Klaus Bußmeyer

Der „intelligente Infrarotmelder“

Kurzbeschreibung

Manuskript

Schenefeld, April 2007

Inhalt

Abbildungsverzeichnis

Die typischen Probleme mit Infrarotmeldern

Fehlalarme

Ein normaler, kaufüblicher, so genannter Passiv-Infrarotmelder (PIR) reagiert auf Wärmestrahlung im infraroten (nicht sichtbaren) Wellenbereich. Hierbei trifft die von einem Körper (z.B. Mensch, Tier oder andere Gegenstände) ausgehende Wärmestrahlung auf die Empfängerfläche des PIR-Melders und wird dort unter Nutzung des so genannten pyroelektrischen Effektes in eine elektrische Spannung umgewandelt. Die entstehende Spannung wird entsprechend verstärkt und kann dann dazu verwendet werden, ein Schaltsignal zu erzeugen, wie z.B. das Betätigen eines Relais zum Zwecke der Schaltung eines elektrischen Kontaktes. Mit diesem Schaltkontakt wiederum kann dann z.B. ein elektrischer Alarmgeber eingeschaltet werden.

Zur Verbesserung der Reaktion auf *bewegte* strahlende Körper ist die Empfängerfläche des PIR-Melders in mehrere Sektoren unterteilt. Wenn die Wärmestrahlung eines *bewegten* Körpers auf den PIR-Melder trifft, dann werden in den verschiedenen Sektoren nacheinander wechselnde elektrische Spannungen festgestellt. Diese wechselnden Spannungen werden entsprechend ausgewertet und bei Überschreiten eines gewissen Schwellwertes in ein Alarmsignal umgesetzt.

Der PIR-Melder reagiert also vorzugsweise auf *bewegte* strahlende Körper, und wegen seiner Sektorenaufteilung vorzugsweise auf solche, die sich *quer* zu seiner Empfangsfläche bewegen.

Die weitaus meisten PIR-Melder sind nicht zur Alarmmeldung im engeren Sinne konstruiert, sondern eher zum Komfort, z.B. um nachts ein Hoflicht einzuschalten. Diese PIR-Melder geben dann nach den oben genannten Kriterien ein Schaltsignal, welches das z.B. das Hoflicht einschaltet. Die Dauer dieser Einschaltung ist häufig am PIR-Melder einstellbar, d.h. sie wird auch noch nach Verschwinden des strahlenden Körpers für eine bestimmte Dauer aufrecht erhalten. Häufig ist auch noch ein Dämmerungsschalter integriert, so dass das Hoflicht wirklich nur dann eingeschaltet wird, wenn ein bestimmter Dämmerungszustand erreicht ist.

Diese bei den meisten kaufüblichen PIR-Meldern gegebenen Charakteristika sind nicht unbedingt für einen Alarmgeber im engeren Sinne besonders wünschenswert. Einerseits soll ein PIR-Alarmgeber nicht nur in der Dämmerung oder bei Nacht Alarm auslösen können. Des weiteren sind sowohl die beschriebene Schaltschwelle wie auch die Einschaltdauer des PIR-Melders nicht unbedingt die ideale Vorbedingung für eine qualifizierte Alarmgabe. So kann z.B. auf Grund der eben genannten Kriterien das Hoflicht fälschlicherweise auch dann eingeschaltet werden, wenn in einer Sturmnacht warme Luftturbulenzen oder erwärmtes Laub den PIR-Melder zur Auslösung bringen, eine Verhaltensweise, die häufig in der Praxis beobachtet werden kann. Solch eine Verhaltensweise ist für den Einsatz einer gezielten Personenerkennung völlig unakzeptabel.

Hier setzen Überlegungen zur Verbesserung des Alarmverhaltens an, wie sie später noch im einzelnen vorgestellt werden sollen.

Verkabelungsprobleme

Sofern ein PIR-Melder z.B. ein Hoflicht einschalten soll, ist die Verlegung eines Elektrokabels dorthin natürlich akzeptabel, weil das Hoflicht ohnehin einen 230-V-Anschluss für den Betrieb der Glühbirne benötigt.

Wenn allerdings ein PIR-Melder zum Zwecke der Auslösung von Alarmen an anderer Stelle, z.B. irgendwo im Hause, eingesetzt werden soll, dann ist die dazu normalerweise notwendige Verkabelung oft sehr störend, weil damit z.B. Stemmarbeiten oder gar Tapezierarbeiten im Hause notwendig sein können.

Die sich hieran anschließenden Überlegungen münden oft in dem Wunsch nach einer Verbindung zwischen dem Alarmgeber und dem Alarmempfänger über eine Funkstrecke, wie sie später ebenfalls noch im Einzelnen vorgestellt werden sollen.

Der „intelligente Infrarotmelder“ als Lösungsansatz

Das Funktionsprinzip

Ausgehend von den eben beschriebenen Unvollkommenheiten der normalen herkömmlichen Infrarotmelder soll hier ein Wunschkatalog entwickelt werden, welche Verbesserungsmaßnahmen für den Zweck der Personenerkennung wünschenswert wären.

Stellen wir uns als Beispiel eine Haustür vor, die vor Einbruch mit einem Infrarotmelder gesichert werden soll. Es sind grundsätzlich zunächst zwei verschiedene Kategorien von Personen, die durch diese (verschlossene) Haustür gelangen möchten: diejenigen Personen, die (legitimerweise) im Besitz eines Schlüssels sind (und somit die Haustür mit wenig Zeitaufwand passieren können), und diejenigen, die zum Zwecke des Einbruchs vor der Haustür eine gewisse Zeit verbringen müssen, um die Öffnung der Tür zu bewerkstelligen.

Der Unterschied zwischen diesen beiden Kategorien von Türbenutzern, angewendet auf die Möglichkeiten eines PIR-Melders, ist also der unterschiedliche Zeitbedarf für das Öffnen der Tür. Hier böte sich ein Ansatz zur Lösung, wenn der eigentliche Meldealarm erst dann ausgelöst werden würde, nachdem eine bestimmte Zeit verstrichen ist, in der der Infrarotmelder schon mehrere Alarme gegeben hat, die aber nur „im Stillen“ aufsummiert worden sind. Erst wenn eine bestimmte Summe von Alarmen des Infrarotmelders aufsummiert worden ist, soll es zum eigentlichen Alarm kommen, wobei natürlich Bedingung sein muss, dass diese Alarmsummierung innerhalb einer festen (relativ kurzen) Zeit zustande gekommen ist. Wir wollen

diese Wartezeit, in der es zunächst nicht zur Auslösung des Meldealarms seitens der Alarmanlage kommt, die „Stillezeit“ nennen.

Sollte also innerhalb dieser Stillezeit die Summe der erforderlichen Infrarotalarme *nicht* aufsummiert werden, dann wäre die Bedingung für einen Meldealarm seitens der Alarmanlage *nicht* gegeben. Die Stillezeit wäre dann zu Ende, und es würde ein neuer Beobachtungszeitraum beginnen (so, als ob nichts geschehen wäre).

Da natürlich jedermann bekannt ist, dass Infrarotmelder auf bewegte Objekte reagieren, wird ein Eindringling, wenn er denn von einem Alarm überrascht wird, sicherlich erst einmal in Ruhe verharren (oder sich schleunigst vom Grundstück weg bewegen). In diesem Sinne wäre es wünschenswert, wenn der einmal ausgelöste Alarm ein gewisse Zeit fortdauert, unabhängig von der Reaktion des Eindringlings. Wir wollen diese Zeit die „Sirenenzeit“ nennen. Der Eindringling sollte also den Eindruck gewinnen, dass der Alarm unabhängig von seiner Verhaltensweise fortdauert.

Aber unabhängig von der Reaktion des Eindringlings (Bleiben oder Entfernen) wird man nach einer gewissen Alarmzeit den Alarm abstellen müssen, und sei es nur aus Gründen der Rücksicht auf die Nachbarschaft. Es erhebt sich die Frage nach der weiteren Reaktion. Ein Vorschlag könnte hier sein, für einer gewisse Zeit eine Alarm-Pause einzulegen, d.h. dass in einer gewissen Folgezeit kein Alarm gegeben wird, auch dann nicht, wenn der Infrarotmelder einen Alarm meldet. Wir wollen dies die „Pausenzeit“ nennen.

Nach Verstreichen dieser Pausenzeit sollte dann das Alarmsystem dann wieder in die Grundstellung gehen und somit wieder für die Auswertung weiterer Alarme vom Infrarotmelder bereit sein.

Zusammengefasst würde die Alarm-Logik also nach folgendem Schema ablaufen:

1. Status „Grundstellung“.

   Nach dem Einschalten des Alarm-Systems befindet sich dieses in einem Status „Grundstellung“, d.h. es wartet auf Alarme vom Infrarotmelder. Falls diese nicht eintreffen, verbleibt das Alarm-System in diesem Status.

2. Status „Stille“.

   Falls nun ein Alarm vom Infrarotmelder eintrifft, wird das Alarm-System nicht gleich eine Sirene auslösen, sondern in den Staus „Stille“ wechseln. Dieser Status dauert eine gewisse definierte Zeit. Falls in dieser definierten Zeit der Infrarotmelder eine bestimmte definierte Menge von Alarmen liefert, wird das Alarm-System einen Sirenenalarm auslösen und in den Status „Sirene“ wechseln.

   Falls in dieser Zeit die Menge der definierten Alarme vom Infrarotmelder *nicht* geliefert wird, wechselt das Alarm-System zurück in den Status „Grundstellung“.

3. Status „Sirene“.

   Der Status „Sirene“ dauert eine gewisse definierte Zeit, unabhängig davon, ob der Infrarotmelder Alarme liefert oder nicht. Nach Ablauf dieser Zeit wechselt das Alarm-System über in den Status „Pause“.

4. Status „Pause“.

   Der Status „Pause“ dauert ebenfalls eine bestimmte definierte Zeit. Während dieser Zeit werden Alarme vom Infrarotmelder ebenfalls ignoriert, d.h. das Alarm-System löst in dieser Zeit keinen weiteren Alarm aus.

   Nach Ablauf dieser Pausenzeit wechselt das Alarm-System zurück in den Status „Grundstellung“, und der Kreislauf ist geschlossen.

Natürlich ist es wünschenswert, dass die oben genannten Zählwerte und Zeiten zumindest in Grenzen frei definierbar sind, so dass eine Anpassung an gegebene Verhältnisse ermöglicht werden kann. Weiterhin sollte es möglich sein, bei hochsensiblen Überwachungssituationen bestimmte Status zeitlich gleich Null werden zu lassen bzw. zu überspringen.

Obwohl die Logik des bis hierher beschriebenen Alarm-Systems den landläufigen Anforderungen durchaus entsprechen mag, so ist doch ein wesentlicher Aspekt bisher noch nicht berücksichtigt, und das ist die Störsicherheit. Jedes Alarm-System, auch dieses, kann Fehlalarme auslösen, und diese Fehlalarme untergraben zunächst einmal die Glaubwürdigkeit oder Verlässlichkeit das Alarm-Systems. Darüber hinaus können Fehlalarme sich auch sehr störend auf die Nachbarschaft auswirken. Beides, die Glaubwürdigkeit und Störung der Umgebung, müssen zuverlässig vermieden werden. Natürlich kann dies die gewünschte Sicherheit des Systems beeinträchtigen. Hier muss also in der Praxis ein vernünftiger Kompromiss gefunden werden.

Das hier vorgestellte Alarm-System versucht, diesem Kompromiss nahe zu kommen, indem es dem oben beschriebenen, in sich geschlossenen Status-Kreislauf der Alarmbearbeitung eine übergeordnete Logik hinzu fügt.

1. Status „Störüberwachung“.

   Wenn in der bisher beschriebenen Status-Logik einmal der Status „Sirene“ aufgetreten ist, dann wird unabhängig vom bisherigen Geschehen der Status „Störüberwachung“ eingeschaltet. Der Status „Störüberwachung“ dauert eine bestimmte definierte Zeit, die groß genug sein muss, um mehrere komplette Phasen mit Sirenenalarm zu überdauern. Wenn in dieser definierten Störüberwachungszeit die Zahl der Sirenenalarme eine bestimmte definierte Zahl übersteigt, dann geht das Alarm-System von einer nachhaltigen Störung aus und schaltet um auf den noch zu beschreibenden Status „Störung“.

   Falls in der Störüberwachungzeit die definierte Anzahl von Sirenenalarmen *nicht* erreicht wird, dann wird der Status „Störüberwachung“ wieder ausgeschaltet. Das Alarm-System verhält sich ab diesem Zeitpunkt so, als ob kein Sirenenalarm aufgetreten wäre.

2. Status „Störung“.

   Dieser Status trifft wie oben gesagt dann ein, wenn innerhalb der Störüberwachungszeit eine definierte Anzahl von Sirenenalarmen überschritten worden ist. Die vorgegebene Zeit für diesen Status ist definiert und sollte so dimensioniert sein, dass das Alarm-System die Nachbarschaft nicht weiter stört. Dies könnte u.U. eine Dauer-Abschaltung mit einschließen. Während dieses Status wird natürlich kein Alarm vom Infrarotmelder berücksichtigt. Das Alarm-System ist also während dieser Zeit stumm.

Nach Ablauf dieser Störungszeit wird das Alarm-System dann wieder in „Grundstellung“ versetzt, so dass das Gesamtsystem dann wieder so reagiert, als wäre es gerade neu eingeschaltet geworden.

Die modulare Implementierung des Alarm-Systems

Der programmierbare Mikroprozessor ATmega 168

Die Wahl des verwendeten Mikroprozessors ATmega 168 von Atmel hatte im wesentlichen folgende Gründe:

- **Der Preis**
  von nur € 3,20 ist ein guter Preis für die Leistung, die der ATmega 168 bietet. Was die Leistung betrifft, so waren folgende Kriterien maßgeblich:

- **Prozessortakt bis 20 MHz**
  Hiermit ist der Prozessor für diese Art der Anwendung allemal schnell genug.

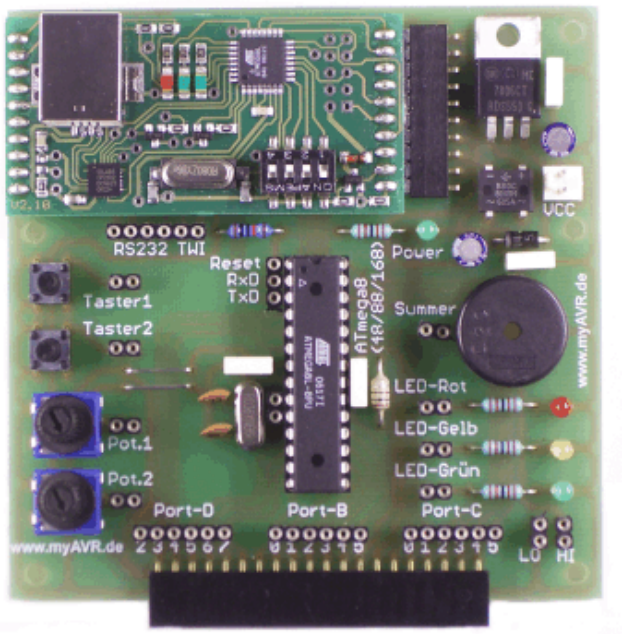

Abbildung 1
ATmega Chip,
montiert auf einem SiSy Experimentierboard
der Firma Laser &Co Solutions GmbH

- **16k Flash-Speicher**
  Im Flash-Speicher wird das verwendete selbst geschriebene Programm abgelegt. 16k Flash sind für gegebene Programm-Größe ausreichend. Der Flash-Speicher hat gegenüber anderen

Speicherarten den großen Vorteil, dass sein Inhalt auch bei einem Stromausfall erhalten bleibt. Der Prozessor kann also ohne weiteres nach erneutem Stromeinschalten seine Arbeit wieder aufnehmen.

- **1024 Byte SRAM-Speicher**
  Der SRAM-Speicher (Static Random Access Memory) ist der Arbeitsspeicher des Prozessors. Hier befinden sich im wesentlichen die Rechenregister, die Adressregister und die Ein-/Ausgaberegister. Auch dieser Speicher reicht in Bezug auf Größe und Schnelligkeit für die Anwendung vollkommen aus.

- **28 programmierbare Ein-/Ausgabeleitungen**
  Der Prozessor kommuniziert mit seiner Umgebung über die Ein-/Ausgabeleitungen. Die Umgebung besteht in diesem Anwendungsfall zunächst einmal aus einem Anschluss an den PC. Über diesen Anschluss wird während der Programmentwicklung das Programm in den Prozessor geladen. Ferner kann auf Wunsch während des Betriebs das Meldegeschehen des Alarm-Systems über denselben Anschluss auf dem Monitor des PC verfolgt werden. Zusätzlich können über diesen Anschluss bei Bedarf die „stillen Aufzeichnungen“ über geschehene Alarme von dem dafür vorgesehenen Zusatz-Chip auf den PC übertragen werden und dort auf dem Monitor dargestellt werden.

  Zur Umgebung des Prozessors sind weiter die angeschlossenen Infrarotmelder zu zählen. Da ein Anschluss von bis zu vier Infrarotmeldern vorgesehen ist, werden entsprechend vier Ein-/Ausgabeleitungen des Prozessors belegt.

  Weiter zählen zur Umgebung des Prozessors noch der Anschluss von weiteren Zusatz-Chips. Je nach Wahl der weiteren Zusatzeinrichtungen sind dies zwei weitere Ein-/Ausgabeleitungen, die diese Zusatz-Chips mit dem Prozessor verbinden. Für diesen Zweck bietet der Prozessor ein spezielles Bus-System:

- **I2C-Bus**
  Der I2C-Bus ist ein spezielles, von Philips entwickeltes Kommunikationsprotokoll, welches vom ATmega 168 ebenfalls unterstützt wird. Beim I2C-Bus reichen zwei Drahtverbindungen zwischen dem Prozessor und den angeschlossenen Zusatz-Chips aus, um eine serielle Kommunikation zwischen den Komponenten zu ermöglichen. Im Falle der hier diskutierten Anwendung mit Infrarotmeldern können über diesen I2C-Bus auf Wunsch ein Speicher-Chip für „stille Alarm-Aufzeichnungen“, sowie ein Speicher-Chip für eine Echtzeituhr angeschlossen werden.

- **Externer Interrupt**
  Eine ganz wesentliche Erfordernis bei Echtzeit-Anwendungen wie in Alarm-Systemen ist die Fähigkeit des Mikroprozessors, augenblicklich auf z.B. Meldesignale von den Infrarotmeldern zu reagieren und diese zu bearbeiten. Hierzu ist die Architektur des Prozessors so angelegt, dass beim Eintreffen eines solchen Signals aus der Umgebung der Prozessor seine Arbeit, die er gerade ausführt, unterbricht, um die Auswertung des externen Alarmsignals augenblicklich in Arbeit zu nehmen. Hierzu bedarf es entsprechender Vorrichtungen innerhalb der Prozessor-Logik. So muss z.B. der Status der momentanen Arbeit, in der sich der Prozessor befand, gespeichert werden, damit der Prozessor später, nachdem die bevorzugte Arbeit an dem Unterbrechungsereignis erledigt ist, wieder die vorher unterbrochene Arbeit nahtlos an der Stelle aufnehmen kann, an der sie unterbrochen worden war. Der ATmega 168 unterstützt mit seiner Architektur diese Anforderung auf ganz hervorragende Weise.

- **ISP (In-System Programming)**
  Die Programmierung des ATmega 168 findet mit Hilfe entsprechender Editoren, Compiler und Lader zunächst auf dem PC statt. Die Programmiersprache, die in diesem Anwendungsfall verwendet wird, ist das BASCOM AVR der Firma MCS Electronics aus Holland. BASCOM AVR ist ein Dialekt der weit verbreiteten Programmiersprache BASIC, der speziell für die Gegebenheiten der Mikroprozessoren von Atmel entwickelt worden ist. Konsequenterweise besitzen die Prozessoren von Atmel auch eine Einrichtung, die es gestattet, ein neues, auf dem PC entwickeltes Programm auf den Prozessor herunter zu laden, und zwar auf eine Weise, bei der der Prozessor nicht aus seiner Schaltungsumgebung herausgelöst zu werden braucht. Dies bezeichnet man als ISP (In-System Programming). Man kann sich also den Vorgang der Programmentwicklung so vorstellen, dass der „große Bruder" PC in dieser Phase ständig mit dem Mikroprozessor verbunden ist (z.B. über USB-Kabel), und dass man als Entwickler vom PC her die gesamte Programmentwicklung, die Programmausführung und die Fehlersuche für den Mikroprozessor durchführen kann. Was die vielen Vorteile dieser Kombination betrifft, so bedenke man nur einmal, dass normalerweise bei einem Mikroprozessor-System kein Bildschirm zur Verfügung steht ! Fehlersuche in einem Mikroprozessor-Programm ohne die Möglichkeit der optischen Darstellung der Inhalte ist allein schon ein Ding der Unmöglichkeit !

Zusammenfassend kann also gesagt werden, dass angesichts dieser vergleichsweise gewaltigen Möglichkeiten, die der ATmega 168 zum Preis von € 3,20 bietet, die Wahl dieses Prozessors sicherlich keine bessere sein kann.

Die implementierte „Alarm-Software"

Da die im Kapitel „Der intelligente Infrarotmelder als Lösungsansatz" vorgestellte Idee eine Eigenentwicklung war, war auch von vornherein klar, dass die Umsetzung dieser Idee in ein funktionierendes Programm selbst „mit der Hand am Arm" durchgeführt werden muss.

Hier taucht natürlich für den Laien die Frage auf, wie viel Zeit für die Entwicklung eines Programms von diesem Umfang so etwa aufgewendet werden muss. Diese Frage ist zumindest allgemein schwer zu beantworten, denn das hängt natürlich vom Vorwissen und der Erfahrung des Verfassers ab. Als Verfasser kann ich hierauf nur eine persönliche Antwort geben, die freilich logischerweise auf dem eigenen Vorwissen und der eigenen Erfahrung beruht.

In meinem Falle war die Sachlage so, dass ich zwar über jahrelange berufliche Erfahrungen in Sachen Datenverarbeitung und auch Programmierung verfügte. Andererseits war dies aber mein erstes Projekt mit dem für mich bisher nicht bekannten Prozessortyp Atmega.xx und auch die erste Berührung mit der verwendeten Programmiersprache BASCOM.

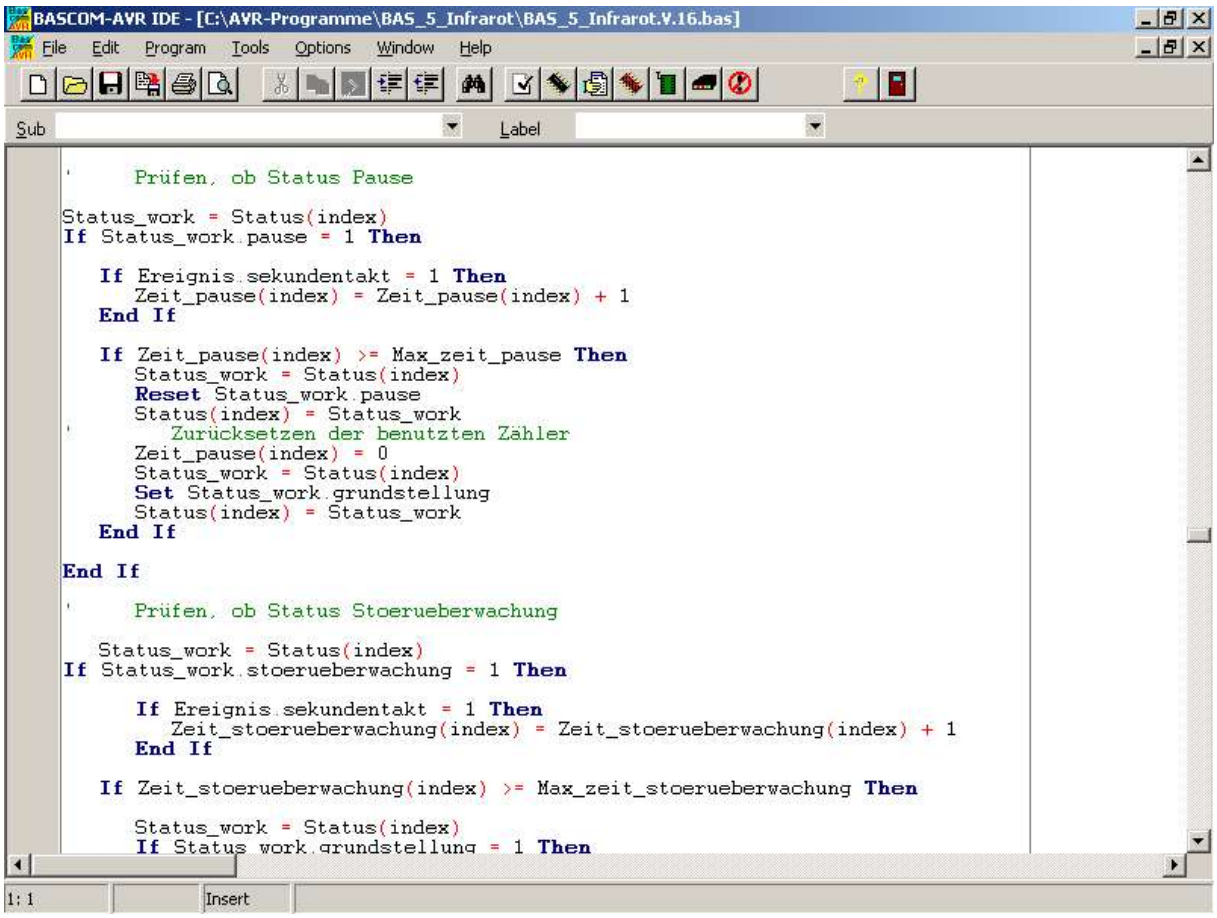

Abbildung 2
Ausschnitt aus der „Alarm-Software“, programmiert in Bascom

Das Endprodukt, also das fertige, voll funktionsfähige Programm, welches die Signale von vier voneinander unabhängigen Infrarotmeldern gleichzeitig zu bearbeiten gestattet, besteht aus 20 Seiten Programmanweisungen einschließlich Kommentaren mit etwa 50 Zeilen pro Seite. Der fertige, auf dem Prozessor ausführbare Code besteht aus ca. 9k Byte Instruktionen und Daten. Für die Erstellung und das Testen habe ich etwa drei Monate (ca. 3 Stunden pro Tag) Arbeitszeit gebraucht. Dabei geschah die Entwicklung der Endversion schrittweise aus fünfzehn aufeinander aufbauenden Vorversionen. Man mag das als sehr viel empfinden. Aber hier möchte ich dagegen halten, dass dies ein Zeitaufwand ist, den viele von uns täglich vor dem Fernseher verbringen, und dass es auch ein großes Erfolgserlebnis ist, wenn man den Einstieg in solch eine komplexe Materie schließlich gemeistert hat.

Zum Programm selbst ist zu sagen, dass es im wesentlichen nach zwei Regeln der Informationstheorie verfasst worden ist, und das ist einerseits die „Finite-State-Machine“ (FSM)- Theorie, die eine bestimmte Reihenfolge von Status-Zuständen darzustellen gestattet, und anderseits die Interrupt-Technik, die die bevorzugte Behandlung von Echtzeit-Ereignissen zu implementieren gestattet.

Gegenüber den im Handel verfügbaren, fertig programmierten Produkten bietet eine Selbst-Entwicklung den unschätzbaren Vorteil, dass sie „open ended“ ist. Der Verfasser des Quellcodes ist jederzeit in der Lage, das vorhandene Programm um nahezu jede beliebige Zusatzfunktion zu erweitern.

Hier sei nur ein kleines Beispiel genannt: Auf Grund der Anwendungssituation möchte der Anwender vielleicht die Alarmdaten, die „still“ erfasst und gespeichert worden sind, nicht auf einem PC, sondern direkt auf einem Display darstellen, welches direkt am Mikroprozessor angeschlossen ist (z.B. ein LCD-Display). Das existierende Programm dahin gehend zu erweitern ist eine leichte Übung: es wird einfach an den schon vorhandenen I2C-Bus ein entsprechendes Display angeschlossen, welches das I2C-Protokoll beherrscht (im Handel verfügbar). Das vorhandene Programm muss dann nur noch dahingehend erweitert werden, dass die gesammelten Daten statt auf dem PC nun auf diesem Display angezeigt werden.

Die Möglichkeit der Programm-Modifikation ist also der Schlüssel zu der beliebigen Erweiterung und Änderung des vorhandenen Programm-Moduls und somit zur höchstmöglichen Flexibilität, wenn es um Anpassungen an individuelle Anwendungssituationen geht.

Das Funk-Haussteuerungs-System FS20 der Firma ELV Elektronik AG

Mit ihrem Funk-Haussteuerungs-System FS20 hat die Firma ELV AG in Leer ein Sortiment von vielen Einzelkomponenten für Steuerungsaufgaben im häuslichen Bereich entwickelt, das vom Funk-Dimmer für die Stehlampe über die Klimasteuerung des Hauses bis hin zu Anwendungen wie der automatischen Markisensteuerung oder einer Sprachausgabe reicht, wenn Post in den Briefkasten geworfen worden ist. Alle diese Komponenten sind miteinander kombinierbar und basieren auf einem gemeinsamen Funk-Protokoll, welches das dafür vorgesehene 868-MHz-Band nutzt.

Natürlicherweise müssen Haussteuerungs-Systeme die Bedingung erfüllen, dass sie sich (wegen der geographischen Nähe) nicht gegenseitig stören, mit anderen Worten, das System muss zuverlässig vermeiden, dass z.B. ein Funk-Signal für die Öffnung der eigenen Garage nicht die Garage des Nachbarn öffnet. In einem sehr schmalen Funk-Band wie dem 868-MHz-Band kann dies nur durch ein konsequent eingehaltenes Adressierungsprotokoll erreicht werden. Dies wird durch ein gesetzlich vorgeschriebenes Normungsprotokoll sichergestellt.

Folglich werden in diesem Funkprotokoll nicht einfach Schaltbefehle gesendet, sondern es wird jeder Schaltbefehl immer jeweils mit einer Komponentenadresse gesendet, die das Zielgerät eindeutig identifiziert. Die Komponentenadresse ist hierbei zum Zweck einer sinnvollen Organisation in verschiedene Teile unterteilt. Der erste, sehr wesentliche Unterscheidungsteil ist der so genannte Hauscode. Er soll sicherstellen, dass z.B. Nachbarn die gleichen Gerätschaften installieren können ohne dass sich die Schaltgeräte gegenseitig „ins Gehege“ kommen können. Der Hauscode bietet mathematisch betrachtet $4^8 = 65536$ Einstellmöglichkeiten, was in der Praxis ausreichend sein dürfte, Störungen im Bereich der Reichweite des Funksignals zu vermeiden. Da die Werksauslieferungen mit zufällig eingestelltem Hauscode erfolgen, dürfte sichergestellt sein, dass gegenseitige Störungen in der Praxis selten sind. Zusätzlich bieten die Geräte die Möglichkeit, den Hauscode vor Ort individuell einzustellen.

Um innerhalb der unter einem Hauscode installierten Geräte unterscheiden zu können, sind mathematisch betrachtet 256 Einzeladressierungsmöglichkeiten gegeben. Es würde hier zu weit führen, die weitere Strukturierung der Einzeladressen im Detail zu diskutieren. Es sei hier aber so viel gesagt, dass eine weitere Strukturierung der Einzeladressen in Funktionsgruppen möglich ist, und dass diese Struktur insgesamt 225 Einzeladressen gestattet, was in der Praxis im häuslichen Bereich allemal ausreichend sein sollte. Näheres ist der FS20-Literatur zu entnehmen.

Der Funk-Infrarot-Bewegungsmelder FS20 PIRA

Aus der wie gesagt recht breit angelegten FS20-Familie sei hier eine Komponente herausgegriffen, die in ganz besonderem Maße die hier zu diskutierenden Anforderungen an einen Infrarotmelder erfüllt. Der Funk-Infrarot-Bewegungsmelder FS20 PIRA erfüllt die hier zu stellenden Anforderungen insbesondere deshalb, weil er

- durch seinen Batteriebetrieb nicht auf eine Verkabelung mit der Stromversorgung angewiesen ist und somit eine Menge von Installationsaufwand und -kosten erspart, weil er
- im Außenbereich eingesetzt werden kann wegen seiner Wasserdichtheit, weil er durch seine
- Kleinheit sehr unauffällig montiert werden kann, und weil er schließlich wegen seines
- vorhandenen Funk-Protokolls in eine bestehende Mikroprozessor-Anwendung leicht integriert werden kann.

Abbildung 3
FS20 PIRA Funk-Bewegungsmelder
der Firma ELV AG,
Quelle: ELV Hauptkatalog

Dieser Infrarotmelder kann seine Alarmsignale via Funkverbindung im 868-MHz-Band an einen weiteren Baustein der FS20-Familie übertragen, dem so genannten 4-Kanal-Schaltmodul. Dieses Modul empfängt auf der einen Seite die Funksignale von diesem Infrarotmelder und setzt sie auf der anderen Seite in Schaltsignale um, deren Norm der Mikroprozessor-Technik entspricht, so dass ein direkter Anschluss an einen Port des Mikroprozessors möglich ist (open-collector interface). Darüber hinaus bietet das 4-Kanal-Schaltmodul

wegen der bereits diskutierten Adressierungstechnik den Anschluss von bis zu vier Infrarotmeldern an eben dieses Schaltmodul und macht damit diese Kombination zu einem absoluten Favoriten für die hier zu diskutierende Anwendung.

Außerdem kann in Ergänzung gesagt werden, dass diese Kombination zu einem akzeptablen Preis-Leistungsverhältnis zu haben ist. Jeder, der vor der Aufgabe stand, einen am Netz anzuschließenden Infrarotmelder im Außenbereich installieren zu müssen, mit den in der Regel damit verbundenen Stemm- und Verlegungsarbeiten, wird diesen Vorteil zu schätzen wissen. Die Kombination bestehend aus Infrarotmelder und Schaltmodul kostet bei ELV ca. 60,- €.

Der Funk-Signalgeber FS20 USB

Der Funk-Signalgeber FS20 USB arbeitet als akustische Alarmausgabe und kann ebenfalls über eine Funkverbindung (und eine zusätzlichen Komponente) an den Mikroprozessor angeschlossen werden. Das ist an jedem Ort innerhalb des Hauses möglich, an dem sich eine Steckdose befindet. Da das Gerät in jede Netzsteckdose gesteckt werden kann, ist es möglich, dass der Anwender das Gerät überall dort anschließen kann,wo er sich gerade im Hause befindet.

Der Signalgeber bietet einiges an Komfort, was allerdings auch einen stolzen Preis kostet (ca. 115,- € zusammen mit dem Batterie-Unterputz-Sender FS20S4UB, letzterer gestattet es, das Alarmsignal vom Mikroprozessor per Funk an den Signalgeber zu übertragen). So ist es z.B. möglich, akustische Ausgaben in Form von Melodien oder auch gesprochenem Text vorzunehmen. Zu diesem Zweck wird das Gerät via USB an einen PC angeschlossen. Mit der zugehörigen Software kann man dann bis zu acht Dateien im WAV-Format auf das Gerät überspielen. Auf diese Weise kann man diese Melodien oder Sprachtexte entsprechend den Alarmquellen zuordnen und im Alarmfall abspielen. Dies ermöglicht es dem Hörer, den Alarm einer Quelle zuzuordnen (z.B. durch den Sprachtext: „Jemand ist der Haustür“). Ferner verfügt das Gerät über eine Blinklampe, die mit dem Alarm auf „Blinken“ geht. Wenn der Benutzer das Blinklicht mit einem Knopfdruck quittiert, wird die akustische Ausgabe nochmals wiederholt, so dass ein nicht sofort gehörter Alarm auch nachträglich zugeordnet werden kann.

Wem dieses Gerät zu teuer ist, der kann auch auf einfachere und preiswertere Geräte zurückgreifen, die ebenfalls, - teils auch als Bausätze -, von der ELV AG angeboten werden. Im übrigen muss ein Signalgeber für den akustischen Alarm ja auch nicht unbedingt über eine Funkstrecke angeschlossen werden. Eine Haustürklingel aus der Bastelkiste, direkt an den Mikroprozessor angeschlossen, tut es unter Umständen auch.

Die Echtzeituhr für Datum und Uhrzeit

Wenn eine Alarmanlage, wie im nächsten Kapitel beschrieben, außer einer unmittelbaren oder aber auch anstatt einer unmittelbaren Signalgabe das Alarmereignis für eine spätere Auswertung aufzeichnen soll, dann ist es erforderlich, dass Datum und Uhrzeit dem aufzeichnenden Mikroprozessor zur Verfügung stehen. Hier gibt es grundsätzlich zwei verschiedene Lösun-

gen. In der ersten Lösung werden Datum und Uhrzeit im Mikroprozessor selbst durch ein entsprechendes Zählprogramm aus dem Mikroprozessor-Takt abgeleitet. Alle älteren Lösungen basierten auf dieser Möglichkeit der Eigenprogrammierung. Die wesentlichen Nachteile dieser Lösung waren die rechte aufwändige Programmierung (man denke an die Monate mit 30 bzw. 31 Tagen und die Schaltjahre) sowie auch die Gefahr des „Verlustes“ der Uhrzeit, wenn der Mikroprozessor auch nur für kurze Zeit stromlos war (ältere Digitaluhren hatten dasselbe Problem). Bei der zweiten, neueren Lösung werden diese Nachteile vermieden, indem man auf Bausteine zurückgreift, die speziell für diese Anwendung geschaffen worden sind. Jeder moderne PC hat heute solch einen Baustein integriert, und deshalb stehen Datum und Uhrzeit jederzeit zur Verfügung, auch dann, wenn der PC abgeschaltet worden war.

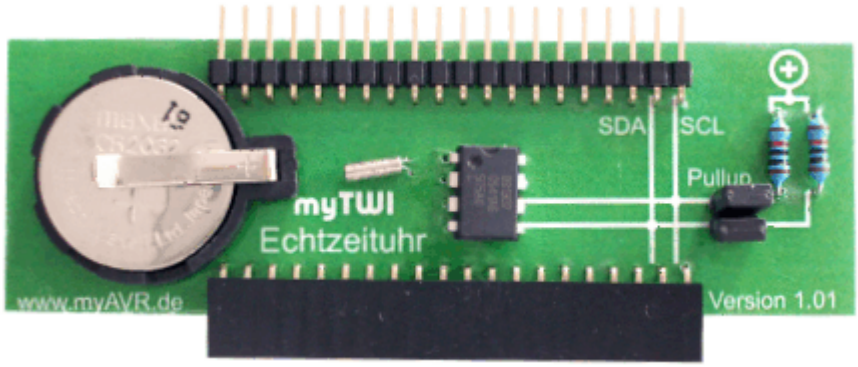

Abbildung 5
Echtzeituhr mit Stützbatterie,
montiert auf einem SiSy Experimentierboard
der Firma Laser & Co Solutions GmbH

Diese Bausteine, auch „Real Time Clock“ (RTC) genannt, arbeiten völlig unabhängig vom Mikroprozessor. Sie haben eine eigene Stromversorgung über eine Langzeit-Batterie und leiten wie bei einer Quarzuhr den Takt von der elektrischen Schwingung eines Quarzes ab. Die komplette Programmlogik ist auf dem eigenständig arbeitenden Chip enthalten. Sie gestattet das Auslesen von Datum und Uhrzeit über einen Anschluss an den Mikroprozessor, wobei hier mittlerweile sehr oft das bereits erwähnte I2C-Protokoll zur Anwendung kommt.

Ein typischer Vertreter dieser Produkt-Kategorie ist der Baustein DS1307 von Maxim-Dallas, der für einen Preis von ca. 5,- € im Handel zu haben ist. Dieser Baustein wird in der hier beschriebenen Schaltung verwendet und ist über das I2C-Protokoll an den Atmega 168 angeschlossen.

Die “stille Aufzeichnung“ von Alarmen auf nicht-flüchtigem Speicher

Wie bereits erwähnt, ist die „stille Aufzeichnung“ bei einer Alarmanlage eine Option, von der man im Bedarfsfall Gebrauch machen kann, die aber nicht unbedingt in jedem Fall erforderlich ist. „Stille Aufzeichnungen“ können z.B. dann interessant sein, wenn man daran interessiert ist, wann und wie oft z.B. ein Privatbereich während der eigenen Abwesenheit von anderen Personen betreten wird. Ähnliches kann aber auch z.B. für die Tierbeobachtung interes-

sant sein. In diesen Fällen wird man den Infrarotmelder zwar dazu benutzen, eine Bewegung nachzuweisen, aber man wird auf eine Alarmauslösung verzichten und anstelle dessen lediglich eine „stille Aufzeichnung“ des Ereignisses vornehmen.

Für den Techniker stellt sich nun die Frage, auf was für einem Medium diese Aufzeichnungen sinnvollerweise vorgenommen werden sollten. Früher geschah das z.B. auf einem Tonband, oder, falls ein PC zur Verfügung stand, auf einem externen Speichermedium wie z.B. der Diskette. Heute bietet die Elektronik-Industrie für diesen Zweck spezielle Chips an, die man Eeprom (Elelectronically Erasable Programmable Memory) nennt.

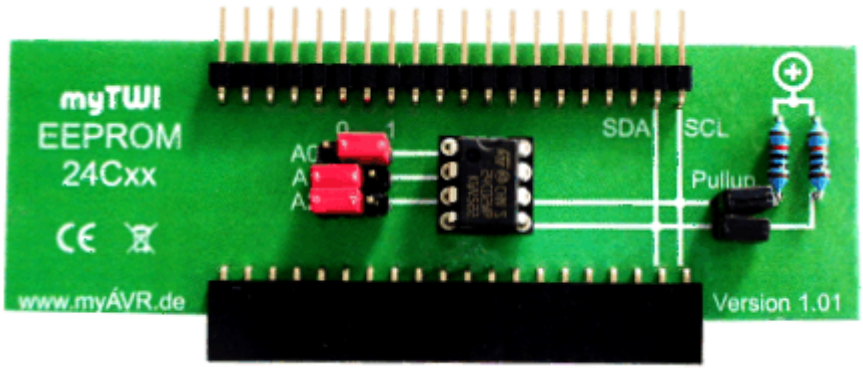

Abbildung 6
Eeprom, montiert auf einem SiSy Experimentierboard
der Firma Laser & Co Solutions GmbH

Diese Bausteine haben die günstige Eigenschaft, dass man sie mit einem Mikroprozessor beschreiben und wieder auslesen kann, und dass ihr Inhalt auch bei Stromausschaltung erhalten bleibt. Diese angenehme Eigenschaft macht es somit möglich, dass man z.B. Daten mit einem Messgerät an beliebiger Stelle erfassen kann, und dass man dann diese erfassten Daten dem Messgerät entnehmen kann, um sie zu Hause oder im Labor auf einem PC auszuwerten.

Einer dieses Chip-Typs ist der Chip AT24C02 der Firma Atmel. Er hat eine Speicherkapazität von 2k Bytes und kostet gerade mal ca. 50 Cent. Dieser Chip kann über das I2C-Protokoll an den Mikroprozessor angeschlossen werden und ist daher sicherlich die erste Wahl bei solchen Anwendungen.

Die Anbindung des Alarm-Systems an einen PC

Alarmsysteme, die auf der Basis eines eigenen Mikroprozessors arbeiten, sollten sinnvollerweise auch an einen PC anschließbar sein, zumindest dann, wenn es darum geht, Eingaben für den Mikroprozessor vorzunehmen oder Ausgaben des Mikroprozessors auszuwerten. Der simple Grund hierfür ist, dass es aus Kostengründen zu aufwändig wäre, einen Mikroprozessor mit einem Monitor bzw. einer Tastatur auszustatten. In solch einem Falle ist es besser, die dem Mikroprozessor zugedachte Aufgabe gleich dem PC zu überlassen. Hiergegen sprechen aber Überlegungen wie die Transportierbarkeit, der Platzbedarf und eben auch der Preis.

Im übrigen wird ja eine Verbindung zwischen dem PC und dem Mikroprozessor meistens nur temporär benötigt. Und hierfür reicht eine Verbindung via der seriellen Schnittstelle oder USB vollkommen aus.

Im hier praktisch vorliegenden Fall wird von der USB-Schnittstelle Gebrauch gemacht. Der Grund ist allerdings ein ganz zufälliger. Er liegt darin, dass das für die Mikroprozessor-Programmierung verwendete Entwicklungssystem namens SiSy AVR der Firma Laser & Co. auf einem USB-Anschluss basiert. Das Entwicklungssystem gestattet es, wie bereits erwähnt, die Programmentwicklung für den ATmega 168 auf dem PC vorzunehmen und diesen Programmcode dann auf den Mikroprozessor zu übertragen (mit der erwähnten Methode ISP).

Nebenbei gestattet es diese Einrichtung, mittels der Einrichtung „Terminal Emulator“ das Geschehen auf dem Mikroprozessor auf dem PC-Monitor mit zu verfolgen (durch entsprechende Programmierung). Was lag da näher als diese bestehende Einrichtung auch für die notwendige Kommunikation mit dem PC zu verwenden ? Im Klartext heißt das, dass über diese Einrichtung z.B. die Start-Daten wie Anfangs-Datum und -Uhrzeit auf den Mikroprozessor übertragen werden können. Und natürlich kann man diese Einrichtung auch dazu verwenden, z.B. die gesammelten Alarmdaten aus dem Eeprom auszulesen und auf dem PC-Monitor darzustellen.

Sobald die Entwicklung dieser Alarmanlage abgeschlossen ist und ein Einsatz in einer Umgebung stattfinden soll, in der dieses Entwicklungssystem nicht mehr zur Verfügung steht, wird diese Möglichkeit der PC-Kommunikation abgelöst werden müssen durch ein noch zu entwickelndes Programm auf dem PC. Die Wahl wird dann höchstwahrscheinlich auf einen Baustein fallen, der es gestattet, vom PC her über das I2C-Protokoll mit dem Mikroprozessor zu kommunizieren. Der Grund für diese Entscheidung ist die Verfügbarkeit entsprechender Bausteine sowie die einfache Programmierbarkeit dieser Schnittstelle.

Der intelligente Infrarotmelder, die Experimentierschaltung

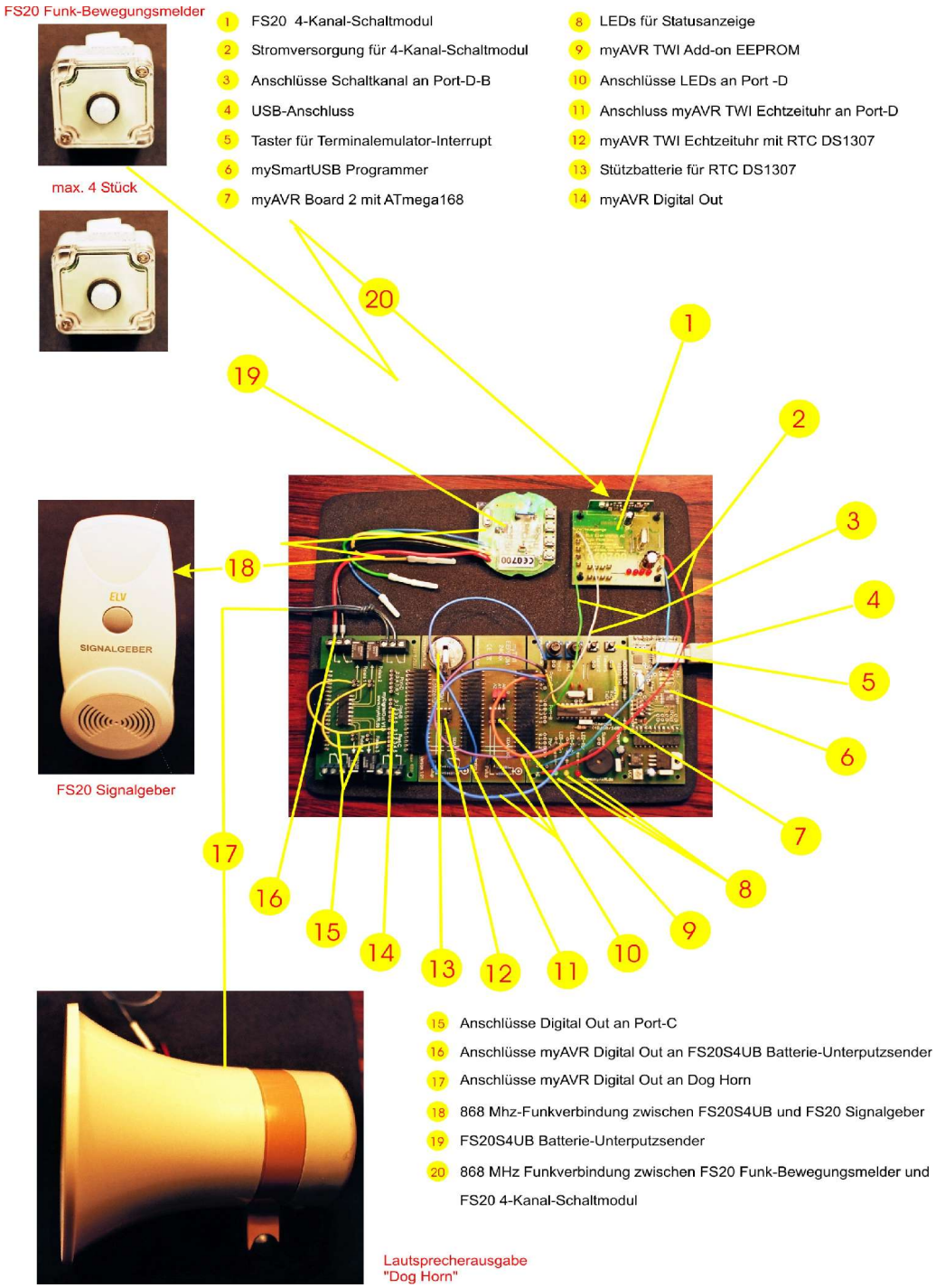

Abbildung 7
Die Experimentierschaltung