Klingel-Türöffner-Automat II

Eine Elektronik wird an die Stromkreise von Hausklingel und Türöffner angeschlossen. Wenn sie ein bestimmtes Klingelzeichen erkennt, betätigt sie per Relais den elektrischen Türöffner. Der memorierbare Code ersetzt den mechanischen Schlüssel, der Klingelknopf wird als bereits vorhandenes und unauffälliges Eingabegerät gleich doppelt genutzt. Hier nun eine aktualisierte Version mit einem kleinen Mikrocontroller!

Projekt | Hardware | Software | Nachbautipps | Installation | Täglicher Einsatz | Links | Index

Bild 1:
KTA-II mit ATtiny25-Mikrocontroller

KTA *reloaded*

Erst einmal danke ich allen Nachbauern des KTA-*classic* [1] für die wertvollen Rückmeldungen und Hinweise!
Die Idee einer Codeeingabe über kurze und lange Klingelzeichen ist irgendwie nicht totzukriegen, sodass eine neue, verbesserte Version des KTA wohl längst überfällig war. Hier mein Vorschlag:

Energieversorgung aus dem Klingelstrom
In der ungünstigen Kombination "schwacher Klingeltrafo und stromhungrige Klingel" konnte es mit der alten KTA-Schaltung schonmal Probleme mit der Energiebilanz geben, sodass man den Code eventuell sehr gehetzt vorklingeln musste, damit das Sesam-Öffne-Dich tatsächlich funktionierte. Problem erkannt! Auch der neue KTA-II gewinnt seine gesamte elektrische Energie aus dem Klingelstrom, aber sein Eigenbedarf liegt unter einem Milliampere, solange nur das Codeprüfungsprogramm abgearbeitet wird. Dieser Strombedarf ist im Vergleich zu jeder Klingel absolut vernachlässigbar. Ein Elektrolytkondensator dient auch hier wieder als Kurzzeit-Energiespeicher. Obwohl er kleiner ausgelegt werden konnte, als beim alten KTA, reicht seine Ladung nun locker aus, um den Mikrocontroller in sekundenlangen Klingelpausen weiterhin mit Strom zu versorgen. Das gesamte Timing der Pausenzeiten ist beim KTA-II nur noch "halb so wild".

Eingabe des Klingelcodes über kurze und lange Klingelzeichen

Der *Morsecode* ist als menschenfreundliche binäre Codierung unschlagbar, auch und gerade wenn man die Informationen mithilfe eines wackeligen Klingelknopfes übertragen muss. Keine Angst, niemand muss für die Bedienung des KTA eine perfekte Gebeweise beherrschen und das Tempo hält sich in Grenzen... Ein "Punkt" ist kürzer als 0,5 Sekunden, alles, was länger dauert, wird als "Strich" interpretiert. Die Pausen werden sehr großzügig gehandhabt. Das vorgeschlagene Firmware-Programm realisiert einen Klingelcode von 6 Schritten, sodass 2^6 = 64 mögliche Schlüssel verwendet werden können.

Flexibler Wechsel von Klingelcodes
Der Klingelcode wird beim KTA-II nicht mehr über ein "Mäuseklavier" vorgegeben, sondern direkt in den EEPROM-Speicher des Mikrocontrollers eingeschrieben. Auch diese Eingabe erfolgt über den Klingelknopf, ist aber selbstverständlich nur dann möglich, wenn man vorher den richtigen bisherigen Code vorgeklingelt hat. Mehr dazu weiter unten.
Die Schrittzahl des Klingelcodes ist beim vorgeschlagenen Controller-Programm auf 6 Schritte festgelegt. Mit einem längeren Schlüssel von 10 oder 12 Bits wäre der KTA-II vielleicht auch schon für Sicherheitsbereiche interessant, wo die Installation eines Zahlenschlosses oder eines Chipkarten-Lesegerätes aus irgendwelchen Gründen nicht infrage kommt. CW-Profis können sich besonders lange Klingelsequenzen auch leicht als Zusammensetzung bekannter Morsezeichen merken.

Nach oben

Hardware

Im Vergleich zur Vorgängerversion stellt sich die Elektronik des KTA-II geradezu minimalistisch dar, siehe Stromlaufplan (Bild 2). Kein Wunder auch, denn die ganze Logik spielt sich ja im Programm des Mikrocontrollers ab.

Die robust ausgelegte Gleichrichterbrücke Br1 bekommt über Klemme X1 das rohe Klingelsignal von ungefähr 8 V Wechselspannung zugeführt. Die Wechselspannung wird über die robuste Gleichrichterbrücke Br1 gleichgerichtet und lädt den Pufferkondensator C1 (entweder 1x4700µF oder 2x2200µF) auf.
Solange der Controller sein Codeprüfungs-Programm abarbeitet, liegt der Strombedarf der gesamten Schaltung unter 1 mA. Obwohl der C1 nur etwa die halbe Kapazität hat, als beim alten KTA-Konzept, kann er den lütten ATtiny25 doch trotzdem für mehrere Sekunden zuverlässig mit Energie versorgen. Der konventionelle Linearregler VR1 (7805) sorgt in dieser Zeit für saubere und stabile 5 Volt. Die Länge der Klingelpausen spielt bei dieser Schaltung also nur noch eine untergeordnete Rolle.

Die Programmierleitungen MISO/MOSI/SCK des ATtiny25 sind auf eine 10-polige Pfostensteckerleiste X3 herausgeführt, sodass sich der Chip direkt auf der Platine "flashen" lässt. Diese Steckerleiste wird auch im späteren Einsatz nicht ganz überflüssig: Wenn wir nämlich die rechten zwei Pins 9 (PB1) und 10 (GND) über eine Steckbrücke kurzschließen, und der Controller das nächste Mal Strom bekommt, dann geht er in eine Art "Reset"-Modus, in dem der EEPROM-Inhalt mit einer Default-Klingelsequenz überschrieben wird.

Die Portleitung PB3 (Pin 2 von IC1) ist als Eingang für die Klingelzeichen konfiguriert. Dazu wird der Klingelwechselstrom vor dem Gleichrichter hochohmig über R1 (10 kOhm) abgegriffen und über die Z-Diode ZD1 (4,9 V) auf TTL-Pegel begrenzt. Parallel zum Porteingang liegt noch der Folienkondensator C5 (100 nF), der höherfrequente Störimpulse auf dem Signal abblocken soll, sowie ein Pull-Down-Widerstand R2 (22 kOhm), welcher den Porteingang in den Klingelpausen auf Low-Pegel herunterzieht. Ohne Klingelwechselspannung, oder bei offener Leitung "sieht" der Controller an diesem Eingang einen eindeutigen Low-Pegel. Wenn jedoch Klingelstrom fließt, dann erscheint an Pin 2 ein pulsierendes, rechteckähnliches 50-Hz-Signal. (Diese Pulse lassen sich softwaremäßig sauber auswerten und nebenbei kann der Controller durch Abzählen der Perioden

recht genau feststellen, wie lange das Klingelzeichen dauerte, und zwar unabhängig von der genauen Programmlaufzeit.)

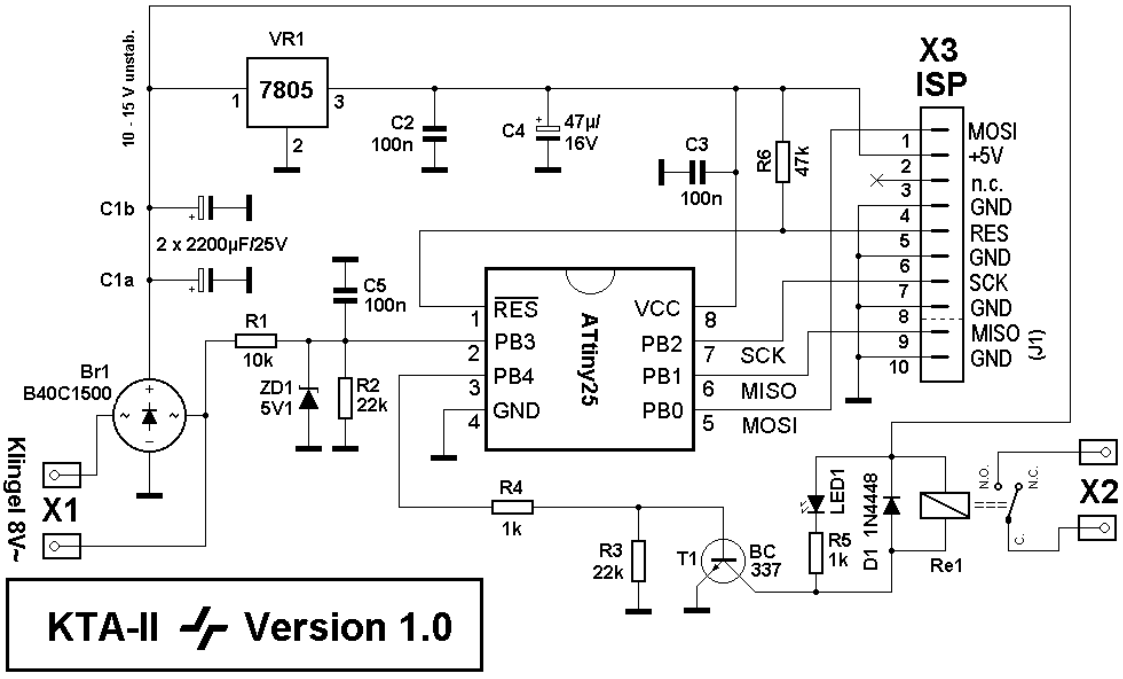
Seinen Ausgang PB4 zieht der Controller auf "High", wenn er den richtigen Code erkannt hat. Damit wird der Schalttransistor T1 (BC337) sicher durchgesteuert, sodass nun die volle unstabilisierte Elkospannung (ca. 12-16V) an der Erregerspule des Relais Re1 anliegt. Das Relais zieht an und der Türöffner-Kontakt wird vorübergehend geschlossen.

Hinweis: Der Summer bleibt auch nach dem Loslassen des Klingelknopfes weiter eingeschaltet, wird aber nach einem Timeout von 5 Sekunden auf jeden Fall wieder ausgeschaltet. Sollte der Klingelknopf zu diesem Zeitpunkt weiterhin gedrückt sein, dann geht der KTA-II in einen "Lernmodus", in dem ein neuer Code eingegeben werden kann. Das Protokoll wird unter "Einsatz" genauer erklärt.

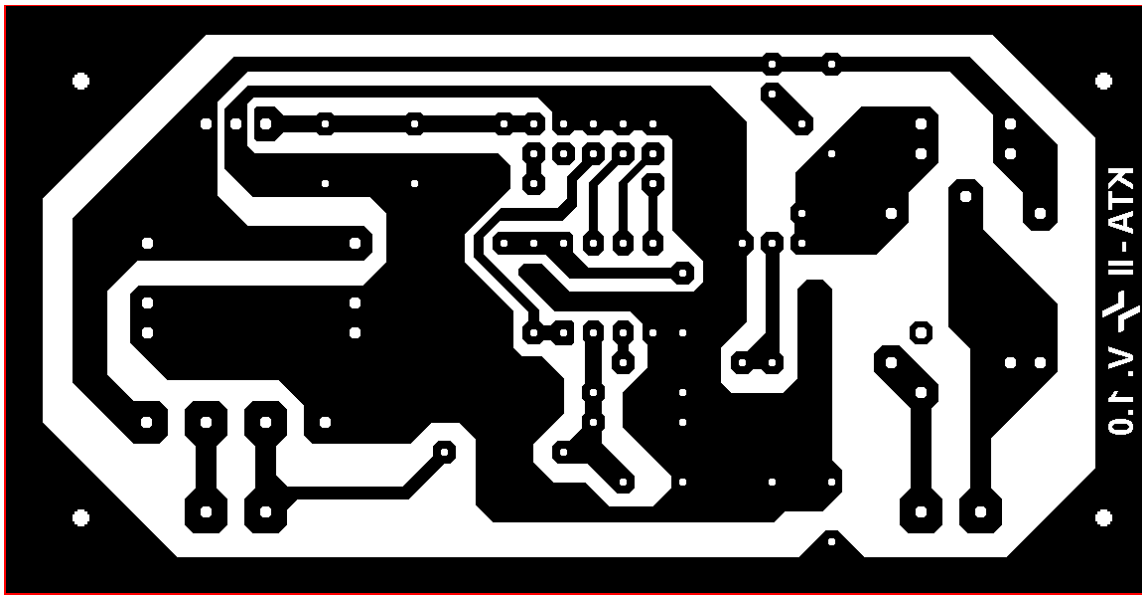
Nach oben

Schaltunterlagen Klingel-Türöffner-Automat II (KTA-II) Version 1.0

1. Schaltplan KTA-II V1.0

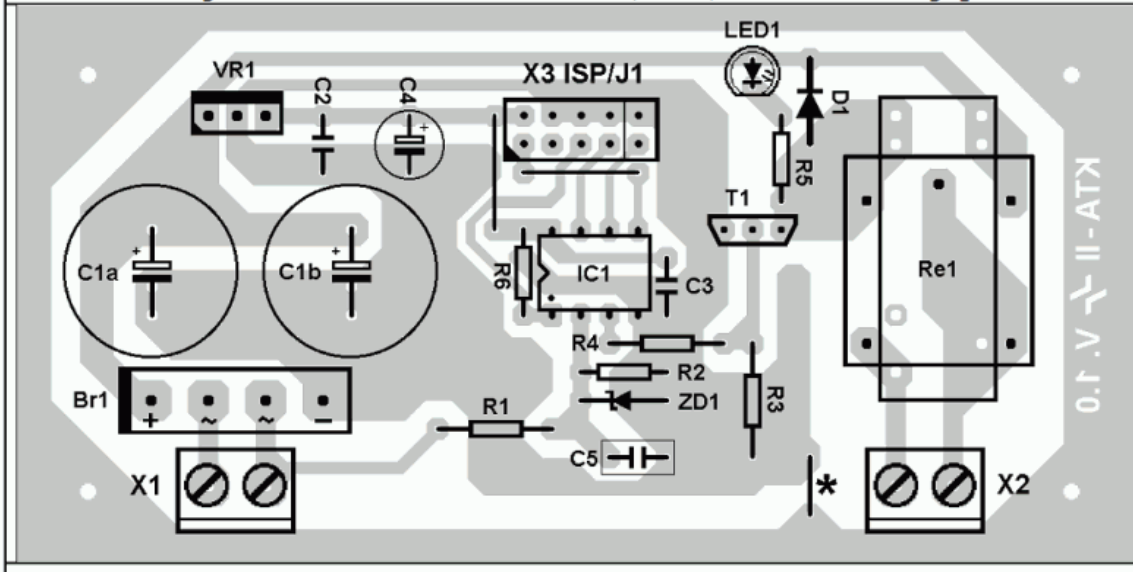


2. Layout KTA-II V1.0, einseitig, 300dpi, 50 x 100 mm (Anklicken und Herausspeichern möglich)



3. Bestückungsplan und Stückliste KTA-II V1.0 - Arbeitsblatt

Klingel-Türöffner-Automat II (V1.0) - Bestückungsplan



Klingel-Türöffner-Automat II (V1.0) - Stückliste

Widerstände:

R1	10k
R2,R3	22k
R4,R5	1k
R6	47k

Kondensatoren:

C1a/C1b	Elko 2 x 2200µF/25V (mind.)
od.	Elko 1 x 4700µF/25V (mind.)
C2, C3	100nF Kerko
C4	Elko 47µF/16V
C5	100nF Folie (MKS)

Halbleiter:

Br1	Gleichrichterbrücke Belastbarkeit 1A (Belegung: + ~ ~ -) z.B. Typ B40C1500
VR1	7805 Festspannungsregler
D1	1N4148/1N4448
ZD1	Z-Diode min. 0,25W, 5,1V od. 4,9V
T1	BC337-25 (NPN-Schalttransistor)
LED1	LED "low current"
IC1	AVR ATtiny25 + Firmware "KTA-II"

Sonstige Teile:

IC-Fassung DIL8

2 x 5 pol. Pfostenstecker f. X3 (ISP)

1 x Steckbrücke RM 2,5 mm,
oder Schalter zw. Pin X3-9 / X3-10

2 x Klemme für X1/X2

Re1 Relais 12V, 1 x Umschalter,
Spulenwiderstand mind. 200 Ohm
z.B. "HJ3FF 12VDC SZ" oder
"FIN 41.31.9 12V"(Reichelt)

Platine nach Layout 50 x 100 mm

Gehäuse, Innenabmaße größer als 100 x
50 mm, z.B.: TEKO WALL 2

(*) Bei Montage in Metallgehäuse:
Drahtbrücke oder 100-nF-Kondensator
gegen Masse

Nach oben

Nachbautipps

Wer sich die Platine selbst anfertigen kann, findet in den Schaltunterlagen ein erprobtes Platinenlayout. Es handelt sich um eine einfache Pixelgrafik im Maßstab 300dpi. Das bedeutet, der Ausdruck auf Papier oder Folie wird mit einem vernünftigen Grafikprogramm und den richtigen Druckertreibern automatisch maßhaltig sein.

Bevor wieder freche Nachfragen zu imaginären "Quelldateien" kommen: Was ich hier veröffentliche, sind alle fertig ausgearbeiteten Informationen, die ich anbieten kann. Glaubt wirklich jemand, dass ein Autorouter solche Layouts zustande gebracht hätte? Nein, ich erstelle meine Schaltpläne und Platinenlayouts mithilfe von allgemein verfügbaren Grafikprogrammen und einer selbstgeschaffenen Sammlung aus grafischen Makros. Auf die unästhetischen Ergebnisse sogenannter Layout-Software, auf eine chronische Versionitis und auf die Erpressung durch proprietäre Formate kann ich verzichten. Nix Adler, nix Zielscheibe, hier ist noch alles hausgemacht!

Den Bestückungsplan habe ich wieder zusammen mit der Stückliste in einer Grafik zusammengefasst. Schnell ausgedruckt, schon haben wir ein praktisches Arbeitsblatt für den Nachbau der Schaltung.

Die Platine sieht für den Pufferkondensator C1 zwei Aufbauvarianten vor: Hier können zum Beispiel zwei separate Elkos von jeweils mindestens 2200µF (radial, Rastermaß 5 mm oder 7,5 mm) eingelötet werden. Falls die Bauhöhe im Gehäuse sehr begrenzt ist, wäre auch ein 4700µF-Elko mit angewinkelten radialen Anschlüssen liegend einzusetzen, wodurch sich ein noch flacheres Profil ergibt. Kleinrelais mit geringer Bauhöhe gibt es ja ebenfalls. Im Layout habe ich zwei der am häufigsten vorkommenden Anschlussvarianten für solche Relais berücksichtigt. Vielleicht hält ja schon die gutsortierte Bastelkiste ein passendes Exemplar bereit! Die technischen Daten sind relativ unkritisch: Ansprechspannung 12V, möglichst hoher Spulenwiderstand, für 230V-Anwendungen geeignet, hochwertiger Schließerkontakt oder 1 x Umschalter, Strombelastbarkeit mindestens 2A.

Bei Verwendung eines Metallgehäuses oder einer metallischen Montageplatte sollte die Schaltung einen eindeutigen Massebezug erhalten, das verbessert die Störsicherheit noch weiter. An der mit einem Sternchen gekennzeichneten Stelle können wir dazu eine Drahtbrücke (oder für rein HF-mäßige Erdung einen 100-nF-Keramikkondensator) einsetzen. Der elektrische Kontakt mit dem Chassis kommt dann über die Befestigungsschrauben zustande.

Platinen oder Bausätze stelle ich auf Anfrage gern zum Selbstkostenpreis zur Verfügung.

Nach oben

Programmierung und Fusebits

Das Programm wurde unter BASCOM-AVR geschrieben und compiliert. Es ist recht einfach strukturiert, man könnte durchaus sagen, *primitiv*. Wer hätte gedacht, dass man eine so anspruchsvolle Aufgabe auch heute noch ganz ohne C-Libraries, ohne Internetanbindung und Touchscreen mit lächerlichen 1 kB Programmspeicher lösen kann?! (Ja, richtig, *Kilobyte*, nicht *Megabyte*!). Und trotzdem erfüllt dieses simple Programm seinen Zweck. Für weitere Spielereien ist aber noch jede Menge Speicherplatz vorhanden. Das Programm lässt sich zum Beispiel sehr einfach durch Ändern einer einzigen Konstante auf kürzere oder längere Klingelsequenzen umschreiben.

Obwohl ein paar Byte vom EEPROM-Speicher genutzt werden, brauchen wir keine separate ".EEP"-Datei, weil für die Initialisierung des EEPROM eine eigene Programmroutine vorgesehen ist. Dies können wir bei der ersten Inbetriebnahme tun, indem wir an X3 die Pins 9+10 kurzschließen. Wenn hier eine Steckbrücke sitzt und die Schaltung Betriebsstrom bekommt, dann wird ein Unterprogramm gestartet, das den EEPROM-Speicher mit dem Code für die

Standardsequenz "6 mal kurz" überschreibt. Auf diese Weise bekommen wir immer wieder einen benutzbaren KTA-II, auch wenn wir den Klingelcode verbummelt haben oder aus irgendeinem Grund undefinierte Werte im EEPROM stehen. (Das gezielte Einstellen eines anderen Klingelcodes erfolgt dann "an der Haustür", nachdem wir den bisherigen Code einmal vorgeklingelt haben, mehr dazu weiter unten.)

Die ursprüngliche Firmware (für 6 Schritte) und Begleitmaterialien sind nach wie vor auf der FA-Website zu finden.

Außerdem habe ich jetzt mal selbst ein Päckchen geschnürt, das alle für den Nachbau erforderlichen Materialien, Hinweise zu den Fusebits und zusätzliche Firmware-Versionen enthält. (Version mit **4 Schritten** und/oder höherem Prozessortakt von **1 MHz**)
Download hier.

Um das KTA-II-Programm in den Chip zu bekommen, benötigen wir entweder ein Experimentierboard für ATtiny25/45/85, oder ein serielles Programmiergerät mit dem 10-poligen ISP-Stecker nach ATMEL-Spezifikationen. Über die Pfostensteckerleiste X3 lässt sich der Chip auch in dieser Schaltung "in circuit" programmieren. Für die Dauer des Programmiervorgangs muss die KTA-II-Platine allerdings aus einer sauberen Spannungsquelle versorgt werden. Hierfür eignet sich ein Wechselspannungsnetzteil 8...12 V AC, welches an die Klemme X1 dauerhaft angeschlossen wird.

"Vorsicht, Fusebits!"

16-kHz-Variante
DAS IST JETZT GAAAANZ WICHTIG: *Sämtliche Zeitschleifen in dem vorcompilierten Programmcode sind auf einen Prozessortakt von nur 16 kHz ausgelegt.* Diese recht niedrige Taktfrequenz reicht für das Codeprüfungs-Programm locker aus, und sie ermöglicht dem ATtiny25 einen besonders stromsparenden und störsicheren Betrieb. Wer also das vorcompilierte Programm direkt verwenden will, muss (am besten erst NACH dem Flashen des Programms) die Fusebits folgendermaßen ändern: **Ext: $FF / High: $DF / Low: $64**
Hiermit wird der interne Watchdog-Oszillator von nominell 128 kHz anstelle des internen RC-Oszillators als Taktquelle eingeschaltet und aufgrund der weiteren Taktteilung durch 8 landen wir bei abgrundtiefen 16 kHz.
Im fabrikneuen Zustand haben die AVRs normalerweise ihren internen RC-Oszillator im MHz-Bereich aktiviert. Die Standardeinstellungen der meisten Brennprogramme gehen ebenfalls von einem AVR-Takt im MHz-Bereich aus, sodass das Flashen eines Programms meistens auf Anhieb funktioniert. Wenn wir nun aber die Fusebits wie oben beschrieben ändern, ist das Timing möglicherweise zu extrem für manchen Brenner. Manche Programmer-Software wirft dann bereits das Handtuch und behauptet, der Chip sei "tot" oder der Programmieradapter "defekt".
Ich habe für dieses Projekt ursprünglich nur TwinAVR [4] eingesetzt, weil es sich sehr flexibel auch an extrem niedrige Chip-Taktfrequenzen anpassen lässt. Einfach nach dem Ändern der Fusebits im Startfenster "AVR Clock" auf "16 kHz" ändern - jetzt nochmal auf "Config" klicken, und da ist es auch schon, das Aha-Erlebnis!

1-MHz-Variante
Für den Betrieb mit 1 MHz kann der ATtiny25 im Grunde mit den "Werkseinstellungen" belassen werden. Wenn wir den Brownout-Detektor zusätzlich einschalten, was ein stabileres Verhalten bei langsam abfallender Betriebsspannung verspricht (Klingelcode verklingelt), dann ergeben sich folgende Werte für die Fuses: **Ext: $FF / High: $DC / Low: $62**
Die kürzlich von mir getestete 1-MHz-Variante scheint bezüglich der Störsicherheit keine erkennbaren Nachteile gegenüber der 16-kHz-Variante zu haben. Mit dieser Version sollte nun wirklich jeder ISP-Adapter zurechtkommen.

Nach oben

Installation

Der erste Funktionstest erfolgt am besten noch am Arbeitstisch, nach dem Flashen des Programms und der Fusebits.
Dafür könnten wir einfach die Primärspannung für ein Wechselspannungsnetzteil im Takt des Klingelsignals ein- und ausschalten. Eleganter geht es natürlich mit einem Taster in der Versorgungsleitung zur Klemme X1.
Während dieser Tests muss der ISP-Stecker vom Programmiergerät unbedingt abgezogen sein!

Die Installation im realen Klingel- und Türöffnerstromkreis ist nicht sonderlich kompliziert, die Eingriffe sind im Bedarfsfall auch leicht wieder rückgängig zu machen. Der KTA-II wird über Klemme X1 mit einem zweiadrigen Kabel parallel zur 8-V-Klingel angeschlossen. Ausgehend von seiner Klemme X2 wird ein weiteres zweiadriges Kabel zum Türöffner-Kontakt geführt. Falls sich der Taster für den Türöffner-"Summer" im Gehäuse einer Türsprechanlage versteckt, muss möglicherweise dieses Gehäuse geöffnet werden, um an die Leitungen heranzukommen. Das andere Ende des Kabels wird mit der Klemme X2 des KTA verbunden. Damit ist die elektrische Installation bereits abgeschlossen.

Zusätzliche Entstörmaßnahmen für den Türöffner-Kontakt sind normalerweise nicht erforderlich, da die Abschalt-Spannungsspitze ja im Normalbetrieb erst dann auftritt, wenn der KTA-II seinen Job bereits erledigt hat.

Was den KTA-II schon eher aus dem Konzept bringen könnte, wären die massiven Funkenstörungen einer altertümlichen Klingel mit Unterbrecherkontakt. Wer diese aus nostalgischen Gründen beibehalten will oder aus vermietertechnischen Gründen beibehalten muss, sollte wenigstens eine Entstörung versuchen. Ein spannungsfester Folienkondensator (0,33µF bis 1µF), parallel zur und möglichst nah an der Klingel angeschlossen, kann hier wahre Wunder bewirken. (In einer solchen Konstellation wurde der KTA-II auch schon erfolgreich eingesetzt!)

Nach oben

Täglicher Einsatz

Türöffner per Klingelcode betätigen:

- Das 1. Klingeln sollte mindestens eine Sekunde dauern, damit sich der Energiespeicher im KTA-II sicher auflädt.
  Dieses 1. Klingeln zählt nicht zum Klingelcode.
- Die folgenden 6 Klingelzeichen sind der eigentliche Klingelcode. Ein "Punkt" dauert nicht länger als 1/2 Sekunde, alles Längere zählt als "Strich".
- Durch das 8. und abschließende Klingelzeichen wird der Türöffner aktiviert, sofern der zuvor geklingelte Code korrekt war.
- Das Relais wird gehalten, bis die Ladung im Speicherkondensator erschöpft ist. (Darüber hinaus begrenzt eine Zeitschleife im Programm die maximale Einschaltzeit auf 5 Sekunden - man hat schon von Türöffner-Magneten gehört, die einen Dauerbetrieb nicht abkönnen.)

Lernmodus:

- Zuerst muss der bisherige Klingelcode mit Start- und Stopp-Klingeln eingegeben werden, wie oben.
- Durch das abschließende 8. Klingelzeichen wird der Türöffner aktiviert. Knopf aber diesmal nicht loslassen, sondern weiter gedrückt halten!
- Nach 5 Sekunden schaltet sich der Türöffner automatisch wieder ab. Wird der Klingelknopf auch jetzt noch weiterhin gedrückt gehalten, dann geht das Programm in den Lernmodus.
- Ein neuer Code kann nun direkt "vorgeklingelt" werden (also 6 kurze oder lange

Knopfdrücke).
- Auch dieser Code muss über ein abschließendes Klingeln bestätigt werden.
- Bleibt der Klingelknopf während des abschließenden Klingelns weiterhin gedrückt, dann wiederholt der KTA-II den neu programmierten Klingelcode über das Türöffner-Relais. (Die Wiederholung ist optional, sie muss nicht abgewartet werden.)

Rücksetzen des Klingelcodes:

- Wenn zwischen den Pins 9 und 10 von X3 ein "Jumper" sitzt, so erkennt das Programm das Vorhandensein dieser Steckbrücke beim nächsten Start und geht sofort in ein spezielles Unterprogramm.
- Klingelknopf weiter gedrückt halten!
- Der bisherige EEPROM-Inhalt wird mit der Standardsequenz "6 x kurz" überschrieben.
- Klingelknopf weiter gedrückt halten. Der KTA-II bestätigt den neuen Klingelcode durch 6-maliges kurzes Summen bzw. Aufleuchten der Kontroll-LED.
- Klingelknopf loslassen, Steckbrücke entfernen!
- Den Standardklingelcode "6x kurz" können wir entweder beibehalten oder im Rahmen des normalen Lernmodus' ändern.

Nach oben

Links

[1] Thomas, J.: "Unverlierbarer Schlüssel: Klingel-Türöffner-Automatik", FUNKAMATEUR 51, 5/02, S.471ff

[2] Thomas, J.: "Morsezeichen öffnen Türen: Klingel-Türöffner-Automatik II", FUNKAMATEUR 58, 4/09, S.394ff,
Download der Layouts und Software auf: www.funkamateur.de

[3] Datenblatt zum ATtiny25, Atmel Corporation: http://www.atmel.com/dyn/resources/prod_documents/doc2586.pdf

[4] Beschaffung der Bauteile: www.reichelt.de

[5] Roland Walter's unübertroffenes AVR-Programmertool "TwinAVR": www.rowalt.de

[6] Download KTA-II-Paket mit verschiedenen Programmvarianten: KTA2.ZIP

Nach oben

Index

Revisionen: 05/2009 09/2009 01/2011 03/2012