

# **Pack d'apprentissage professionnel sur les microcontrôleurs**

**Code : 192286**



Les appareils électriques et électroniques usagés (DEEE) doivent être traités individuellement et conformément aux lois en vigueur en matière de traitement, de récupération et de recyclage des appareils.

Suite à l'application de cette réglementation dans les Etats membres, les utilisateurs résidant au sein de l'Union européenne peuvent désormais ramener gratuitement leurs appareils électriques et électroniques usagés dans les centres de collecte prévus à cet effet.

En France, votre détaillant reprendra également gratuitement votre ancien produit si vous envisagez d'acheter un produit neuf similaire.

Si votre appareil électrique ou électronique usagé comporte des piles ou des accumulateurs, veuillez les retirer de l'appareil et les déposer dans un centre de collecte.

## **Note de l'éditeur**

Cette notice est une publication de la société Conrad, 59800 Lille/France.  
Tous droits réservés, y compris la traduction. Toute reproduction, quel que soit le type (p.ex. photocopies, microfilms ou saisie dans des traitements de texte électronique) est soumise à une autorisation préalable écrite de l'éditeur.

Reproduction, même partielle, interdite.

Cette notice est conforme à l'état du produit au moment de l'impression.

**Données techniques et conditionnement soumis à modifications sans avis préalable.**

© Copyright 2001 par Conrad. Imprimé en CEE. XXX/03-13/JV

Cette notice fait partie du produit. Elle contient des informations importantes concernant son utilisation. Tenez-en compte, même si vous transmettez le produit à un tiers.

**Conservez cette notice pour tout report ultérieur !**

## Contrôle programmable par touches

Les microcontrôleurs sont partout : dans les appareils électroménagers, les appareils électroniques grand public, les véhicules, les appareils de mesure et même les véhicules spatiaux sans équipage. Ils effectuent partout ce que leur programme leur ordonne. C'est très intéressant de créer, ne serait-ce qu'une fois, un programme de commande simple.

La première étape consiste toujours à choisir un microcontrôleur ou un processeur le mieux adapté à la tâche désirée. Vous avez le choix entre d'innombrables types de différentes entreprises. Et même la langue de programmation peut être sélectionnée. Assembler et C sont proposés la plupart du temps, mais Basic et un autre langage le sont également dans de nombreux cas. Généralement, un logiciel coûteux ainsi qu'un programmeur sont nécessaires pour la programmation. Le temps d'initiation n'est pas à négliger, tout comme l'aspect financier.

Le microcontrôleur utilisé ici est très différent. Vous n'avez pas besoin de plus de deux boutons-poussoirs pour la programmation. Le contrôle programmable par touches (TPS) a un nombre d'ordres limité, qui s'apprennent rapidement et se programment à l'aide des touches du contrôleur. Une modification du programme est possible à tout moment, sans outil particulier.

Le système est par ailleurs parfaitement adapté pour les applications compacte dans le domaine de la mesure, de la commande et de la régulation. De nombreuses opérations peuvent complètement être déclenchées avec ce système. A cela s'ajoute le fait que le microcontrôleur peut être intégré à vos propres circuits après une programmation réussie. Des connaissances en électronique sont donc requises.

Le système est aussi parfaitement comme base pour des formations et pour vos premiers pas dans la programmation d'un microcontrôleur. Les réussites apparaîtront plus rapidement que dans les autres systèmes. Les structures sont cependant similaires aux autres langage de programmation, de sorte que la transition sera plus simple par la suite.

## 1 Introduction

Le principe du contrôleur TPS est simple. Il dispose de quatre entrée numériques E1 à E4 et de quatre sorties numériques A1 à A4. Il y a en plus deux entrées analogiques AD1 et AD2 ainsi qu'une entrée PWM quasi-analogique. L'entrée Reset permet de réinitialiser un programme lorsqu'un bouton Reset est branché. Le contrôleur est alimenté par trois piles LR6 (AA) d'env. 4,5 V et peut fonctionner sur une plage de 2,2 V à 5,5 V.

### Caractéristiques techniques :

Microcontrôleur : HT46F47

Fréquence : 2 MHz

EEPROM interne : 128 octets

Alimentation VCC : 2,2 V à 5,5 V

Consommation absorbée : 1 mA à 4,5 V

4 ports de sortie : charge admissible jusqu'à 10 mA

1 port PMW : charge admissible jusqu'à 10 mA

4 ports d'entrée : 1 état de repos

2 entrées analogiques : 0V ... VCC

2 entrées de touches : 1 état de repos

73	5	4	Port = A
74	2	3	délai 10 ms
75	C	E	S1 = 1?
76	3	2	saut -2
77	C	F	S2 = 1?
78	E	0	Return
79	C	C	S1 = 0?
7A	3	3	saut -3
7B	7	1	A = A + 1
7C	2	3	délai 10 ms
7D	C	C	S1 = 1?
7E	3	1	saut - 1
7F	3	C	saut -12

CC 31 40 54 23 CE 32 CF E0 CC 33 71 23 CC 31 3C

Page 7 : sous-programme saisie par touches

### Tableau des commandes

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
	Port =	Wait	Jump -	A =	... = A	A = ...	A = ...	Page	Jump	C*	D*	Skip if ...	Call	Ret
0	0	1 ms	0	0				0	0	0	0		0	
1	1	2 ms	1	1	B = A	A = B	A = A+1	1	1	1	1	A>B	1	
2	2	5 ms	2	2	C = A	A = C	A = A-1	2	2	2	2	A<B	2	
3	3	10 ms	3	3	D = A	A = D	A = A+B	3	3	3	3	A = B	3	
4	4	20 ms	4	4	Dout.0 = A.0	A = Din.0	A = A-B	4	4	4	4	Din.0 = 1	4	
5	5	50 ms	5	5	Dout.0 = A.0	A = Din.0	A = A*B	5	5	5	5	Din.1 = 1	5	
6	6	100 ms	6	6	Dout.1 = A.0	A = Din.1	A = A/B	6	6	6	6	Din.2 = 1	6	
7	7	200 ms	7	7	Dout.2 = A.0	A = Din.2	A = A And B	7	7	7	7	Din.3 = 1	7	
8	8	500 ms	8	8	Dout.3 = A.0	A = Din.3	A = A Or B		8	8	8	Din.0 = 0	8	
9	9	1 s	9	9	PWM = A	A = AD1	A = A Xor B		9	9	9	Din.1 = 0	9	
A	10	2 s	10	10		A = AD2	A = Not A		A	A	A	Din.2 = 0	A	
B	11	5 s	11	11					B	B	B	Din.3 = 0	B	
C	12	10 s	12	12					C	C	C	S1 = 0	C	
D	13	20 s	13	13					D	D	D	S2 = 0	D	
E	14	30 s	14	14					E	E	E	S1 = 1	E	
F	15	60 s	15	15					F	F	F	S2 = 1	F	

Adresse	Commande	Données	Remarque
50	4	E	A = 15 «son long»
51	9	3	Adr 03
52	4	5	A = 5 «son court»
53	5	3	D = A «son variable»
54	1	9	A4 = 1
55	1	1	A4 = 0
56	2	1	2 ms
57	1	9	A4 = 1
58	1	1	A4 = 0
59	2	1	2 ms
5A	1	9	A4 = 1
5B	1	1	A4 = 0
5C	2	0	1 ms
5D	B	4	D x 04
5E	1	0	Dout 0
5F	E	0	Return

4F 93 45 53 19 11 21 19 11 21 19 11 20 B4 10 E0

**Page 5** : sous-programme sortie sonore

Adresse	Commande	Données	Remarque
60	2	3	délai 10 ms «délai S1»
61	C	E	S1 = 1?
62	3	2	saut -2
63	2	3	délai 10 ms
64	C	C	S1 = 0?
65	3	1	saut -1
66	E	0	Return
67	F	F	-
68	2	3	délai 10 ms»délai S2»
69	C	F	S2 = 1?
6A	3	2	saut -2
6B	2	3	délai 10 ms
6C	C	D	S2 = 0?
6D	3	1	saut -1
6E	E	0	Return
6F	F	F	-

23 CE 32 23 CC 31 E0 FF 23 CF 32 23 CD 31 E0 FF

**Page 6** : sous-programme délai S1 et délai S2

Adresse	Commande	Données	Remarque
70	C	C	S1 = 0? «saisie par touche»
71	3	1	saut -1
72	4	0	A = 0

## Eléments du pack d'apprentissage :

Carte enfichable

Compartiment à piles 3 x LR6 (AA)

Fil

HT46F47 avec logiciel TPS

3 boutons-poussoirs

4 LED 5 mm, rouge

1 LED 5 mm, verte

1 LDR

3 condensateurs céramiques 100 nF

1 condensateur électrolytique 47 µF

5 résistances 2,2 kΩ

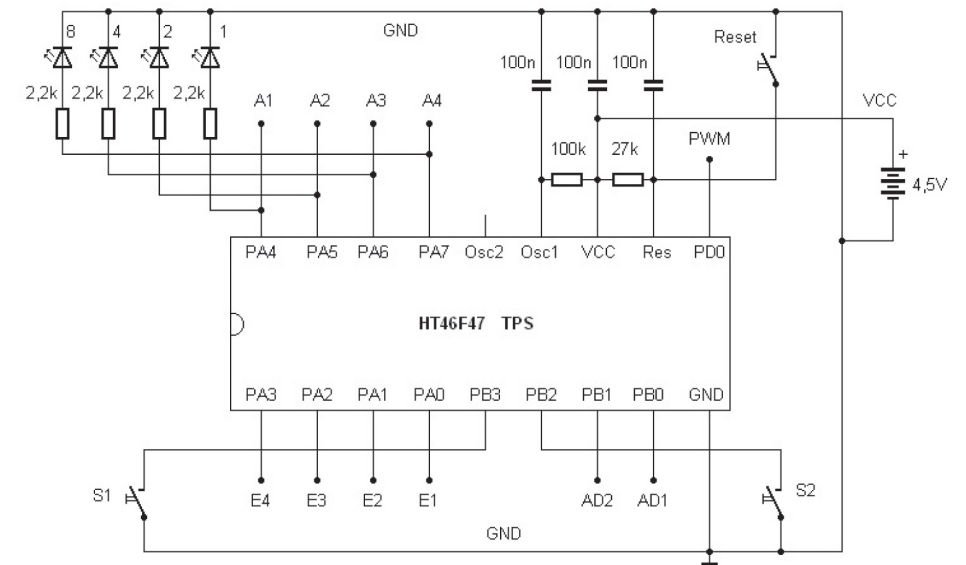
1 résistance 10 kΩ

1 résistance 27 kΩ

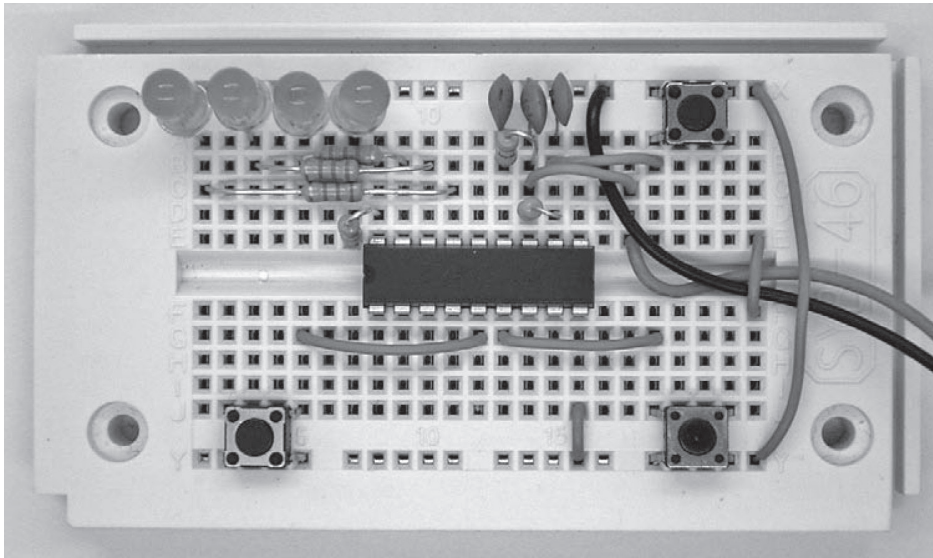
2 résistances 100 kΩ

Les deux boutons S1 et S2 sont nécessaires pour la programmation ainsi qu'un affichage LED simple composé de quatre LED sur les sorties A1 à A4. Il y a au total 14 ordres simples avec les données et les sous-commandes associées.

Les commandes et les données sont chacune codées en nombre binaire à 4 bits sur la plage de 0000 à 1111 (décimal : 0 à 15). Elles sont ensuite directement visibles au niveau des LED d'affichage. Chaque chiffre est programmé par une pression sur le bouton S1 lors de la programmation. S2 change constamment entre les ordres et les données et augmente l'adresse de la ligne de commande. L'ensemble de la structure du programme est si simple, qu'il est tout à fait possible de la mémoriser après quelques exercices.



**Img. 1** : circuit de base du système



Img. 2 : structure standard avec boutons-poussoirs

Certains programmes de base sont déjà disponibles par défaut (à la livraison) dans le contrôleur TPS. Ils peuvent donc être démarrés directement. C'est pour cette raison que vous devez faire fonctionner le contrôleur par étapes. Familiarisez-vous avec les caractéristiques matérielles, vous pourrez commencer seulement ensuite avec vos propres programmes.

Lors des premiers tests, des petits programmes se trouvant déjà dans le contrôleur sont démarrés. La liste des programmes jointés permet d'apercevoir un premier aperçu des possibilités offertes. Chacun sera décrit brièvement. Retrouvez dans le chapitre suivant la description exacte de chaque commande.

Lors de votre premier essai, montez seulement la configuration de base avec le contrôleur et les éléments supplémentaires nécessaires sur la platine enfichable. Sont absolument requis :

- Raccordement de la tension d'alimentation à GND (moins) et VCC (plus)
- Un condensateur de blocage 100 nF entre VCC et GND
- Résistance reset vers VCC et condensateur Reset vers GND
- Résistance oscillateur de 100 kΩ vers VCC et condensateur vers GND.

Le microcontrôleur HT46F47 fonctionne avec son oscillateur RC interne. La résistance de l'entrée Osc1 détermine la fréquence d'horloge. Une fréquence d'env. 2 MHz est réglée pour 100 kΩ. Le cas échéant, il peut fonctionner avec une vitesse plus petite ou plus grande. Le condensateur branché sert uniquement au blocage et n'a pas d'influence sur la fréquence d'horloge. La connexion Osc2 reste libre. Si besoin, vous pouvez toutefois brancher ici une résistance supplémentaire à VCC et coupler les impulsions à un quart de la fréquence d'horloge.

2E	3	4	Saut -4
2F	F	F	-

11 28 18 28 34 71 54 59 26 34 69 54 59 26 34 FF

Page 2 : programmes d'exemple : clignotant alterné, incrémentation, AD/PWM

Adresse	Commande	Données	Remarque
30	5	4	Port = A «aléatoire»
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Saut -3
34	2	2	délai 5 ms «chronomètre S1»
35	C	C	S1 = 0?
36	3	2	saut - 2
37	4	0	A = 0
38	2	2	délai 5 ms
39	7	1	A = A + 1
3A	5	4	Port = A
2B	C	E	S1 = 1?
3C	3	4	Saut -4
3D	3	9	Saut -9
3E	F	F	-
3F	F	F	-

54 CE 71 33 22 CC 32 40 22 71 54 CE 34 39 FF FF

Page 3 : programmes d'exemple : aléatoire, chronomètre S1

Adresse	Commande	Données	Remarque
40	8	6	page 6 «marche/arrêt chronomètre»
41	D	0	appel «délai S1»
42	4	0	A = 0
43	7	1	A = A + 1
44	5	4	Port = A
45	2	3	délai 10 ms
46	C	D	S2 = 0?
47	3	4	saut -4
48	D	8	appel «délai S2»
49	4	0	A = 0
4A	5	4	Port = A
4B	3	B	saut -11
4C	F	F	-
4D	F	F	-
4E	F	F	-
4F	F	F	-

86 D0 40 71 54 23 CD 34 D8 40 54 3B FF FF FF FF

Page 4 : programme d'exemple chronomètre marche/arrêt

0A	C	3	A = B?
0B	9	E	saut 0E
0C	8	2	page 2
0D	9	A	saut 2A, »AD/PWM«
0E	4	B	A = 1011
0F	8	1	page 1

64 51 4E 80 C3 98 82 95 4D 80 C3 9E 82 9A 4B 81

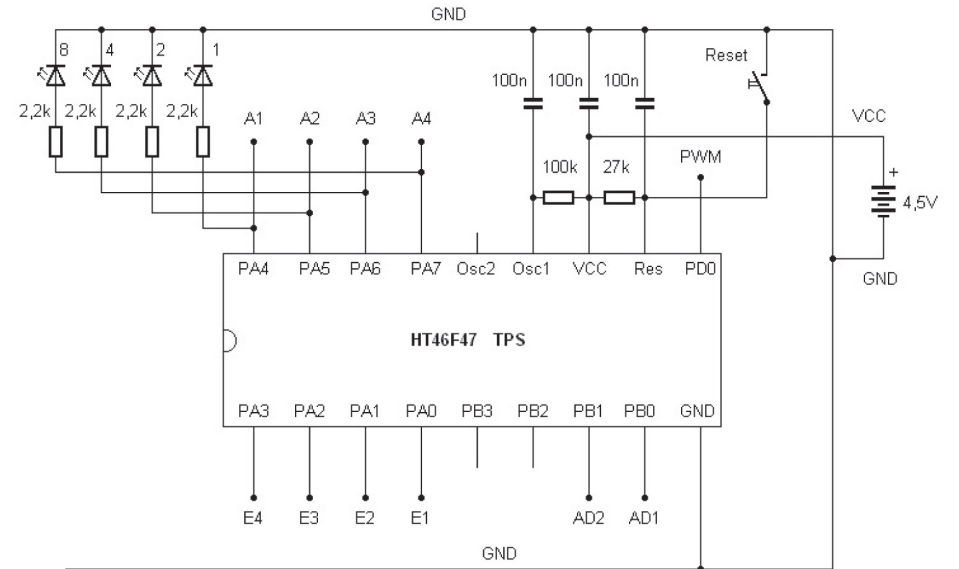
**Page 0** : sélection et démarrage du programme d'exemple

Adresse	Commande	Données	Remarque
10	C	3	A = B?
11	9	4	saut 14
12	8	3	page 3
13	9	0	saut 30, «aléatoire»
14	4	7	A = 0111
15	8	1	page 1
16	C	3	A = B?
17	9	A	saut 1A
18	8	3	page 3
19	9	4	saut 34, «chronomètre S1»
1A	4	3	A = 0011
1B	8	2	page 2
1C	C	3	A = B?
1D	9	0	saut 20 «clignotement alterné»
1E	8	4	page 4
1F	9	0	saut 40 «chronomètre S1/S2»

C3 94 83 90 47 81 C3 9A 83 94 43 82 C3 90 84 90

**Page 1** : sélection et démarrage d'un programme d'exemple

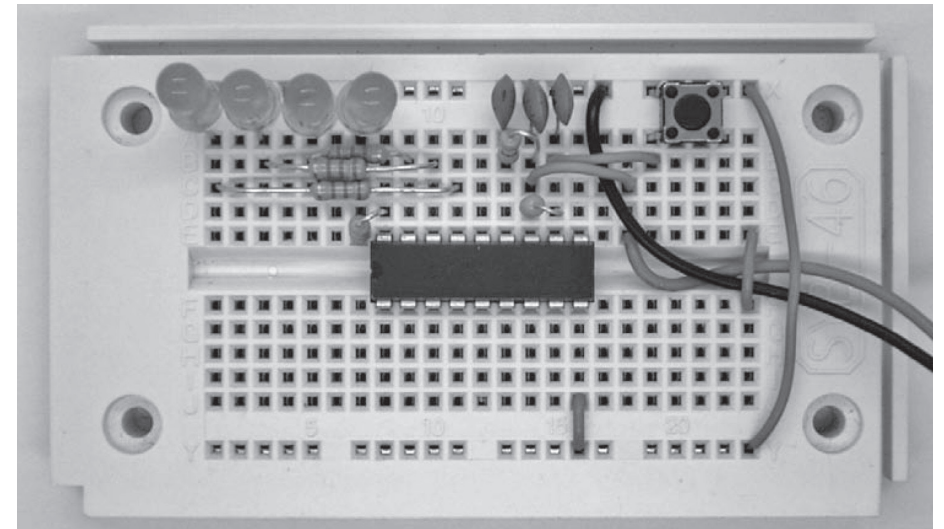
Adresse	Commande	Données	Remarque
20	1	1	0001 «clignotant alterné»
21	2	8	délai 500 ms
22	1	8	1000
23	2	8	délai 500 ms
24	3	4	Saut -4
25	7	1	A = A + 1 »incrémementation«
26	5	4	Port = A
27	5	9	PWM = A
28	2	6	Délai 100 ms
29	3	4	Saut -4
2A	6	9	A = AD1 »AD/PWM«
2B	5	4	Port = A
2C	5	9	PWM = A
2D	2	6	Délai 100 ms



**Img. 3** : quatre LED au niveau des sorties

Utilisez les rails d'alimentation supérieurs et inférieurs de la platine enfichable comme mise à la terre GND.

Le câble noir du compartiment à piles (pôle négatif) est à brancher ici. Le ligne positive VCC est à connecter avec le câble rouge du compartiment à piles. Vous devez absolument éviter les inversions de polarité. Elles peuvent entraîner la destruction du contrôleur. Intégrez un petit morceau de fill comme décharge de traction. L'alimentation branchée une fois doit, si possible, toujours rester connectée. Pour l'éteindre, enlevez pile du compartiment.



**Img. 4** : câblage minimum avec LED

Installez déjà le bouton Reset et branchez les quatre LED avec les résistances en série de 2,2 kΩ. Celles-ci sont utilisées pour les premiers tests de matériel. Attention à l'ordre. A1 est branché avec la LED gauche et A4 avec la droite. Vous obtenez ainsi un affichage binaire avec le bit le plus élevé à gauche. C'est tout particulièrement utile pour la future programmation.

## 2 Clignotant alterné

Installez les trois piles 1,5V ou bien trois accus NiMh dans le compartiment à piles. Vous pouvez ainsi démarrer le premier programme d'exemple avec un clignotant alternant avec les LED de gauche et de droite. La fréquence de clignotement est d'env. 1 Hz. La liste de programme affiche le programme simple sur cinq lignes seulement. Il commute alternativement la LED 1 et la LED 8. Entre deux se situent des commandes d'attente avec un délai de 0,5 s. Un retour au début permet de garantir la répétition continue du clignotement. Les différentes commandes sont expliquées plus précisément un peu plus loin. Vous pouvez toutefois déjà constater la simplicité de la programmation avec cet exemple. Le micrologiciel du contrôleur dispose d'un interpréteur, qui détecte et exécute les commandes simples. Les programmes sont donc bien plus compacts que dans bien d'autres systèmes.

L'exemple occupe la zone d'adresse à partir de 20h (décimal 32). Plusieurs programmes de la zone d'adresse supérieure peuvent également être démarrés plus tard avec vos propres applications. Les adresses peuvent aussi être remplacées par votre propre code de programmation. Le contrôleur peut, si besoin, également être réinitialisé. Les programmes d'exemple d'origine sont restaurés.

Adresse	Commande	Données	Remarque
20	1	1	LED 1
21	2	8	délai 500 ms
22	1	8	LED 8
23	2	8	délai 500 ms
24	3	4	Saut -4

Liste 1 : clignotant alternant

S'il n'y a pas de résultat, vérifiez d'abord la polarité des LED. Une mesure de certaines tensions peut également être utile. Utilisez p. ex. un multimètre numérique sur la plage de 20 V et laissez le raccordement négatif sur la terre. Toutes les tensions sont ainsi mesurées avec GND :

VCC : 4,5 V  
 Reset : 4,5 V  
 Osc1 : 1,5 V  
 E1 à E4 : 4,5 V  
 A1 : alternatif  
 A2, A3 : 0V  
 A3 : alternatif

## 3 Compteur binaire et sortie PWM

Toutes les entrées numériques sont tirées vers le haut au niveau de VCC via une résistance interne (résistance pull up) et ont une tension de repos au même niveau que la tension de service. Vous pouvez cependant appliquer chaque entrée à la terre via un fil ou un contact. Au démarrage, le programme par défaut lit l'état du port et l'analyse. Les différents branchements peuvent être effectués sur GND, entraînant la lecture d'un état zéro. Plusieurs programmes sont appelés en fonction du résultat.

La sortie PWM est traitée dans cet exemple comme un port numérique normal. Cela est nécessaire, car les quatre entrées A1 à A4 sont requises pour la saisie des chiffres. Les quatre LED s'éteignent après chaque saisie complète, afin de donner le moins d'informations possible sur la combinaison secrète à tout éventuel observateur.

Adresse	Commande	Données	Remarque
00	8	7	page 7
01	4	3	A = 3
02	5	1	B = A
03	D	0	call 70
04	C	3	Skip if A=B
05	3	0	fin
06	1	0	LED éteinte
07	4	5	A = 5
08	5	1	B = A
09	D	0	call 70
0A	C	3	Skip if A=B
0B	3	0	fin
0C	1	0	LED éteinte
0D	4	2	A = 2
0E	5	1	B = A
0F	D	0	call 70
10	C	3	Skip if A=B
11	3	0	fin
12	1	0	LED éteinte
13	4	F	A = 15
14	5	9	PWM=A
15	3	0	fin

87 43 51 D0 C3 30 10 45 51 D0 C3 30 10 42 51 D0 C3 30 10 4F 59 30

Liste 26 : la serrure à combinaison

## 21 Annexe

Liste des programmes d'exemple

Adresse	Commande	Données	Remarque
00	6	4	A = Din
01	5	1	B = A
02	4	E	A = 1110
03	8	0	page 0
04	C	3	A = B?
05	9	8	saut 08
06	8	2	page 2
07	9	5	saut 25, «incrémentation»
08	4	D	A = 1101
09	8	0	page 0



Adresse	Commande	Données	Remarque
25	7	1	A = A + 1
26	5	4	Port = A
27	5	9	PWM = A
28	2	6	Délai 100 ms
29	3	4	Saut -4

**Liste 2** : compteur binaire avec sortie LED et PWM

Le programme de comptage peut être utilisé comme programme d'exercice pour la lecture de nombres binaires. Ces derniers sont à maîtriser pour vos propres programmations. Chaque LED représente un bit. Un nombre à 4 bits peut être affiché. Les LED sont représentées dans le circuit selon leur valence : 8, 4, 2 et 1. Vous obtenez le nombre décimal grâce à l'addition de chaque valeur. Pour l'écriture hexadécimale, les nombres 10 à 15 sont représentés par les lettres majuscules A à F.

8	4	2	1	Décimal	Hexadécimal
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

Le programme peut également être utilisé comme clignotant pour différentes fréquences.

La sortie suivante la plus élevée possède soit la moitié de la fréquence soit une durée de période doublée :

A1 : 200 ms  
A2 : 400 ms  
A3 : 800 ms  
A4 : 1600 ms

Les valeurs numériques croissantes peuvent également être indiquées au niveau de la sortie PWM (modulation de largeur d'impulsion).

Le signal PWM est un signal carré avec une fréquence d'env. 16 kHz. La durée d'impulsions est alors commandée de façon à ce que le rapport impulsion-pause corresponde à la durée d'activation moyenne et à la luminosité des LED. Cette dernière est par ailleurs commandée sur 15 niveaux en zéro et luminosité pleine.

Un signal PWM peut être égalisé comme une tension continue à l'aide d'un filtre passe-bas RC. La sortie PWM est donc une sortie analogique. Grâce à ce programme, vous obtenez une tension

Adresse	Commande	Données	Remarque
00	1	0	LED 0000
01	4	5	A = 5
02	5	1	B = A
03	6	9	A = AD1
04	C	1	Skip if A>B
05	1	0	LED 0000
06	4	9	A = 9
07	5	1	B = A
08	6	9	A = AD1
09	C	2	Skip if A<B
0A	1	F	LED 1111
0B	3	A	saut -10

10 45 51 69 C1 10 49 51 69 C2 1F 3A

**Liste 23** : interrupteur crépusculaire avec hystérésis

## 19 Variateur LED

Le but de cet exemple de programme est une lampe à LED commandable. La luminosité d'une LED sur la sortie PWM doit pouvoir se régler via les boutons-poussoirs. Vous pouvez donc appuyer brièvement sur un bouton pour obtenir le niveau de luminosité suivant, ou bien exercez une pression continue pour modifier la luminosité en continu.

Les commandes skip que vous connaissez déjà interviennent au coeur du programme. Lorsqu'aucun bouton-poussoir n'est enclenché, la commande associée pour augmenter ou réduire le contenu de l'accu est ignorée.

La difficulté réside cependant dans le fait qu'un glissement de 15 à 0 ou de 0 à 15 peut se produire. Cela nécessite donc un peu plus d'effort pour empêcher ce glissement. Vous devez donc vous demander à chaque fois si l'extrémité inférieure (0) ou l'extrémité supérieure (15) est déjà atteinte ou non. Comme, en principe, un accu est utilisé lors d'une comparaison, son état doit être mis à chaque fois en mémoire tampon. La variable C est donc utilisée.

Adresse	Commande	Données	Remarque
00	8	0	AdrHi = 0
01	5	9	PWM = A
02	2	7	200 ms
03	5	2	C = A
04	4	F	A = 15
05	5	1	B = A
06	6	2	A = C
07	C	2	Skip if A<B
08	9	B	saut 0B
09	C	F	Skip if S2 = 1
0A	7	1	A = A + 1
0B	5	2	C = A
0C	4	0	A = 0



Le résultat de ce programme est un compteur binaire décroissant avec des temporisations irrégulières. Testez-le également pour d'autres commandes du sous-programme.

Les programmes d'exemples disponibles par défaut disposent de plusieurs sous-programmes utiles pour une utilisation générale. Ils sont tous listés en annexe. Vous devez connaître l'adresse d'entrée pour leur utilisation :

- 50 : Sous-programme : son long
- 52 : Sous-programme : son court
- 53 : Sous-programme : son au choix, longueur en A
- 60 : Sous-programme : délai lors de la pression sur S1
- 68 : Sous-programme : délai lors de la pression sur S2
- 70 : Sous-programme : saisie de chiffres avec S1 et S2

Le sous-programme à partir de l'adresse 60 est uniquement utilisé pour créer un compteur commandé via le bouton-poussoir S1. Le compteur commence à 0. Le programme principal est relativement court, car les tâches complexes de la requête de touche a été transférée dans le sous-programme.

Adresse	Commande	Données	Remarque
00	4	0	A = 0
01	5	4	sortie
02	7	1	A = A+1
03	8	6	page 6
04	D	0	Call 60, bouton S1
05	3	4	saut -4

40 54 71 86 D0 34

**Liste 22** : compteur commandé via S1

Testez le programme. Lorsque vous appuyez dix fois sur S1, vous devez obtenir le résultat 1010. Modifiez le programme de façon à ce que le sous-programme soit utilisé à partir de l'adresse 68. Le compteur réagit désormais à S2.

## 18 Interrupteur crépusculaire

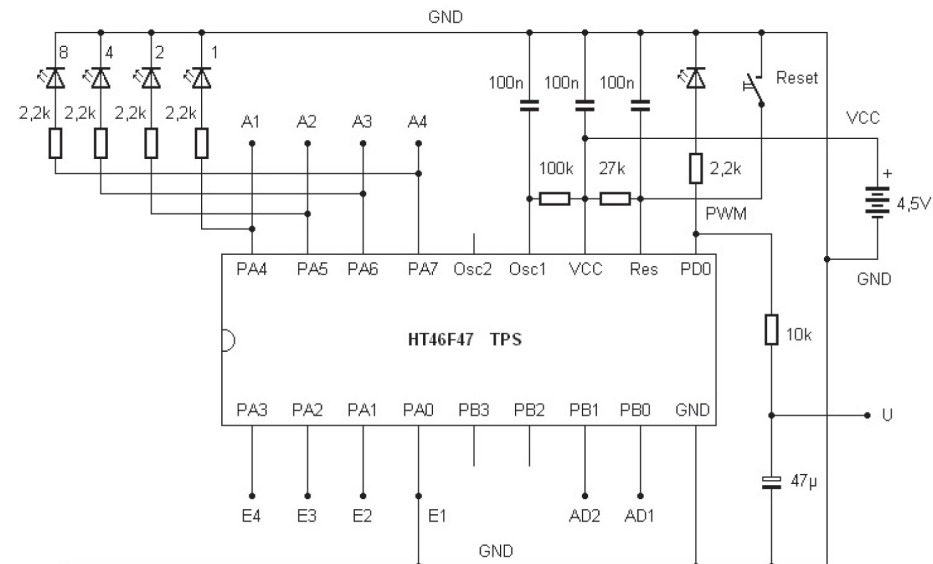
Un interrupteur crépusculaire doit allumer la lumière lorsque la luminosité ambiante est inférieure à un certain seuil. Lorsqu'il fait plus clair, la lumière doit de nouveau s'éteindre. Il convient de vérifier que la lumière ne vacille pas à la frontière clair/sombre. Cela est possible avec une hystérésis, soit un écart certain entre la luminosité d'activation et de désactivation. Le programme présenté ici fonctionne selon les règles suivantes :

- Lorsque la tension à AD1 n'est pas supérieure à 5, il est désactivé.
- Lorsque la tension à AD1 n'est pas inférieure à 9, il est activé.

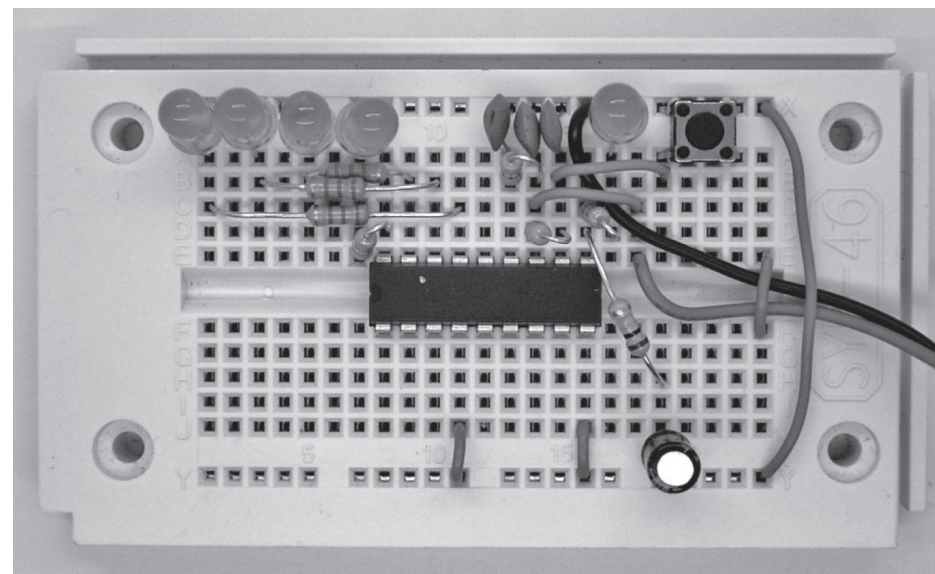
Il existe donc une plage centrale, dans laquelle aucune modification de l'état de sortie ne peut se produire. Cet écart empêche ainsi un scintillement des LED.

- 0-5 : LED éteintes
- 6-8 : LED inchangées
- 9-15 : LED allumées

continue croissante progressive de 0 V à 4,5 V. Suivez le tracé de la tension avec i, appareil de mesure ou un oscilloscope.



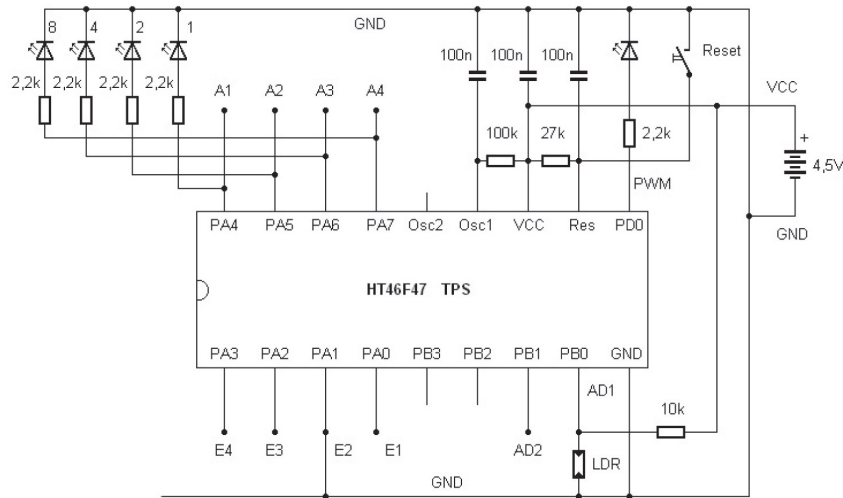
Img. 7 : filtre passe-bas sur la sortie PWM



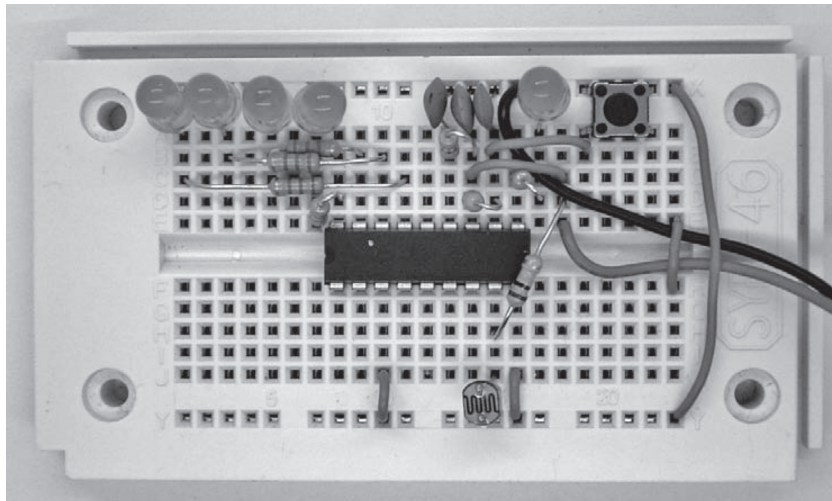
Img. 8 : tension de sortie PWM aplatie

## 4 Convertisseur analogique numérique

Avec la connexion de E2 sur GND et une pression sur le bouton Reset, vous démarrez un petit programme d'exemple de convertisseur analogique numérique. La tension analogique sur l'entrée analogique AD1 est mesurée et est transformée en une valeur numérique. Comme le contrôleur TPS fonctionne en continu avec des valeurs 4 bits, le résultat de la conversion analogique numérique est un nombre situé dans la plage 0 à 15. Le résultat 0 est pour la tension d'entrée 0, le résultat 15 est pour une tension correspondant à la tension de service, soit, p. ex. 4,5 V. La valeur analogique numérique est indiquée comme nombre binaire au quatre LED et transmis aussi à la sortie PWM. Branchez un répartiteur de tension, constitué d'une résistance fixe et d'une photorésistance (LDR) à l'entrée analogique AD1.



Img. 9 : branchement d'un capteur lumineux



Img. 10 : la LDR à l'entrée AD1

Adresse	Commande	Données	Remarque
00	6	4	A = Din
01	5	1	B = A
02	4	3	A = 3
03	7	7	A = A And B
04	5	4	port = A
05	3	5	saut -5

64 51 43 77 54 35

Liste 20 : utilisation de la fonction AND

Modifiez le programme et testez les autres fonctions logiques. La fonction OR (78) peut être utilisée pour positionner fondamentalement certains états d'entrée sur 1 : 64 51 43 78 54 35

1010 OR

0011 =

1011

La fonction XOR (ou exclusif, 79) permet d'inverser des bits : 64 51 43 79 54 35

1010 XOR

0011 =

1001

## 17 Sous-programmes

Lorsqu'une partie d'un programme est utilisée plusieurs fois, elle doit être écrite dans un sous-programme.

Vous économisez ainsi de l'espace mais également un fastidieux travail de saisie. L'exemple suivant illustre l'utilisation d'un sous-programme, appelé à deux moments dans un programme principal. Le sous-programme contient ici uniquement une instruction (A = A-1) et la commande de retour. Aucun espace n'est donc économisé ici, seulement l'exemple sert uniquement à illustrer les commandes CALL et RET.

Programme principal :

Adresse	Commande	Données	Remarque
00	8	0	AdrHi = 0
01	D	8	Call 08
02	5	4	sortie
03	2	9	délati 1 s
04	D	8	Call 08
05	5	4	sortie
06	2	8	délati 0,5 s
07	3	7	saut -7

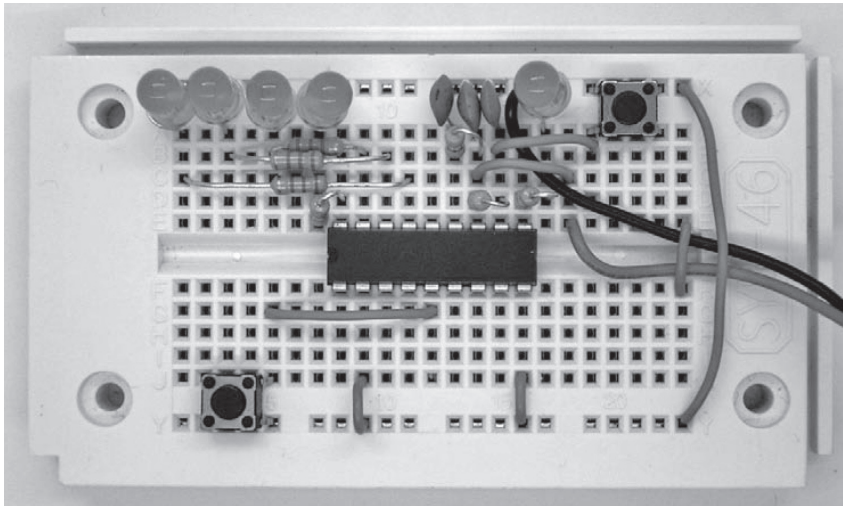
Sous-programme :

Adresse	Commande	Données	Remarque
08	7	2	A = A-1
09	E	0	Ret

80 D8 54 29 D8 54 28 37 72 E0

Liste 21 : appel d'un sous-programme





Img. 12 : pont entre E3 et GND

Le programme utilise une instruction de saut conditionnel. Lorsque l'état de l'entrée 1 est en marche, la commande suivante est ignorée. Lorsque vous appuyez sur la touche, l'état est zéro, l'augmentation de la variable A est exécutée. Cela conduit à une incrémentation rapide de l'état de sortie. Lorsque vous relâchez, le dernier relevé de compteur est maintenu. Compte tenu de la vitesse de comptage importante, il n'y a pas d'influence possible sur le résultat. Celui-ci est donc aléatoire.

Adresse	Commande	Données	Remarque
30	5	4	Port = A
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Saut -3

Liste 4 : générateur aléatoire

Appuyez brièvement sur la touche, pour obtenir un nouveau résultat aléatoire. Testez la fonction aléatoire, en créant une statistique des résultats. Après un nombre suffisamment important de cycles, il s'avère que tous les résultats sont à peu près à fréquence égale. Le programme est idéal pour les jeux, où, p. ex., le nombre 1111 doit être «tiré».

Le programme est en même temps un compteur avec une vitesse de fonctionnement la plus rapide possible, car aucune instruction d'attente n'est utilisée. Cet exemple vous permet donc d'analyser la vitesse de fonctionnement du contrôleur TPS. Tant que la touche est enfoncée, un signal carré apparaît au niveau de la sortie A1 avec une fréquence d'env. 133 Hz et une durée de période de 7,5 ms. Le port change donc son état tous les 3,75 ms. Le programme parcourt dans la boucle de comptage quatre instructions. Environ une milliseconde est utilisée par instruction. La dernière sortie A4 présente une fréquence de 16,6 Hz, détectable par le scintillement visible.

Lorsque des vitesses de fonctionnement plus élevées sont requises à titre exceptionnel pour des opérations urgentes, vous avez la possibilité d'augmenter la cadence d'horloge du contrôleur en diminuant la résistance au niveau de Osc 1. Vous obtenez donc une fréquence d'horloge de 2 MHz pour 100 kΩ. Remplacez la résistance par une de 27 kΩ. Vous obtenez ainsi une fréquence d'horloge jusqu'à quatre fois plus rapide et un délai d'instruction d'env. 0,25 ms. En temps normal, le contrôleur doit cependant fonctionner avec 100 kΩ au niveau de Osc 1. Une faible consommation d'énergie et un fonctionnement sûr sont ainsi garantis, même pour un très petite tension de fonctionnement, jusqu'à 2,2 V.

Adresse	Commande	Données	Remarque
00	4	5	A = 5
01	5	2	C = A
02	1	5	port = 0101
03	2	8	500 ms
04	1	A	Port = 1010
05	2	8	500 ms
06	8	0	page 0
07	A	2	C x 02
08	3	0	fin

45 52 15 28 1A 28 80 A2 30

Liste 17 : une boucle de comptage

Testez le programme. Les LED indiquent à chaque passage les exemples 0101 et 1010. Cependant, cette partie du programme ne sera manifestement pas exécutée cinq fois, mais bien six fois. La commande de saut est certes exécutée cinq fois dans l'adresse 07, mais pour accéder la première fois à cet emplacement, un processus de clignotement a déjà été effectué. C'est pourquoi le programme clignote au total six fois.

Modifiez la variable de comptage sur la valeur 4 et testez de nouveau le programme. Les LED clignotent désormais cinq fois exactement.

Vous pouvez également utiliser la boucle de comptage pour effectuer non pas un saut en arrière mais en avant. Cette fois, l'opération est bien exécutée cinq fois, lorsque C a été chargé avec la valeur 5 au début. Chaque adresse 04 ignorée contient un saut relatif sur elle-même et donc une boucle sans fin qui sert de fin de programme.

Adresse	Commande	Données	Remarque
00	4	5	A = 5
01	5	2	C = A
02	8	0	AdrHi = 0
03	A	5	C x 05
04	3	0	fin
05	1	5	port = 0101
06	2	8	délai 500 ms
07	1	A	Port = 1010
08	2	8	délai 500 ms
09	3	6	Saut -6

45 52 80 A5 30 15 28 1A 28 36

Liste 18 : clignotant x 5

## 15 Comparaisons

Deux valeurs numériques doivent être comparées. En fonction du résultat de la comparaison, un saut est exécuté. Les deux valeurs numériques se situent en A et B. B est chargé avec le chiffre 5 dans l'exemple ci-dessous. A reçoit son résultat par l'entrée analogique AD1. Vous pouvez ici, p. ex., brancher un photocapteur comme dans le chapitre 4. Le programme doit alors en permanence exécuter l'action suivante :

### 13 Aperçu des commandes

Toutes les commandes en un seul coup d'oeil - pour simplifier le travail avec le contrôleur. Le tableau suivant contient l'ensemble du stock de commandes sous forme compacte.

1	2	3	4	5	6	7	8	9	A	B	C	D	E
Port=	Wait	Jump	A=	... = A	A = ...	A = ...	Page	Jump	C*	D*	Skip if ...	Call	Ret
0	0	1 ms	0	0			0	0	0	0		0	
1	1	2 ms	1	1	B = A	A = B	1	1	1	1	A>B	1	
2	2	5 ms	2	2	C = A	A = C	2	2	2	2	A<B	2	
3	3	10 ms	3	3	D = A	A = D	3	3	3	3	A = B	3	
4	4	20 ms	4	4	Dout = A	A = Din	4	4	4	4	Din.0 = 1	4	
5	5	50 ms	5	5	Dout.0 = A.0	A = Din.0	5	5	5	5	Din.1 = 1	5	
6	6	100 ms	6	6	Dout.1 = A.0	A = Din.1	A = A/B	6	6	6	Din.2 = 1	6	
7	7	200 ms	7	7	Dout.2 = A.0	A = Din.2	A = A And B	7	7	7	Din.3 = 1	7	
8	8	500 ms	8	8	Dout.3 = A.0	A = Din.3	A = A Or B	8	8	8	Din.0 = 0	8	
9	9	1 s	9	9	PWM = A	A = AD1	A = A Xor B	9	9	9	Din.1 = 0	9	

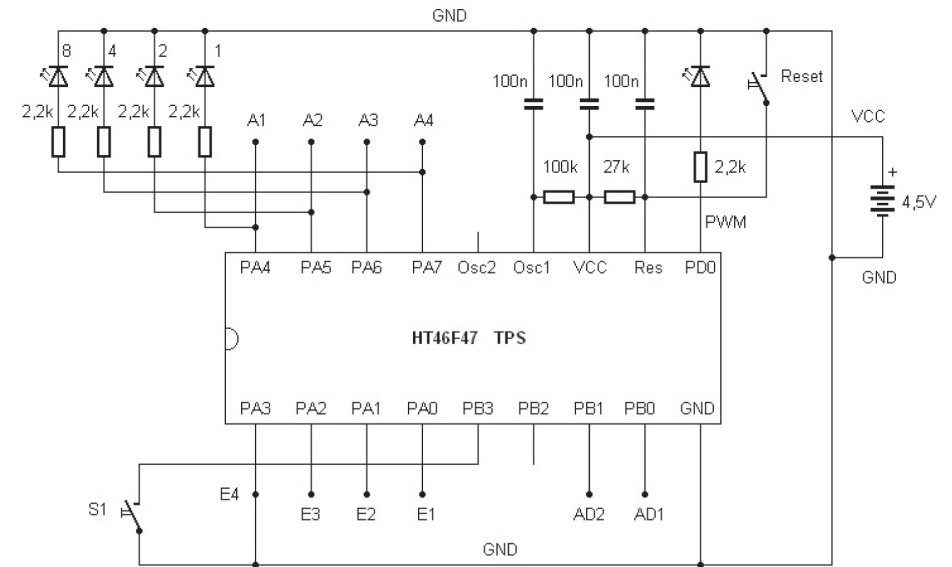
A	B	C	D	E	F
10	2 s	10	10		A = AD2
11	5 s	11	11		A = Not A
12	10 s	12	12		A = Not A
13	20 s	13	13		A = Not A
14	30 s	14	14		A = Not A
15	60 s	15	15		A = Not A

### 14 Boucles de comptage

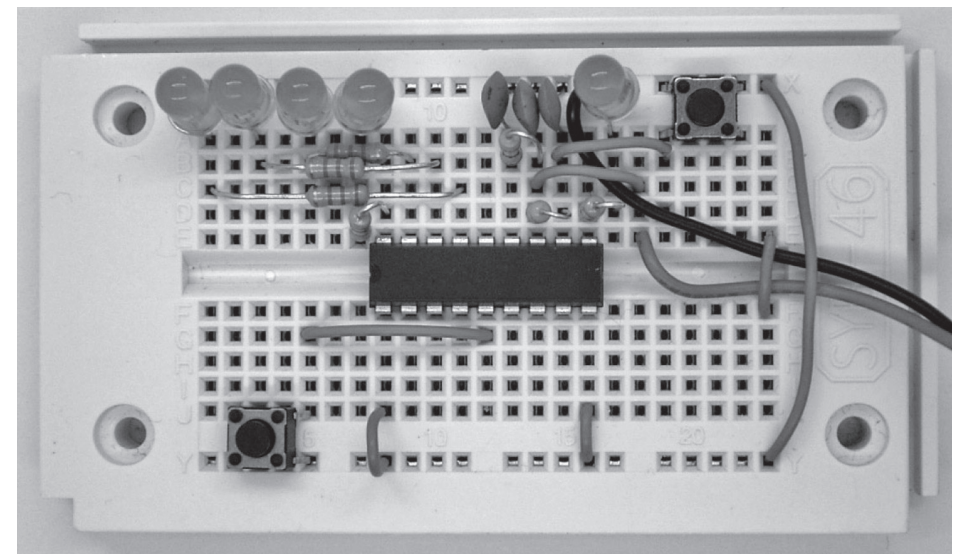
Un processus doit, p. ex., être exécuté cinq fois précisément. Vous devez donc pour cela faire une boucle de comptage. Dans ce cas, une commande de saut est exécutée cinq fois, pas plus. La variable de comptage s'intitule C. La valeur du compteur 5 doit d'abord être chargée dans A et de là dans C. La commande A2 exécute un saut absolu sur 02 et réduit en même temps le contenu de la variable C à 1. Lorsque C a atteint la valeur 0, le saut n'est plus exécuté. L'adresse de saut absolu correspond à la page indiquée. Lors d'un programme sur la page 0, la commande de page 80 doit également être omise. Cela n'est pas absolument nécessaire lorsqu'une autre page est démarrée.

### 6 Mesure des largeurs d'impulsion

Avec E4 sur GND et après un reset, un programme d'exemple est lancé pour mesure la largeur d'une impulsion. L'état est, ici aussi, analysé au niveau de l'entrée S1.



Img. 13 : E4 sur GND



Img. 14 : démarrage de la mesure de la largeur d'impulsion

Le chronométrage démarre lorsque état S1 = 0, soit lorsque la touche est enfoncée. Pour un délai d'attente de 5 ms, s'ajoute encore une fois env. 5 ms pour l'exécution d'au total cinq commandes dans la boucle de comptage. L'unité de temps de la mesure est donc 10 ms.

Adresse	Commande	Données	Remarque
34	2	2	délai 5 ms
35	C	C	S1 = 0?
36	3	2	Saut -2
37	4	0	A = 0
38	2	2	délai 5 ms
39	7	1	A = A + 1
3A	5	4	Port = A
3B	C	E	S1 = 1?
3C	3	4	Saut -4
3D	3	9	Saut -9

**Liste 5 :** mesure du temps

Appuyez le plus brièvement possible sur la touche S1. Vous obtenez, p. ex., le résultat 1010, soit en décimal 10. Comme l'unité de temps du programme est de 10 ms, l'affichage est donc de 100 ms. Avec un peu d'entraînement, vous pouvez également obtenir des périodes plus courtes, jusqu'à 50 ms.

## 7 Lire le programme

Les boutons S1 (entrée de données, gauche) et S2 (programmer, droite) sont utilisés pour la programmation.

Par ailleurs, le bouton reset est nécessaire. Seul avec ces boutons, les programmes peuvent être lus et n'importe quel programme peut être saisi. Il est possible d'entrer de nouveaux programmes ou de modifier des programmes déjà existants en un très bref délai en s'exerçant un peu.

Vous accédez au mode de programmation en appuyant sur le bouton reset tout en maintenant la touche de programmation S2 enfoncée. Ce n'est qu'une demi seconde après le reset que vous pouvez relâcher S2. Vous pouvez faire défiler le programme existant et consulter les instructions et les données avec la touche S2 seule. Chaque adresse nécessite à cet effet deux actionnements de S2. Vous commutez ainsi entre l'affichage de la commande et les données. Par ailleurs, l'adresse actuelle est affichée brièvement chaque fois.

Adresse	Commande	Données	Remarque
00	C	C	S1 = 0?
01	3	1	saut -1
02	4	0	A = 0
03	7	1	A = A + 1
04	5	4	port = A
05	C	E	S1 = 1?
06	3	3	saut -3
07	3	7	saut -7

CC 31 40 71 54 CE 33 37

**Liste 15 :** réaction sur le bouton S1

La commande de saut CC dans l'adresse 00 analyse l'état au niveau du bouton-poussoir S1. En état de veille, S1 = 1. La condition n'est donc pas possible et la commande n'est pas sautée dans l'adresse 01. Une commande de saut relative est disponible au début. Le programme répète la commande dans l'adresse 00 et 01 tant que le bouton-poussoir est enfoncé. La condition devient alors possible et l'adresse 01 est sautée. Le véritable processus de mesure commence alors. L'accu est enlevé, puis toujours augmenté à 1 et indiqué au niveau des LED. Une autre commande de saut conditionnel CE est disponible à l'adresse 05. La condition pour le saut d'une commande est ici S1 = 1. Comme la touche est toujours enfoncée, la condition n'est pas possible. La commande est alors exécutée dans 06 et conduit à un retour vers 03. Ce n'est que lorsque la touche est relâchée que le programme accède à l'adresse 07 et donc à un retour au début.

Entrez le programme et testez-le. Le temps de réponse est désormais beaucoup plus rapide. L'unité de temps est d'env. 5 ms.

Le programme d'exemple d'origine se trouve encore en mémoire à partir de l'adresse 34h, car les adresses inférieures ont été écrasées. Ecrivez un petit programme, contenant uniquement un saut à cette adresse. Vous devez d'abord renseigner ici la page 3. Le saut absolu suivant et l'adresse 4 indiquée cible la véritable adresse 34.

Adresse	Commande	Données	Remarque
00	8	3	page 3
01	9	4	Adresse = 34

83 94

**Liste 16 :** saut absolu pour le programme de chronométrage

Le programme d'exemple d'origine est ainsi de nouveau appelé. Testez-le également pour d'autres exemples. Retrouvez un aperçu complet de tous les programmes utilisables dans l'annexe.

## 12 Sauts et branchements

Il y avait jusqu'à présent un simple retour (commande 3) qui remontait 15 adresses max. Maintenant s'ajoute également un saut absolu (jump). Comme la destination de saut ne peut indiquer que 4 bits, il faut une commande supplémentaire qui détermine le nibble élevé de l'adresse. C'est pourquoi on obtient l'espace d'adressage 0-255.

C'est plus que nécessaire, car l'EEPROM du contrôleur contient uniquement 128 octets, soit la plage 00 à 7F (décimal 0 à 127). La mémoire est donc répartie en huit pages, de la page 0 à la page 7 (page 0-7).

La page de la destination de saut doit être déterminée avec un saut absolu.

Deux boucles de comptage avec les variables C et D effectuent des sauts absolus. La page de l'adresse doit également être définie au préalable.

Les sauts conditionnels fonctionnent comme commande «skip» (omettre, ignorer). Lorsque chaque condition est possible, l'adresse est ignorée. Une commande de saut ou une commande de calcul peuvent, p. ex., être disponibles. Les comparaisons entre A et B ainsi que la requête directe de bit du port d'entrée sont disponibles comme conditions.

Il existe en outre encore un appel de sous-programme (Call) et la commande de retour associée (Return). Plusieurs sous-programmes sont autorisés, mais aucun sous-programme supplémentaire ne doit être appelé depuis un sous-programme. En effet, l'interpréteur ne se souvient que d'une adresse de retour.

80-8F: Adr-high = 0-15

90-0F : saut direct (Jump) sur Adr-high, Adr-low (0-15)

A0-AF : boucle de comptage Cx Adr-high, Adr-low (0-15)

A0-AF : boucle de comptage Dx Adr-high, Adr-low (0-15)

C1-CF: saut conditionnel : lorsque (condition 1-15) alors ignorer

C1: if A > B then Adr = Adr + 1

C2: if A < B then Adr = Adr + 1

C3: if A = B then Adr = Adr + 1

C4: if Din.0 = 1 then Adr = Adr + 1

C5: if Din.1 = 1 then Adr = Adr + 1

C6: if Din.2 = 1 then Adr = Adr + 1

C7: if Din.3 = 1 then Adr = Adr + 1

C8: if Din.0 = 0 then Adr = Adr + 1

C9: if Din.1 = 0 then Adr = Adr + 1

CA: if Din.2 = 0 then Adr = Adr + 1

CB: if Din.3 = 0 then Adr = Adr + 1

CC: if S1 = 0 then Adr = Adr + 1

CD: if S2 = 0 then Adr = Adr + 1

CE: if S1 = 1 then Adr = Adr + 1

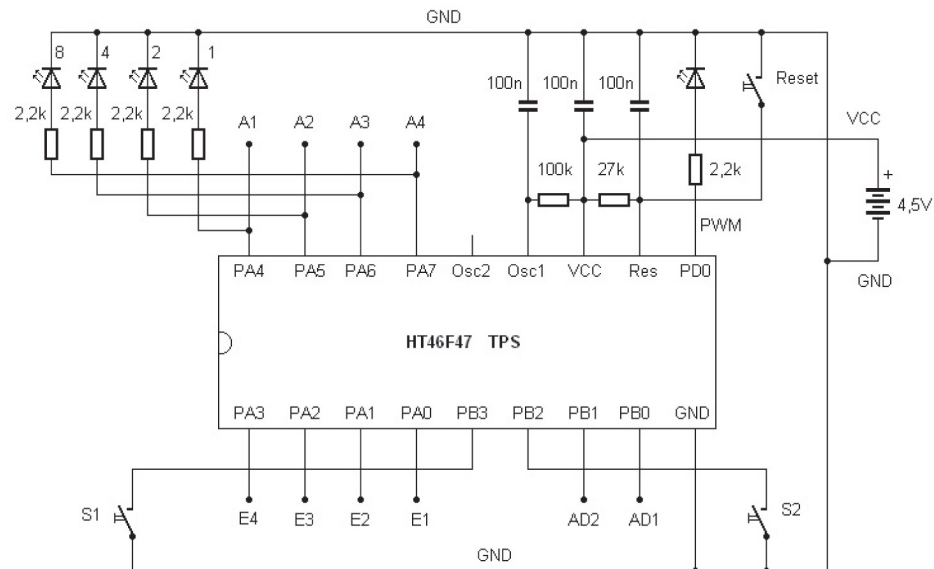
CF: if S2 = 1 then Adr = Adr + 1

D0-DF : appel sous-programme (Call) Adr-high, Adr-low (0-15)

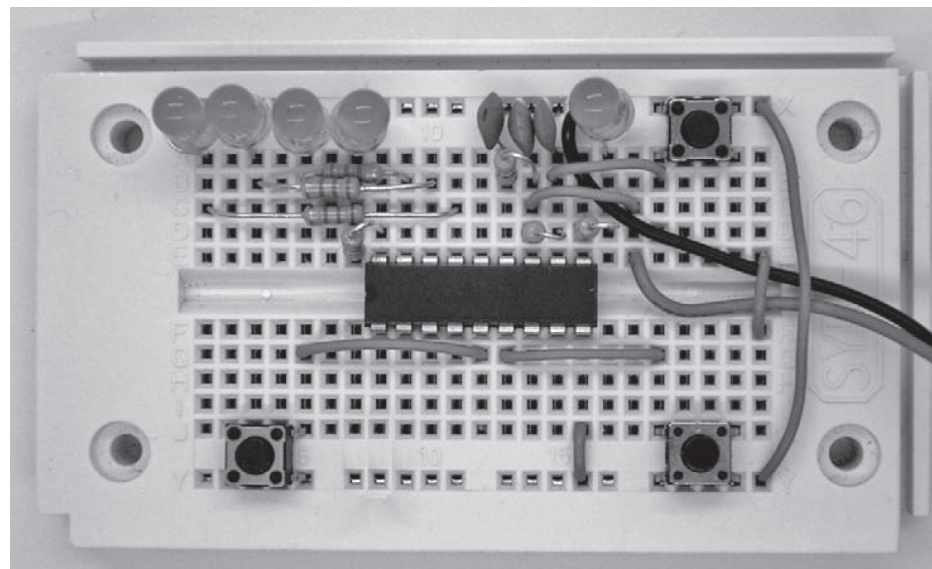
E0-EF : retour du sous-programme (Return)

Retrouvez un exemple d'utilisation de la commande de saut conditionnel dans le programme d'exemple du chapitre 6. Il a été placé légèrement modifié à l'adresse 0. Comme la partie supérieure de l'adresse (Adr-hi) est en état de repos 0, le contrôle commence alors à la page 0. La commande 80 ne doit pas être utilisée ici.

La longueur d'une pression est de nouveau mesurée et affichée. Toutes les commandes d'attente ont été supprimées du programme, de sorte qu'il fonctionne désormais avec une résolution temporelle plus élevée.



Img. 15 : S1 et S2 pour le mode de programmation



Img. 16 : trois boutons-poussoirs et témoin LED

- Première du bouton S2
- Afficher adresse (inférieure à 4 bits), 300 ms
- Affichage éteint, 300 ms
- Afficher commande
- Deuxième pression du bouton S2

- Afficher les données
- Troisième pression du bouton S2
- Afficher l'adresse suivante, 300 ms
- etc.

Lorsque, p. ex., vous souhaitez afficher uniquement un programme existant avec cinq étapes, mais sans le modifier, vous y accédez avec pas moins de 10 actionnements de S2 jusqu'à la fin. L'orientation se fait très facilement puisque l'adresse actuelle est à chaque fois affichée brièvement. Vous savez ainsi toujours si l'affichage en cours représente une commande ou des données. En état par défaut (à la livraison), les instructions suivantes sont dans les cinq premières adresses. Il s'agit ici du début du programme de sélection pour démarrer chaque programme d'exemple.

Adresse	Commande	Données	Remarque
00	6	4	A = Din
01	5	1	B = A
02	4	E	A = 14
03	8	0	AdrHi = 0
04	C	3	A = B?

**Liste 6** : code de programmation à l'état initial

Une commande de 4 bits et les données 4 bits associées forment ensemble un octet, soit un nombre à 8 bits. Un demi-octet est également appelé «nibble». Le nibble de valeur plus élevée forme l'instruction, le nibble de valeur plus faible forme les données associées. L'EEPROM du contrôleur contient au total 128 octets. Un programme peut ainsi posséder 128 instructions max. Cela suffit pour la plupart des applications, car le code de programmation est extrêmement compact. De nombreux programmes utiles s'en tirent avec moins de dix commandes. Affichez chaque commande et données et comparez-les avec le contenu de la mémoire. Appuyez ensuite de nouveau sur le bouton reset. L'ancien programme démarre de façon inchangée.

## 8 Saisir des programmes

La touche S1 est uniquement utilisée lorsque la commande ou ses données doivent être modifiées ou de nouveau saisies. En principe, seules des valeurs numériques entre 0 et 15 peuvent être saisies. La première pression permet de régler S1 sur 0. Chaque pression suivante sur le bouton augmente le nombre de 1. L'état actuel est affiché à chaque fois par les 4 LED de façon binaire. Si, p. ex., vous souhaitez entrer 4, vous devez appuyer cinq fois sur S1 : 0, 1, 2, 3, 4. L'affichage binaire est donc alors 0100.

Si l'instruction ou les données ou les deux ont été saisis de cette manière, la deuxième pression de bouton de S2 fait en sorte que cet octet soit programmé dans l'EEPROM. C'est-à-dire que le témoin LED est désactivé pendant 600 ms, avant que l'adresse suivante, puis ensuite la commande suivante ne s'affichent. Cette petite pause doit être comprise intuitivement comme processus de programmation. Vous pouvez ainsi vous représenter que le système économise de l'énergie pour l'affichage et l'utilise pour la programmation de l'EEPROM. Cela fonctionne un peu de la même manière dans les voitures : lorsque vous actionnez le démarreur, la lumière et la radio s'éteignent brièvement.

Il est donc possible de modifier un seul point d'un programme existant. Accédez à l'emplacement souhaité avec S2 et modifiez avec S1 les commandes ou les données, afin de les mémoriser ensuite avec S2.

Un programme avec seulement deux commandes est saisi pour le premier test. Trois LED s'allument et s'exécutent ensuite dans une boucle infinie.

64 : A = Din  
 65 : A = Din.0  
 66 : A = Din.1  
 67 : A = Din.2  
 68 : A = Din.3  
 69 : A = AD1  
 6A : A = AD2

71–7A : A = imprimé 1–10  
 71 : A = A + 1  
 72 : A = A - 1  
 73 : A = A + B  
 74 : A = A - B  
 75 : A = A \* B  
 76 : A = A / B  
 77 : A = A And B  
 78 : A = A Or B  
 79 : A = A Xor B  
 7A : A = Not A

Retrouvez un exemple d'utilisation de la variable A sous les exemples de programme du chapitre 3. Le programme a été positionnée ici à l'adresse zéro et légèrement développé. Un début défini avec la valeur 0 est ajouté dans la variable A. Une instruction d'opération se trouve dans l'adresse 01, il s'agit ici d'une augmentation de 1. Le contenu de la variable A est ensuite transféré à la sortie PWM et au port de sortie.

Adresse	Commande	Données	Remarque
00	4	0	A = 0
01	7	1	A = A + 1
02	5	4	port = A
03	5	9	PWM = A
04	2	6	délai 100 ms
05	3	4	saut -4

40 71 54 59 26 34

**Liste 13** : augmenter de 1

Retrouvez un autre exemple dans le chapitre 4 Les données proviennent alors de l'entrée analogique AD1 et sont transférées au port de sortie et à la sortie PWM. Le programme modifié contient encore une étape de calcul supplémentaire, à savoir l'inversion du contenu de la variable A. Vous obtenez ainsi de la valeur 0000 la nouvelle valeur 1111, c'est-à-dire que 0 devient 15 et vice-versa. La tension d'entrée croissante mène de ce fait à une sortie PWM décroissante.

Adresse	Commande	Données	Remarque
00	6	9	A = AD1
01	5	4	port = A
02	7	A	A = Not A
03	5	9	PWM = A
04	2	6	délai 100 ms
05	3	5	saut -5

69 54 7A 59 26 35

**Liste 14** : inverser



Adresse	Commande	Données	Remarque
00	1	F	LED 1111
01	2	F	délai 1 min
02	1	0	LED 0000
03	3	0	fin
1F	2F	10	30

**Liste 12** : minuterie pour une minute

## 11 Calcul avec variables

Jusqu'à présent, seules des valeurs constantes ont été utilisées dans les paramètres de chaque commande. C'est utile uniquement lorsqu'un programme doit toujours fonctionner de la même manière. En revanche, les programmes plus complexes fonctionnent avec des données variables. Vous pouvez ainsi, p. ex., exécuter un calcul tel que  $A = A + B$ . Selon le contenu des variables A et B, il en résultera chaque fois quelque chose de différent. Le résultat pourrait, p. ex., commander les LED au niveau des sorties.

La commande possède les quatre variables A, B, C et D. La variable la plus importante est A. Elle est également désignée par un accumulateur ou accu. A est utilisé pour toutes les opérations de calcul et reçoit le résultat du calcul. De plus, A est également utilisé pour le transport de données. B est principalement requis pour les opérations de calcul. C et D peuvent servir de mémoire tampon et peuvent être utilisés plus tard comme compteur pour les boucles de comptage.

Il existe par ailleurs deux entrées analogiques (AD1 et AD2) et une sortie PWM. Les données traitées sont limitées à 4 bits et uniquement accessibles via la variable A ( $A = AD1$ ,  $PWM = A$ ). L'accu A peut également être directement chargé avec un nombre (commande 40-4F). Pour remplir B, C ou D, vous devez d'abord charger A puis affecter le contenu des autres variables (commandes 51-53). Certaines étapes de calcul (commande 71-7A) peuvent être exécutées à l'aide de A et B.

Les commandes 40-4F attribuent une nouvelle valeur à A. Le groupe de commande 51-5A transmet le contenu de A à une cible telle qu'une autre variable ou une sortie PWM. Ce groupe contient également des commandes, qui déterminent un bit unique du port de sortie.

Justement les autres directions de données sont présentes avec le groupe de données 61-6A, où les données d'une source sont lues en A. Le groupe de commande 71-7A effectue certaines opérations de calcul. Le résultat s'affiche principalement en A. Le port de sortie Dout comprend les quatre sorties A1 à A4, qui peuvent être attribuées soit ensemble, soit par bits Dout.0 à Dout.3 uniques. Les entrées E1 à E4 réagissent de la même façon qu'un port d'entrée Din.

40-4F :  $A = 0-15$

51-5A : cible 1-9 = A

51 :  $B = A$

52 :  $C = A$

53 :  $D = A$

54 :  $Dout = A$

55 :  $Dout.0 = A.0$

56 :  $Dout.1 = A.0$

57 :  $Dout.2 = A.0$

58 :  $Dout.3 = A.0$

59 :  $PWM = A$

61-6A : A = source 1-10

61 :  $A = B$

62 :  $A = C$

63 :  $A = D$

Adresse	Commande	Données	Remarque
00	1	7	A1-4 = 0111
01	3	0	Saut 0

**Liste 7** : mise sous tension des LED

Vous pouvez également sélectionner une brève notation au lieu d'une liste détaillée. Les deux octets sont alors rassemblés en nombres hexadécimaux : 17h, 30h. La notation hexadécimale est en principe utilisée ci-dessous. Les programmes sont donc écrits en courte notation sans symboles hexadécimaux : 17 30

La saisie nécessite que vous appuyiez sur :

S2 + Reset

2 x S1

S2

8 x S1

S2

4 x S1

S2

1 x S1

S2

Si vous appuyez une fois de trop sur le bouton S1, vous pouvez tout de même atteindre le bon nombre. Vous devez aller au-delà de 15, car vous passez ensuite à la valeur 0.

Lorsque la saisie est complète, le nouveau programme est démarré à l'aide du bouton reset. On voit donc que trois LED sont allumées. Plus rien ne se passe ensuite. Le contrôleur ne réagit également plus aux états des entrées E1 à E4, car le programme d'origine a été partiellement écrasé. Les programmes de démonstration ne peuvent ainsi plus être démarrés.

Cependant, comme vous avez seulement modifier les deux premières adresses de mémorisation, vous pouvez facilement retourner au programme d'origine. Pour cela, saisissez uniquement les deux premières commandes (64 51) correspondant à la liste du dernier paragraphe.

Testez la fonction d'origine des programmes d'exemple. Saisissez au mieux le nouveau programme d'entraînement. Vous obtenez ainsi plus de confiance lors de la saisie de programmes.

## 9 Restauration des programmes d'exemple

La restauration de l'état initial du contrôleur est possible avec la saisie de deux octets FF. Il s'agit en fait de l'état de l'EEPROM vierge.

Le micrologiciel du contrôleur TPS contient une fonction de démarrage, qui vérifie tout d'abord les deux premières adresses, pour détecter une mémoire vide. Lorsque les deux octets FF sont lus ici, le contrôleur considère qu'aucun programme n'a encore été saisi. Dans ce cas, l'EEPROM est automatiquement rempli avec le logiciel d'exemple. Cette fonction permet en fait de joindre les programmes d'exemple de l'EEPROM au contrôleur lors de son premier démarrage. Elle peut cependant être utilisée à tout moment pour restaurer l'état initial.

Adresse	Commande	Données	Remarque
00	F	F	-
01	F	F	-

**Liste 8** : restauration de l'état initial

Démarrez alors le mode de programmation avec reset en maintenant le bouton S2. Indiquez ensuite quatre fois la valeur F (décimale 15), à laquelle se trouvent toutes les LED A1 à A4. Terminez également la dernière saisie avec S2.

Appuyez sur le bouton Reset. Le contrôleur n'a besoin que d'un moment plus long que d'habitude pour programmer de nouveau tous les octets des programmes d'exemple. L'état initial est donc restauré. Testez, p. ex., le clignotant alternatif sans pont sur les entrées.

## 10 Commandes de base TPS

Le contrôle commandable par touches comprend 14 commandes au total (1-14). Bon nombre de ces commandes comprend un paramètre sous la forme d'un nombre 4 bits de 0000 à 1111 (0-F), avec donc une plage de nombres allant jusqu'à 15 (décimal). D'autres instructions comprennent des sous-fonctions, indiquées sous la forme de paramètre. Jusqu'à 16 sous-fonctions peuvent se cacher derrière un code d'instruction. P. ex., la commande 7 représente «compte A = ...» Le paramètre indique la fonction de calcul exécutée.

Les commandes et les données sont ensuite écrites ensemble comme un octet dans la notation hexadécimale.

Ainsi, la commande 14h est issu de la commande 1 et du paramètre 4. Le symbole hexadécimal est abandonné, car toutes les commandes et les adresses sont en principe en écriture hexadécimale.

Les trois premières commandes sont les suivantes :

10-1F : sortie de port directe sur A1-A4, 0-15, binaire 0000 à 1111

20-2F : délai d'attente 0-15

(1, 2, 5, 10, 20, 50, 100, 200, 500, 1 000, 2 000, 5 000, 10 000, 20 000, 30 000, 60 000 ms)

30-3F : saut retour 0-15

La commande 1 sert à la sortie de port d'un nombre constant. Vous pourrez ainsi émettre n'importe quelle configuration binaire et, p. ex., activer plusieurs LED simultanément.

La commande d'attente 2 utilise un paramètre qui contient le temps en millisecondes et échelonné en 1-2-5 niveaux. Compte tenu du volume insuffisant de nombre de 0 à 15, les périodes de temporisation sont exécutés entre une milliseconde et une minute. Pour de plus longues périodes, vous devriez programmer une exécution répétée d'une commande d'attente, p. ex. dans une boucle de comptage.

La commande de retour 3 est particulièrement simple et suffit pour de nombreuses opérations, dans lesquelles un processus doit être répété indéfiniment. La longueur du saut est limité sur la plage jusqu'à 15. Etant donné que la longueur de saut s'applique relativement à l'adresse actuelle, des parties de programme peuvent être déplacées à d'autres adresses grâce à ce retour.

Le programme de clignotant alterné fonctionne avec ces trois commandes. Il doit être légèrement modifié ici dans la plage d'adresse et écrit à partir de 00. Même la configuration binaire de la sortie et les temps d'attente ont été modifiés.

Adresse	Commande	Données	Remarque
00	1	1	A1-4 = 0001
01	2	7	délai 200 ms
02	1	4	A1-4 = 0100
03	2	7	délai 200 ms
04	3	4	saut -4

Liste 9 : programme clignotement

Le programme ressemble donc à ça en écriture hexadécimale :

11 27 14 27 34

Un grand nombre de programmes simples peut ainsi être écrit sur la base de ces trois premières commandes.

Analysez et testez les trois programmes suivants. Le but étant que vous puissiez utiliser ces commandes de manière intuitive. Les séquences simples de programme, telles que celle-ci, peuvent être programmées et saisies directement après quelques exercices, même de tête. Par exemple, un simple chenillard avec quatre configurations de sortie :

Adresse	Commande	Données	Remarque
00	1	1	LED 0001
01	2	8	délai 500 ms
02	1	2	LED 0010
03	2	8	délai 500 ms
04	1	4	LED 0100
05	2	8	délai 500 ms
06	1	8	LED 1000
07	2	8	délai 500 ms
08	3	8	Saut -8

11 28 12 28 14 28 18 28 38

Liste 10 : chenillard 1

Ajoutez deux configurations de sortie supplémentaire, de sorte que le point lumineux fonctionne en va et vient. Testez également avec d'autres configurations de sortie et délais de temporisation.

Adresse	Commande	Données	Remarque
00	1	1	LED 0001
01	2	8	délai 500 ms
02	1	2	LED 0010
03	2	8	délai 500 ms
04	1	4	LED 0100
05	2	8	délai 500 ms
06	1	8	LED 1000
07	2	8	délai 500 ms
08	1	4	LED 0100
09	2	8	délai 500 ms
0A	1	2	LED 0010
0B	2	8	délai 500 ms
0C	3	C	saut -12

11 28 12 28 14 28 18 28 14 28 12 28 3C

Liste 11 : lumière défalante 2, va et vient

Une minuterie peut contenir avec une commande d'attente une temporisation pouvant aller jusqu'à une minute. Un retour avec une longueur de saut 0 se trouve à la fin, soit une boucle infinie sans contenu qui sert de fin de programme. Un nouveau démarrage est déclenché avec le bouton reset. Développez encore une fois le programme pour en faire une minuterie de cuisine de trois minutes. Vous pouvez alors représenter le temps restant par le nombre de LED allumées servant d'indicateur.