

# Programmierung von ATMEL AVR Mikroprozessoren am Beispiel des ATtiny13

Eine Einführung in Aufbau, Funktionsweise,  
Programmierung und Nutzen von Mikroprozessoren

## Teil I: Was ist ein Mikrocontroller?

## Was ist ein Computer?

- Ein Computer ist eine programmgesteuerte Rechenmaschine (von eng. „to compute“ = rechnen)
- Mechanische Rechenmaschine: fest verdrahtetes Programm => Rechenablauf ist durch die Mechanik festgelegt
- Computer: Der Rechenablauf wird durch das Programm bestimmt. Wechselt das Programm, wird ein anderer Rechenablauf ausgeführt.
- Jeder Computer braucht daher:
  - eine zentrale Recheneinheit (engl. „Central Processing Unit“, CPU), die Daten holt, die Programmschritte holt (Ablaufsteuerung), rechnet und die Ein- und Ausgabedaten verteilen kann,
  - einen Programmspeicher, in dem die Rechenanweisungen (Befehle, Instruktionen) gespeichert sind, die von der CPU abgeholt und ausgeführt werden können, und
  - Ein- und Ausgabe-Einheiten (engl. „Input/Output-Units“, I/O), über die Eingangsdaten eingelesen werden können und über die Ergebnisse (Resultate) der verarbeiteten Daten ausgegeben werden können.

# Vergleich PC – Mikrocomputer - Mikrocontroller

## • PC

- CPU: Intel oder AMD, Takt: x GHz, 32/64 Bit Busbreite = 32/64 Leitungen für Befehlsadressen plus 32/64 Leitungen für Daten-Kommunikation
- Befehlsspeicher: extern, x Gigabyte, gelesen von externen Speichern (z.B. Festplatte, CD, DVD)
- I/O-Einheiten: Festplatten und andere Speichermedien, Tastatur, parallele und serielle Schnittstellen, Graphikausgabe, u.v.a.m.

• = Elefant!

## • Mikrocomputer

- CPU: Zilog etc., Takt: x MHz, 8/16 Bit Busbreite = 8/16 Leitungen für Befehlsadressen plus 8/16 Leitungen für Daten-Kommunikation
- Befehlsspeicher: extern, x kB, Festspeicher (EPROM),
- RAM-Speicher: extern, x kB, nach Bedarf
- I/O-Einheiten: parallele und serielle Schnittstellen, Graphikschnittstelle, u.v.a.m. extern zubaubar

• = Maus!

## • Mikro-controller

- CPU: diverse, Takt: 1..20 MHz, 8/16 Bit Busbreite, alle intern
- Befehlsspeicher: intern x kB, Festspeicher (EPROM/EEPROM),
- RAM-Speicher: intern x Bytes, Statisches RAM, zusätzlich nicht-flüchtiges EEPROM x Byte
- I/O-Einheiten: programmierbare Pins nach extern

• = Floh!

# Einsatzgebiete PC, Mikrocomputer, Mikrokontroller

## •PC

- große Datenmengen
- hohe Verarbeitungsgeschwindigkeit
- intensive Anwenderbedienung (graphische Bedienung)
- umfangreiche graphische Darstellung nötig
- sehr viele verschiedene Programme müssen komplex zusammenarbeiten

## •Mikrocomputer

- mittlere Daten-mengen bis x MB
- hohe Verarbeitungsgeschwindigkeit
- sehr kurze Reaktionszeit auf Ereignisse
- mittlere Anwenderbedienung (Tastatur)
- einfache graphische Darstellung nötig
- ein einziges oder wenige Programme erforderlich

## •Mikrokontroller

- kleine Datenmengen bis x kB
- hohe Verarbeitungsgeschwindigkeit
- sehr kurze Reaktionszeiten auf Ereignisse
- keine/geringe Anwenderbedienung
- einfachste graphische Darstellung (z.B. LCD)
- ein einziges Programm erforderlich

# Aufwand an Hard- und Software bei PC, MC und $\mu$ C

## •PC

- sehr aufwändige Hardware,  $x*100$  €
- Abspecken praktisch kaum möglich
- hoher Energiebedarf,  $x*100$  W
- hoher Verschleiß durch Temperatur und mechanische Teile
- Betriebssystem erforderlich
- hochkomplexe Software
- deshalb sehr hohe Fehleranfälligkeit

## •MC

- aufwändige Hardware,  $x*10$  €
- flexibel im Aufbau (Stecksysteme)
- mittlerer Energiebedarf,  $x*10$  W
- kein Verschleiß durch Temperatur, geringer Verschleiß bei mechanischen Teilen (Stecker etc.)
- gering komplexe Software, standardisierbar
- daher geringe Fehleranfälligkeit

## • $\mu$ C

- kaum Hardware-Aufwand,  $x*1$  €
- ersetzt auf einfache Weise komplizierte Elektronik/Logik
- speziell auf Bedarf abgestimmt, kein Hardware-Ballast
- geringster Energiebedarf,  $x*1$  mW
- kein Verschleiß
- einfachste Software, genau auf Bedarf zugeschnitten
- daher kaum Fehleranfälligkeit

# Beispiel Ersatz von komplizierter Elektronik: Ruftonauswerter Analog/Mikrokontroller

## •Analog

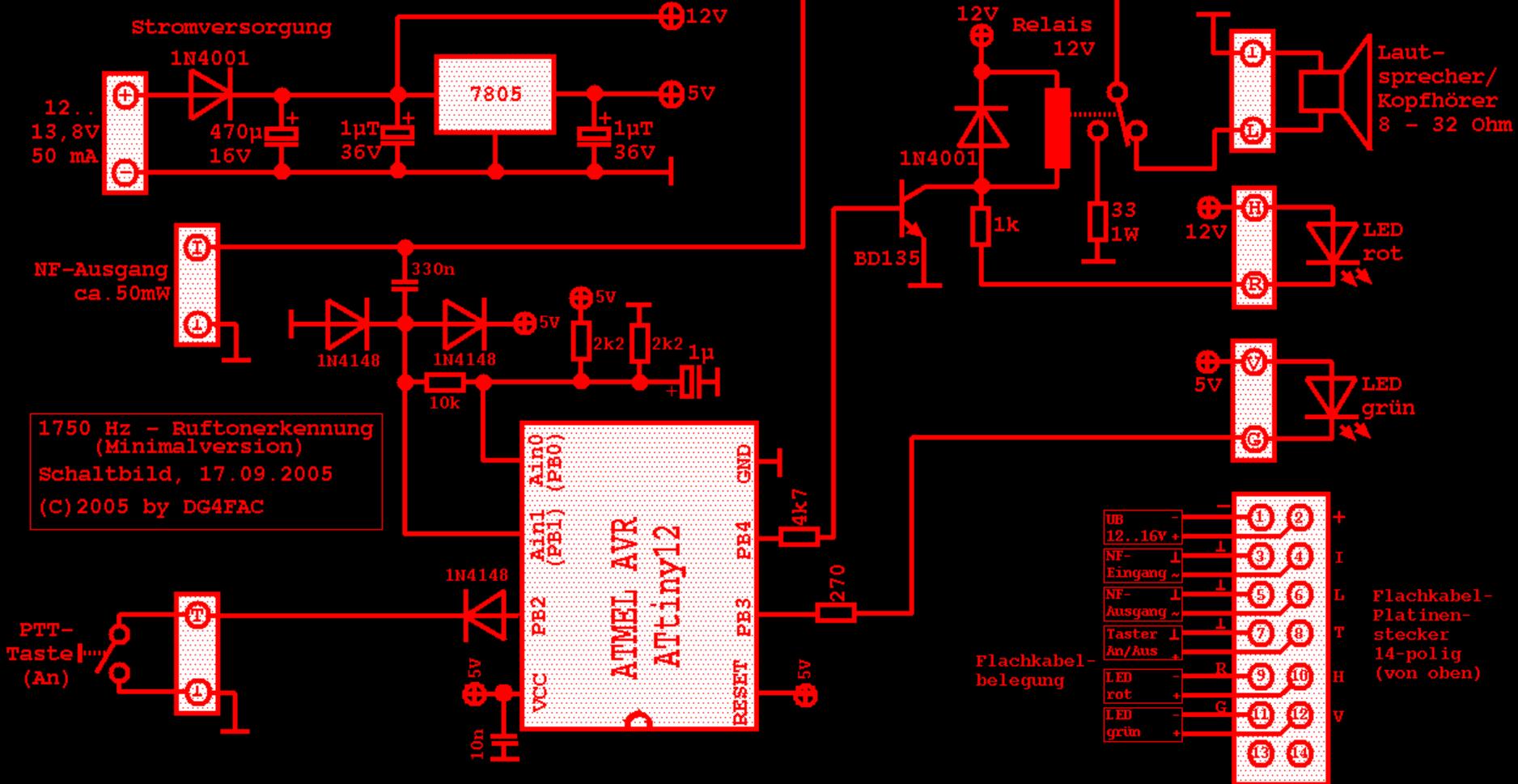
- Hardware: Tonerkennung (16-poliges IC), Zeitverzögerung (8 bis 14 poliges IC), viel Peripherie (Widerstände, Kondensatoren, Potis, etc.)
- unflexibel: Umbau auf andere Frequenzen, Toleranzen und Zeiten macht erheblichen Umbau erforderlich, Umbau nur in engen Grenzen
- anfällig: Alterung erfordert Nachjustieren, spannungsabhängig

## • $\mu$ C

- Hardware: alles in einem 8-poligen IC, wenig Peripherie
- hochflexibel: Umbau auf andere Frequenzen, Toleranzen und Zeiten macht nur Software-Änderung erforderlich, Umprogrammierung in weiten Grenzen möglich
- unanfällig: kein Nachjustieren, nicht spannungsabhängig

# Schaltbild Ruftonauswerter mit Mikrokontroller

## 1750 Hz Ruftonerkennung



# Aufgaben Ruftonauswerter

- **Erkennung:** Der Ruftonauswerter überwacht einen Lautsprecherausgang auf Töne um 1750 Hz (genau: 1700 bis 1800 Hz). Bei einem reinen Ton dieser Frequenz über eine vorwählbare Zeit (derzeit 3 Sekunden) schaltet der Ruftonauswerter das Signal für eine bestimmte Zeit lang (derzeit 30 Sekunden) zum angeschlossenen Lautsprecher durch.
- **Manuelle Schaltung:** Mit einem Taster kann das Signal dauerhaft durchgeschaltet oder wieder auf Ruftonerkennung geschaltet werden.
- **Inaktiv:** Ohne Betriebsspannung des Ruftonauswerter ist das Eingangssignal dauernd auf den Lautsprecher durchgeschaltet.
- **Abschluss:** Im nicht aktivierten Zustand ist der Signaleingang mit 33 Ohm abgeschlossen.
- **Zustandsanzeige:** Eine grüne LED zeigt an, ob
  1. kein NF-Signal vorhanden ist: 100% Intensität,
  2. ein NF-Signal vorhanden ist, aber kein Rufton vorliegt: 25% Intensität,
  3. ein Rufton mit Störsignalen vorliegt: 12,5% Intensität,
  4. ein reiner Rufton vorliegt: 6% Intensität.
- **Ausgangsanzeige:** Eine rote LED zeigt bei vorhandener Betriebsspannung an, ob der Lautsprecher abgeschaltet (LED an) oder zum Signaleingang durchgeschaltet ist (LED aus).