micro:bit TPS

This is my implementation of the TPS. The instructions will be compatible to my ArduinoSPS Version. And you will get some nice new Commands, implementing some of the micro:bit features, like images, Soundlevel, Logo...

If you find a bug, feel free to create a issue in the tracker.

Installation

To install the micro: bit TPS version, please simply copy the file microbit_tps.hex to the micro: bit drive.

Command implementation Chart

The actual command implementation list for the micro:bit V2:

	0	1	2	3	4	5	6	7
	n.n.	Port [DOUT]	Delay [WAIT]	Jump back relative [RJMP]	A=# [LDA]	=A	A=	A=Ausdruck
0	NOP [NOP]	aus	1ms	0	0	A<->B [SWAP]		
1	SetPixel(X,Y) X=A, Y=B	1	2ms	1	1	B=A [MOV]	A=B [MOV]	A=A + 1 [INC]
2	ClearPixel(X,Y) X=A, Y=B	2	5ms	2	2	C=A [MOV]	A=C [MOV]	A=A - 1 [DEC]
3	0: ClearDisplay 1..63: show(Image)	3	10ms	3	3	D=A [MOV]	A=D [MOV]	A=A + B [ADD]
4		4	20ms	4	4	Dout=A [STA]	Din [LDA]	A=A - B [SUB]
5		5	50ms	5	5	Dout.1=A.1 [STA]	Din.1 [LDA]	A=A * B [MUL]
6		6	100ms	6	6	Dout.2=A.1 [STA]	Din.2 [LDA]	A=A / B [DIV]
7		7	200ms	7	7	Dout.3=A.1 [STA]	Din.3 [LDA]	A=A and B [AND]
8		8	500ms	8	8	Dout.4=A.1 [STA]	Din.4 [LDA]	A=A or B [OR]
9		9	1s	9	9	PWM.1=A [STA]	ADC.1 [LDA]	A=A xor B [XOR]
a		10	2s	10	10	PWM.2=A [STA]	ADC.2 [LDA]	A= not A [NOT]
b		11	5s	11	11	Servo.1=A [STA]	RCin.1 [LDA]	A= A % B (Rest) [MOD]
c		12	10s	12	12	Servo.2=A [STA]	RCin.2 [LDA]	A= A + 16 * B [BYTE]
d		13	20s	13	13	E=A [MOV]	A=E [MOV]	A= B - A[BSUBA]
e		14	30s	14	14	F=A [MOV]	A=F [MOV]	A=A SHR 1 [SHR]
f		15	60s	15	15	Push A [PUSH]	Pop A [POP]	A=A SHL 1 [SHL]

new commands for the micro:bit

SetPixel: sets a pixel directly with x,y coordinates. X=A Y=B

ClearPixel: clears a pixel

ShowImage(image): if image is set to 0, the display is cleared, otherwise it will set a nice image on the display. Number to image, see appendix.

	8	9	a	b	c	d	e	f
	Page [PAGE]	Jump absolut (#+16*page) [JMP]	C* C>0: C=C-1; Jump # + (16*page) [LOOPC]	D* D>0:D=D-1; Jump # + (16*page) [LOOPC]	Skip if	Call # + (16*Page) [Call]	Callsub/Ret	Byte Befehle
0	0	0	0	0	A==0 [SKIP0]	0	ret [RTR]	A=ADC.1 [BLDA]
1	1	1	1	1	A>B [AGTB]	1	Call 1 [CASB]	A=ADC.2 [BLDA]
2	2	2	2	2	A<B [ALTB]	2	2 [CASB]	A=RCin.1 [BLDA]
3	3	3	3	3	A==B [AEQB]	3	3 [CASB]	A=RCin.2 [BLDA]
4	4	4	4	4	Din.1==1 [DEQ1 1]	4	4 [CASB]	PWM.1=A [BSTA]
5	5	5	5	5	Din.2==1 [DEQ1 2]	5	5 [CASB]	PWM.2=A [BSTA]
6	6	6	6	6	Din.3==1 [DEQ1 3]	6	6 [CASB]	Servo.1=A [BSTA]
7	7	7	7	7	Din.4==1 [DEQ1 4]	7		Servo.2=A [BSTA]
8	8	8	8	8	Din.1==0 [DEQ0 1]	8	Def 1 [DFSB]	Tone=A [TONE]
9	9	9	9	9	Din.2==0 [DEQ0 2]	9	2 [DFSB]	GetACC a=acc.x, E=acc.y, F=acc.z
a	10	10	10	10	Din.3==0 [DEQ0 3]	10	3 [DFSB]	A= Compass (in 5°)
b	11	11	11	11	Din.4==0 [DEQ0 4]	11	4 [DFSB]	A=SoundLevel()
c	12	12	12	12	S_PRG==0 [PRG0]	12	5 [DFSB]	A=LightLevel (0..255)
d	13	13	13	13	S_SEL==0 [SEL0]	13	6 [DFSB]	A=LogoTouched
e	14	14	14	14	S_PRG==1 [PRG1]	14		A=Gesture()
f	15	15	15	15	S_SEL==1 [SEL1]	15	restart [REST]	PrgEnd [PEND]

new commands for the micro:bit

GetACC: get values from the accelerator, A will be the x-axis, E the y-axis, and F the z-axis all Values range form 0..255

Compass: get the value of the compass, the value is in 5° Steps, so 0 = 0° 1 = 5°, 2=10°...

SoundLevel: level of the microphone

LightLevel: level of the ambient light

Gesture: is the gesture you where making with the micro:bit. The following gestures will be detected:

No.	Gesture	No.	Gesture
0	nothing	6	face down
1	moving up	7	freefall
2	moving down	8	3g
3	moving left	9	6g
4	moving right	10	8g
5	face up	11	shake

LogoTouched: the logo is touched.

Hardware assignments:

Caution: Due to the dual assignment of pins (especially the two A / D converters) can cause effects on the circuit in both directions. Protective diodes may be required there.

Button A is PRG or S1 (pin 5)
Button B is SEL or S2 (pin 11)
servo pins: Servo 1: pin 8, Servo 2: pin 9
ppm pins: pin 3, pin 4

Micro:bit pin mapping table

pin number	micro:bit function	TPS function
0	a/d	DOut.1
1	a/d	DOut.2
2	a/d	DOut.3
3	LED Col 3 a/d	A/D 1, RCin 1
4	LED Col 1 a/d	A/D 2, RCin 2
5	Button A	PRG/S1
6	LED Col 4	unusable
7	LED Col 2	unusable
8		D/A 1, Servo 1
9		D/A 2, Servo 2
10	LED Col 5 a/d	unusable
11	Button B	SEL/S2
12	reserved	DOut.4
13		DIn.1
14		DIn.2
15		DIn.3
16		DIn.4
19	I2C	unusable
20	I2C	unusable

Debug mode

This micro: bit TPS version supports debug and single step mode. In debug mode, additional information is made available on the serial interface while the program is being executed. A terminal program (such as hterm: https://www.der-hammer.info/pages/terminal.html) is required for this. Settings: 115200 baud 8N1.

```
-
PC: 0000
INST: 1, DATA: 1
Register:
A: 00, B: 00, C: 00
D: 00, E: 00, F: 00
Page: 00, Ret: 0000
```

PC is the program counter. INST and DATA are the nibbles of the command. The current status of the registers is shown under Register. PAGE is the page register and RET contains the return address for a subroutine call (via command 0xD #).
While the single step mode can only be set via source code, the pure debug mode can be started by touching the logo during a reset.

Apendix

Image List

Here is the image list:

0: clear display,
1: Image.HEART,
2: Image.HAPPY,
3: Image.SMILE,
4: Image.SAD,
5: Image.CONFUSED,
6: Image.ANGRY,
7: Image.ASLEEP,
8: Image.SURPRISED,
9: Image.SILLY,
10: Image.FABULOUS,
11: Image.MEH,
12: Image.YES,
13: Image.NO,
14: Image.CLOCK1,
15: Image.CLOCK2,

16: Image.CLOCK3,
17: Image.CLOCK4,
18: Image.CLOCK5,
19: Image.CLOCK6,
20: Image.CLOCK7,
21: Image.CLOCK8,
22: Image.CLOCK9,
23: Image.CLOCK10,
24: Image.CLOCK11,
25: Image.CLOCK12,
26: Image.ARROW_N,
27: Image.ARROW_NE,
28: Image.ARROW_E,
29: Image.ARROW_SE,
30: Image.ARROW_S,
31: Image.ARROW_SW,

32: Image.ARROW_W,
33: Image.ARROW_NW,
34: Image.TRIANGLE,
35: Image.TRIANGLE_LEFT,
36: Image.CHESSBOARD,
37: Image.DIAMOND,
38: Image.DIAMOND_SMALL,

```
39: Image.SQUARE,
40: Image.SQUARE_SMALL,
41: Image.RABBIT,
42: Image.COW,
43: Image.MUSIC_CROTCHET,
44: Image.MUSIC_QUAVER,
45: Image.MUSIC_QUAVERS,
46: Image.PITCHFORK,
47: Image.XMAS,

48: Image.PACMAN,
49: Image.TARGET,
50: Image.TSHIRT,
51: Image.ROLLERSKATE,
52: Image.DUCK,
53: Image.HOUSE,
54: Image.TORTOISE,
55: Image.BUTTERFLY,
56: Image.STICKFIGURE,
57: Image.GHOST,
58: Image.SWORD,
59: Image.GIRAFFE,
60: Image.SKULL,
61: Image.UMBRELLA,
62: Image.SNAKE,
63: Image.HEART_SMALL
```