

Asynchronous DRAM over SPI

Implementierung eines DRAM-Controllers mit Web-API

Maximilian Gaul

04.02.2020

Komponenten

ATmega809



Parametrics

Name	Value
Program Memory Type	Flash
Program Memory Size (KB)	8
CPU Speed (MIPS/DMIPS)	20
SRAM (B)	1,024
Data EEPROM/HEF (bytes)	256
Digital Communication Peripherals	4-UART, 1-SPI, 1-I2C
Capture/Compare/PWM Peripherals	16 Input Capture, 11PWM
Timers	5 x 16-bit
ADC Input	16 ch, 10-bit
Number of Comparators	1
Temperature Range (°C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	48

Komponenten

ATmega809

- 5V-kompatibel (passend zum DRAM)
- Modernste AVR-Architektur (u.a. Programmierschnittstelle mit nur noch einem Pin)

Komponenten

ESP8266



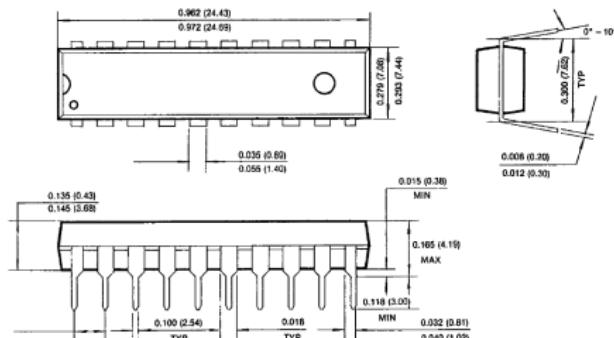
Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n (HT20)
	Frequency Range	2.4 GHz ~ 2.5 GHz (2400 MHz ~ 2483.5 MHz)
	TX Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
Hardware	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SPI/SDIO/I2C/IR Remote Control GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5 V ~ 3.6 V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	-40 °C ~ 125 °C
	Package Size	QFN32-pin (5 mm x 5 mm)
Software	External Interface	-
	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Komponenten

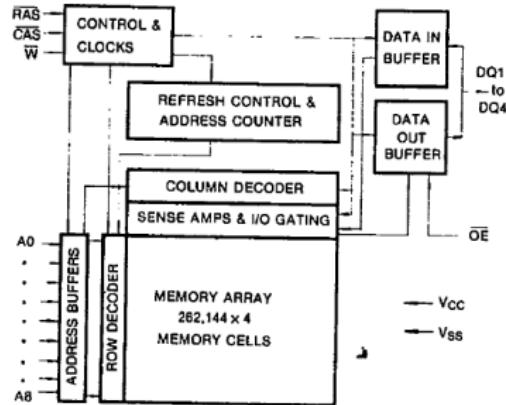
KM44C256BP-8

PACKAGE DIMENSIONS

20-LEAD PLASTIC DUAL IN-LINE PACKAGE



FUNCTIONAL BLOCK DIAGRAM



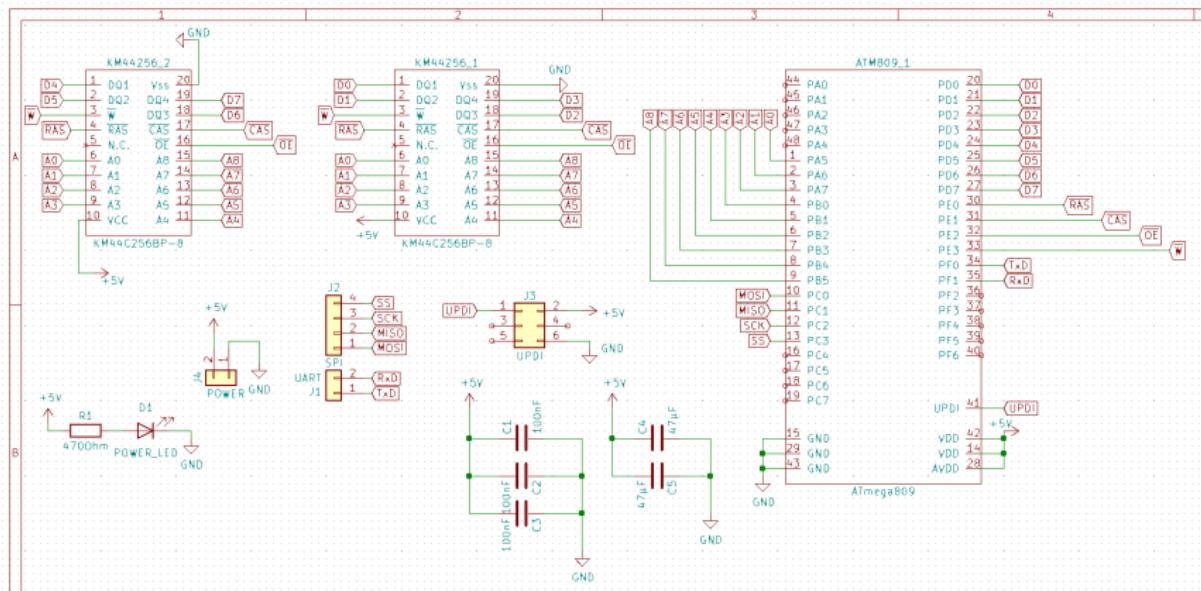
Komponenten

KM44C256BP-8

- 256K × 4 Bit CMOS Dynamic RAM
- $A_0 - A_8$: Adressbus
- \overline{RAS} , \overline{CAS} , \overline{W} , \overline{OE} : Steuerleitungen
- $DQ_1 - DQ_4$: Datenbus
- Max. Random read / write: $150\text{ns} = 150 \cdot 10^{-9}\text{s} = 6.67\text{MHz}$
 - Wird bei weitem nicht erreicht

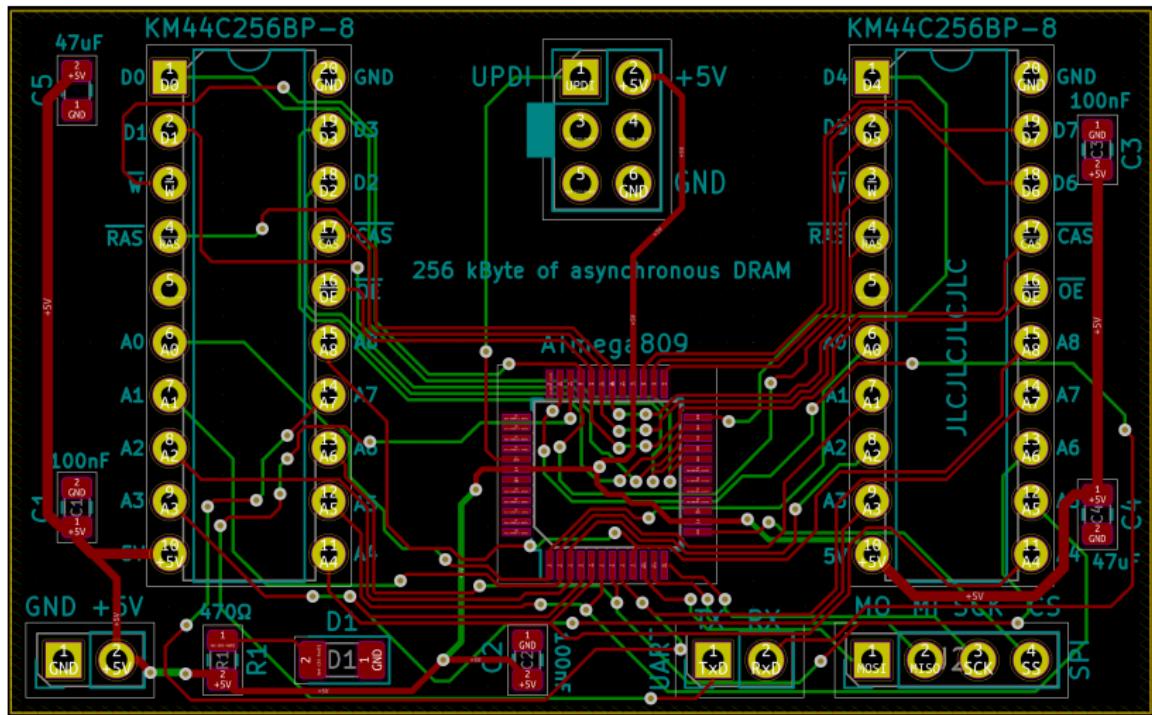
PCB

Schematic



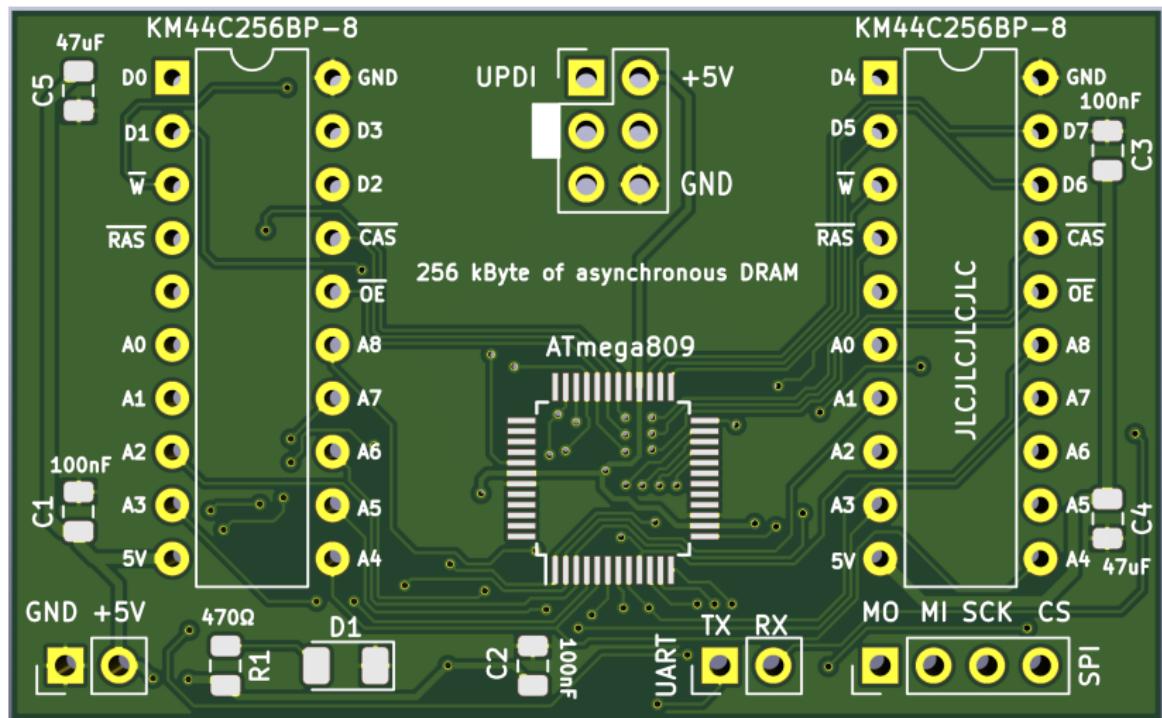
PCB

Layout 2D



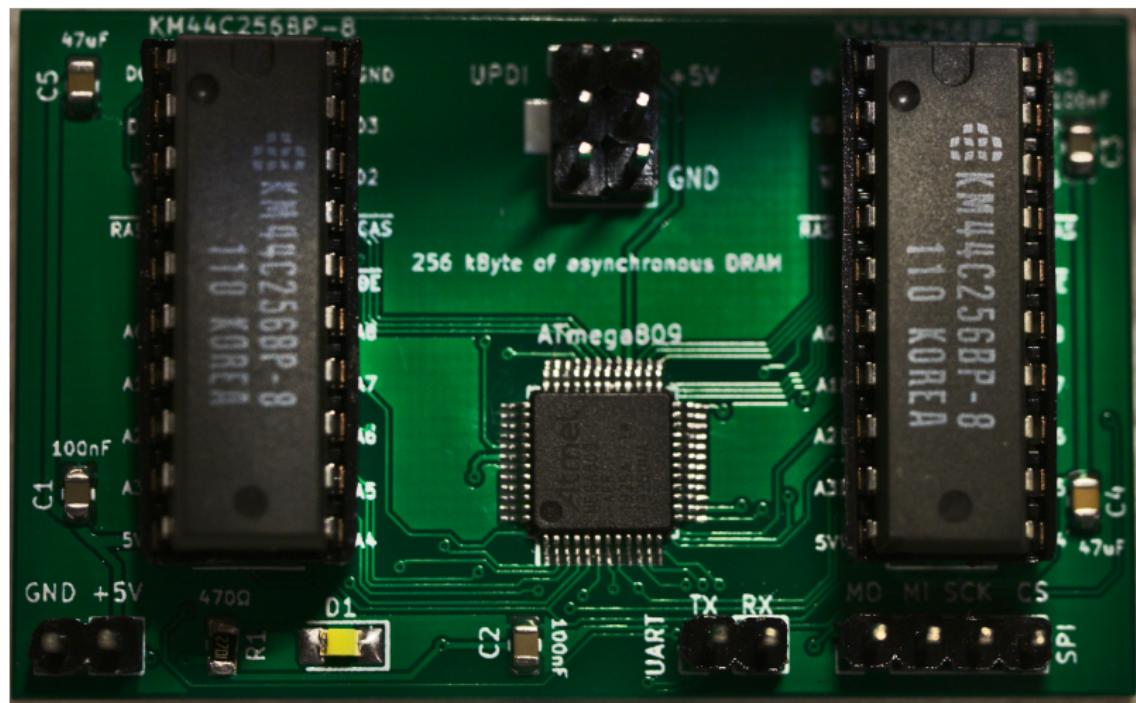
PCB

Layout 3D



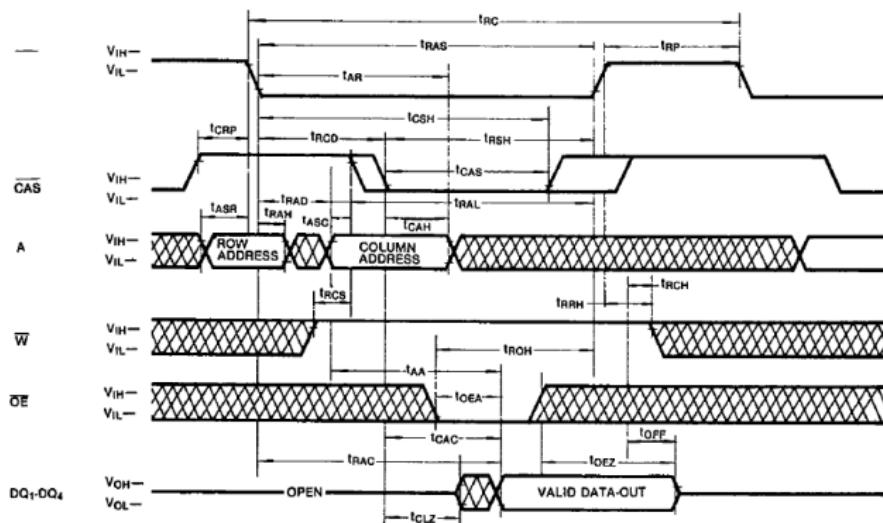
PCB

Fertige Platine



TIMING DIAGRAMS

READ CYCLE

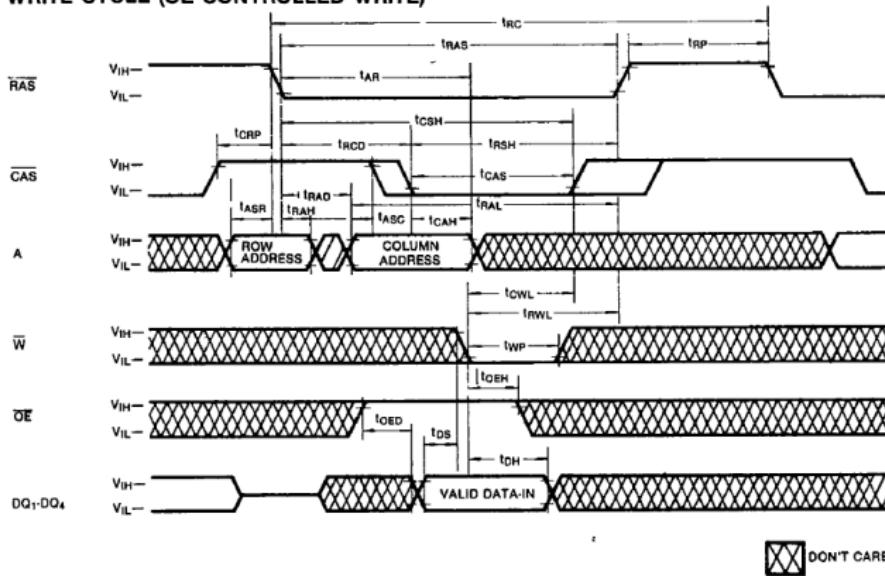


DON'T CARE

KM44C256BP-8

WRITE

WRITE CYCLE (OE CONTROLLED WRITE)



KM44C256BP-8

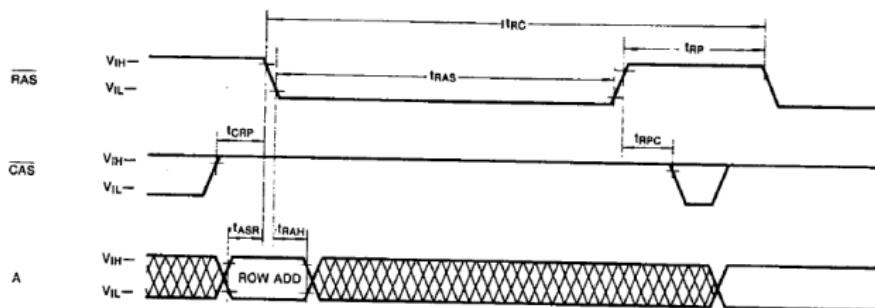
REFRESH

T-46-23-17

TIMING DIAGRAMS (Continued)

RAS-ONLY REFRESH CYCLE

Note: \overline{W} , \overline{OE} = Don't care



ATmega809

- Reagiert auf eintreffende SPI-Bytes des ESP8266
- Erkennt READ bzw. WRITE Befehle, steuert Adress- und Datenleitungen zum RAM entsprechend
- Führt zyklische \overline{RAS} -Refreshes aus (Timer-Interrupt)

Probleme zwischendurch

Microchip Device Support Packs

Version	Description	
1.4.346 (2019-12-20)	Set interrupt vector size for ATmega809/808 to 4 bytes. Corrected data memory size.	 Download

Probleme zwischendurch

Spannungslevel

- DRAM: 4.5V - 5.5V
- ATmega809: 1.8V - 5.5V
- ESP8266: 2.5V - 3.6V

ATmega809

```
1 int main(void) {
2     initDRAMHandler(&dramHandler);
3
4     initCPU();
5     initSPI();
6     initTimer0();
7
8     while (1) {
9         if (dramHandler.hasPendingRefresh) {
10             dramHandler.refreshRASonly(&dramHandler);
11             dramHandler.hasPendingRefresh = false;
12         }
13         if (dramHandler.hasPendingBufferUpdate) {
14             dramHandler.processAndRespondBuffer(&dramHandler);
15             dramHandler.hasPendingBufferUpdate = false;
16         }
17     }
18 }
```

ATmega809

```
1 ISR(TCA0_CMP0_vect) {
2     dramHandler.hasPendingRefresh = true;
3
4     /* Clear interrupt flag */
5     TCA0.SINGLE.INTFLAGS |= (1 << TCA_SINGLE_CMP0EN_bp);
6 }
7
8 ISR(SPI0_INT_vect) {
9     if(SPI0.INTFLAGS & SPI_RXCIE_bm) {
10         const uint8_t data = SPI0.DATA;
11         dramHandler.buffer.push(&dramHandler.buffer, data);
12         dramHandler.hasPendingBufferUpdate = true;
13     }
14 }
```

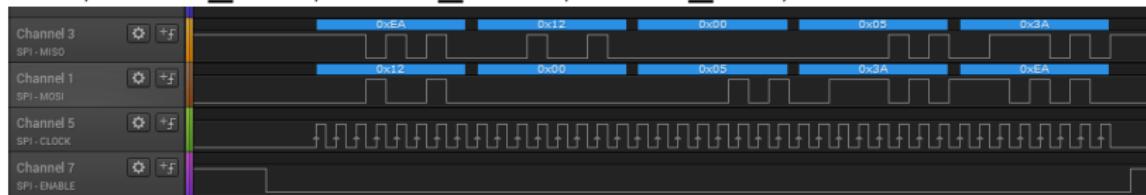
ESP8266

- Der ESP8266 agiert als SPI-Master
- Reagiert auf HTTP-Requests
- Sendet *READ/WRITE*-Befehle an ATmega809
- Sendet Ergebnis eines READ-Requests als HTML an Webbrowser

ESP8266

WRITE-Befehl

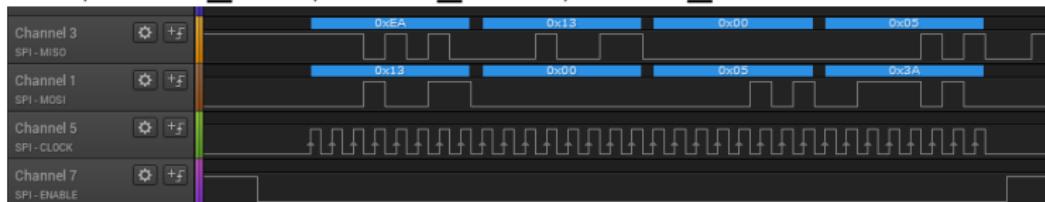
- Besteht aus
 - 0x12, ADDR_MSB, ADDR_MLSB, ADDR_LSB, DATA



ESP8266

READ-Befehl

- Besteht aus
 - 0x13, ADDR_MSB, ADDR_MLSB, ADDR_LSB



- RET_VAL



ESP8266

Web-API

- READ-Befehl
 - 192.168.4.1/get?addr=xxxxxx
- WRITE-Befehl
 - 192.168.4.1/set?addr=xxxxxx&value=yyy

Demo