen



**Bild 2-1 Programmieradapter auswählen**

2. Zum Unterpunkt *Memories* wechseln und korrekten Bootloader unter Flash auswählen

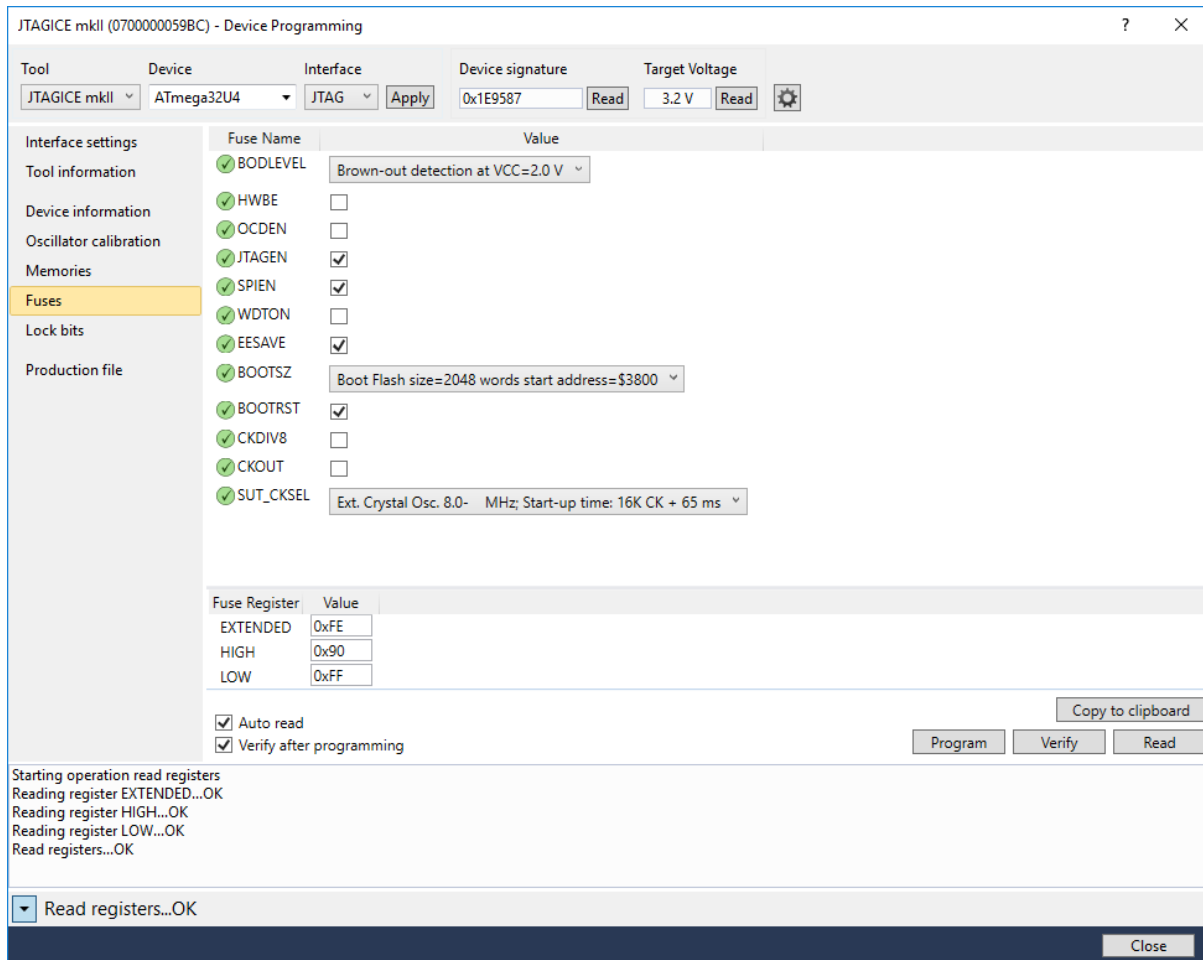


**Bild 2-2 Bootloader programmieren (Flash)**

3. *Program* klicken
4. Eventuell Programmierung mit *Verify* prüfen
5. Fuse Bits kontrollieren / setzen (siehe Kapitel 2.3)

## 2.3 Fuse Bits

Die Einstellungen für die Fuse Bits sind im Bild 2-3 dargestellt.



**Bild 2-3 Einstellungen Fuse Bits ATmega32u4**

In der **Tabelle 2-1** sind die Einstellungen zusammengefasst.

**Tabelle 2-1: Einstellungen Fuse Bits, Zusammenfassung**

Fuse Register	Value
EXTENDED	0xFE
HIGH	0x90
LOW	0xFF



### 3 Verwendung

Das Hauptprogramm ist als State-Maschine aufgebaut. Nachfolgend sind die verschiedenen Zustände dokumentiert.

**Achtung:** Alle Kabel müssen von COM, A und V Port vor dem Start des DMM entfernt werden. Ansonsten kalibriert sich das Messgerät falsch.

#### 3.1 STM\_STATE\_STARTUP

Der Startup beinhaltet:

- Die gespeicherten Werte werden aus dem EEPROM geladen
- Messung der Offsetspannung für die Strom- und Spannungsmessung
- Begrüssung über UART und Display

Der State wechselt automatisch zum Nächsten.

#### 3.2 STM\_STATE\_CONFIGURATION

Hier kann das Multimeter konfiguriert werden. Die Tabelle 3-1 listet die Konfiguration der Reihe nach auf. Ohne diese Werte wird die Spannung bzw. der Strom falsch berechnet. Die Daten werden im EEPROM abgespeichert. Bei Starten des DMM werden die Werte aus dem EEPROM abgerufen.

**Tabelle 3-1: Konfiguration Multimeter**

Beschreibung	Wert
Enter damping Du,high (float)	...
Enter damping Du,low (float)	...
Enter D Vaule Poti High (uint8_t)	0 ... 255
Enter D Vaule Poti Low (uint8_t)	0 ... 255
Enter Gi (float)	...
Enter Shunt Value in Ohm (float)	0.05 nominal

Begriffserklärung: High bedeutet hohe Eingangsspannung (grosser Eingangsspannungsbereich).

- Spannungsmessung
  - Du,high: Dämpfung ( $\frac{1}{G}$ ) bei hoher Eingangsspannung (50 V)

$$G_{U,high} = \frac{R_{eq2,high}}{R_{700} + R_{eq2,high}} = \dots \rightarrow D_{U,high} = \frac{1}{G_{U,high}}$$

- Du,low: Dämpfung bei kleiner Eingangsspannung

$$G_{U,low} = \frac{R_{eq2,low}}{R_{700} + R_{eq2,low}} = \dots \rightarrow D_{U,low} = \frac{1}{G_{U,low}}$$

- D Value Poti High: Potentiometer Parameter bei hoher Eingangsspannung

$$D_{high} = \frac{R_{high} \cdot 256}{R_{AW}}, R_{AW} = 100 \text{ k}\Omega$$

- D Value Poti Low: Potentiometer Parameter bei kleiner Eingangsspannung

- Strommessung

- $G_I$ : Betrag der Verstärkung von der Strommessung  $|G_I| = \left| -\frac{R_{204}}{R_{202}} \right|$
- Shunt Value: Widerstandswert des Shunts (nominal 50 m $\Omega$ )

**Taster:**

Links: Vorhergehendes Menü

Enter: Startet Konfiguration

Rechts: Nächstes Menü

### 3.3 STM\_STATE\_VOLTAGE\_MEASUREMENT

Misst die Spannung und zeigt diese auf dem Display an.

**Taster:**

Links: Vorhergehendes Menü

Enter: Wechselt Verstärkungsfaktor (high und low)

Rechts: Nächstes Menü

### 3.4 STM\_STATE\_VOLTAGE\_OSCILLOSCOPE

Misst die Spannung (ADC Wert) und zeigt diesen auf dem Display als Waveform auf.

**Taster:**

Links: Vorhergehendes Menü

Enter: Wechselt Verstärkungsfaktor (high und low)

Rechts: Nächstes Menü

### 3.5 STM\_STATE\_VOLTAGE\_UART

Misst die Spannung (ADC Wert und Umrechnung) und gibt diese im csv-Format auf der UART-Schnittstelle aus. Aufbau:

Sample, ADC Wert, Berechneter Wert

**Taster:**

Links: Vorhergehendes Menü

Enter: Wechselt Verstärkungsfaktor (high und low)

Rechts: Nächstes Menü

### 3.6 STM\_STATE\_CURRENT\_MEASUREMENT

Misst den Strom und zeigt diesen auf dem Display an.

**Taster:**

Links: Vorhergehendes Menü

Enter: Wechselt Darstellung

Rechts: Nächstes Menü

### 3.7 STM\_STATE\_LED\_TEST

Prüft die Funktion der RGB LEDs und der LEDs.

**Taster:**

Links: Vorhergehendes Menü

Enter: Startet LED Test

Rechts: Nächstes Menü

### 3.8 STM\_STATE\_POWER\_TEST

Maximaler Stromverbrauch:

- Alle RGB LEDs auf Weiss
- Alle LEDs ein
- Bildschirm mit weissen Hintergrund

**Taster:**

Links: Vorhergehendes Menü

Enter: Startet Power Test

Rechts: Nächstes Menü

### 3.9 STM\_STATE\_NTC\_TEST

Liest den ADC Wert der Temperaturmessung aus.

**Taster:**

Links: Vorhergehendes Menü

Enter: Misst ADC Wert von der Temperaturmessung

Rechts: Nächstes Menü

### 3.10 STM\_STATE\_DIY

Beispiel State, für den eigenen Schaltungsteil.

**Taster:**

Links: Vorhergehendes Menü

Enter: Nicht implementiert

Rechts: Nächstes Menü

## 4 Verbesserungen

Folgende Funktionen sind noch nicht optimal ausgelegt:

- Effiziente Berechnung Spannung / Strom aus ADC Werten
  - o Fix-Punkt? Optimierungen
- Effizientere Darstellung auf dem Display
- Achsenbeschriftungen Display
- NTC: Zurzeit nur ADC Wert
- Nur Spannung- und Strommessung. Andere Messungen möglich?
- Automatische Bereichsumschaltung (z.B. wenn ADC-Wert eine Schwelle überschreitet)
- Logging Funktion (Achtung: Speicher knapp)
- Speicheroptimierung